

1.1 INTRODUCCIÓN

La Cooperativa de Servicios Telefónicos Tarija, tiene como misión: "SERVICIOS DE TELECOMUNICACIONES DE CALIDAD QUE CONTRIBUYAN AL DESARROLLO SOCIAL Y HUMANO, PROMOVRIENDO LA INTEGRACIÓN DEPARTAMENTAL CON RESPONSABILIDAD, ÉTICA Y SOLIDARIDAD"

Es indudable que Internet es una excelente herramienta para aprender y comunicarse. El internet tiene hoy en día un impacto profundo en nuestros trabajos, estudios, e incluso en la forma de entretenernos en nuestros ratos libres. Nos permite tener acceso al conocimiento a nivel mundial, enciclopedias, noticieros, acceso a bibliotecas y otros materiales educativos de valor e investigar sobre temas muy variados.

El internet ha pasado a convertirse en un servicio básico más en nuestros hogares, permitiéndonos comunicarnos con personas ubicadas en cualquier parte del mundo.

La cooperativa pretende que sus socios tengan acceso al conocimiento mundial, y sean parte de la era de la información; brindando a los socios la conexión con la información disponible en la red, y tratando de abarcar la mayor cantidad de hogares posibles.

El procesamiento manual de la información está siendo reemplazado por el proceso automático, mediante el uso de sistemas de información adecuado a las necesidades y requerimientos de cada empresa.

Actualmente el nodo de internet no cuenta con un sistema automatizado, para el manejo de la información, al realizar un control manual, la información está sujeta a pérdidas y demoras, derivando en un lento proceso de instalación de nuevos usuarios. Al no contar con estadísticas del consumo de los usuarios ya existentes, se ve complicada la tarea de hacer

informes para poder responder frente a los reclamos presentados por algunos usuarios.

Debemos tener información a mano y confiable para poder encarar ampliaciones. También es de vital importancia poder gestionar de manera eficaz los usuarios que ya existen, disponiendo de estadísticas de conexiones, consumo de ancho de banda y todos los datos necesarios.

El presente trabajo propone desplegar una herramienta automatizada que facilite gestionar todos los procesos llevados a cabo en el nodo de Internet.

Se desarrollará un sistema que consta de tres módulos, el módulo de Control de Módems, módulo de Control Puertos y el módulo Mikrotik, que es el que usara una API desarrollada en PHP, para conectarse con el Router, y así poder guardar en una base de datos todas las estadísticas de conexión de los usuarios.

Las herramientas que darán solución a los problemas ya enmarcados, serán el lenguaje PHP, con el framework LARAVEL, XAMPP como servidor web. y MySQL como base de datos.

1.2 ANTECEDENTES

1.2.1 Sistema Actual

En la actualidad la sección de Internet no cuenta con un Sistema de Gestión para el manejo de la información del nodo. El proceso es manual, haciendo el registro de algunos datos en una hoja de cálculo. El nodo carece de un sistema que posibilite realizar un adecuado control de los módems que ingresan al área de Internet; no existiendo un registro del destino de los mismos, desconociendo si son vendidos a los usuarios o dados en calidad de comodato. Tampoco existe una herramienta informática que permita llevar un apropiado control de los puertos en las diferentes tarjetas, que existen en los DSLAMS de las distintas sub-centrales. No teniendo una información certera y oportuna de la cantidad de los puertos disponibles y el estado de

los mismos. Para el control de usuarios Adsl, se utiliza Winbox, una herramienta para Windows que interactúa con un Router Mikrotik, que es donde los usuarios Adsl se autentifican.

Los datos y estadísticas de conexión de los usuarios que maneja winbox, son volátiles; se pierden cuando el router se reinicia.

Existen en el mercado software para gestión de un ISP, tales como Sequeisp y Isp-Soft.

1.3 PLANTEAMIENTO DEL PROBLEMA

El proceso y manejo de la información en el área de Internet se lo realiza de manera manual, lo que implica pérdida de información y es susceptible a errores. Al no contar con información oportuna sobre la cantidad de puertos disponibles y el estado de los mismos, como así también la cantidad de modems disponibles; implica un retardo a la hora de planificar futuras ampliaciones y nuevas instalaciones.

Cuando un socio solicita la instalación del servicio Internet ADSL, debemos ir a los registros físicos para verificar la existencia de un puerto disponible y un modem para brindarle al socio, tarea ardua y tediosa, la misma que ocasiona demora. Por este motivo muchos socios se molestan en el tiempo que demora en instalarles el servicio. Todo esto sin contar que a veces los datos en los registros físicos se pierden, y es necesario acudir tanto al almacén para verificar la existencia de modems o acudir a la subcentral a realizar la verificación física de la disponibilidad de puerto para la nueva instalación requerida.

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar un Sistema de Gestión para automatizar y agilizar el manejo de la información en el Nodo Internet Cosett.

1.2.2 Objetivos Específicos

- Realizar un análisis de los requerimientos del área de internet dentro la empresa de telecomunicaciones Cosett.
- Implementar un sistema de interfaz amigable para mejor interacción del usuario.
- Diseñar una Base de Datos confiable para el control de módems, estado de puertos y gestión de usuarios ADSL.
- Instrumentar el sistema con la tecnología adecuada.
- Aplicar las fases de la metodología RUP para el desarrollo del trabajo.
- Adaptar el lenguaje de modelado unificado (UML) para la modelación lógica y tecnológica.
- Elaborar la base de datos actualizada de puertos.
- Implementar la base de datos de módems
- Obtener datos reales para la comprobación al momento de la realización del Sistema de gestión.

1.5 JUSTIFICACIÓN

En la actualidad los avances tecnológicos nos permiten contar con técnicas y herramientas de software para poder desarrollar sistemas informáticos mediante los cuáles se pueda tener integridad, seguridad y confiabilidad de la información, por lo que contar con un sistema que permita llevar los registros de la información de una manera automatizada es fundamental para poder tener una buena administración. Por este motivo y haciendo uso de estos recursos tecnológicos se justifica el desarrollo de un sistema que

cuenta con una base de datos que administre la información referente a instalaciones y usuarios ADSL.

1.5.1 Justificación social

Cosett al ser el principal proveedor de internet en nuestra ciudad, y dado el rápido crecimiento del internet, es que tiene la urgente necesidad de contar con una herramienta informática que le permita contar con información oportuna y rápida.

Para el desarrollo del presente trabajo se pone en práctica los conocimientos adquiridos en los años de estudio en la Carrera de Ingeniería Informática, demostrando así que la universidad genera profesionales capaces y aptos para enfrentar cualquier desafío aplicando las metodologías y técnicas adquiridas.

1.5.2 Justificación institucional

El sistema permitirá en primera instancia al área de Internet tener un mejor control de la información que maneja en los procesos de instalaciones Adsl.

El sistema será una herramienta de ayuda para nuestra cooperativa, facilitando el intercambio de información entre distintas áreas que necesiten de dicha información. Permitiendo de esta manera contar con información inmediata y confiable.

El sistema será flexible y fácilmente adaptable en cualquier otro escenario que posea características similares.

1.5.3 Justificación tecnológica

La herramienta a desarrollar es justificada tecnológicamente debido a que en el nodo de Internet se cuenta con equipos de computación provistos de las características necesarias para la explotación del sistema.

La herramienta puesta a consideración se desarrollará con herramientas de software libre, bajo plataforma Linux. Evitando así el tema de licencias.

1.6 ALCANCES Y LIMITACIONES

1.6.1 Alcances

El sistema se desarrollará sobre plataforma Linux.

La disponibilidad del sistema será las 24 horas los 7 días de la semana.

Módulo Control Modems

- Administración de módems que entran al área de internet.
- Administración del destino de módems.
- Reporte Módems Vendidos.
- Reporte Módems Como-dato.
- Reporte Módems Devueltos.
- Alerta número mínimo de módems en el nodo.
- Reporte módems disponibles.

Módulo Control puertos

- Administración Puertos
- Número de puertos disponibles en total.
- Número de puertos disponibles por subcentral.
- Reporte estado de cada puerto.
- Gestión puertos vía web.

Módulo Mikrotik

- Administración de Usuarios.

- Altas, Bajas y Modificaciones de Usuarios.
- Administración de categorías o planes ADSL.
- Estadísticas y reporte consumo de ancho de banda por usuario.
- Control y reporte tiempo conexión de cada usuario.
- Búsqueda por ip para conocer usuarios conectados.
- Configuración remota de módems.
- Control y reporte del total de usuarios creados en router.
- Control y reporte de usuarios activos.
- Reporte de usuarios por perfil.
- Reporte de usuarios por zona(Vlan).
- Reporte consumo de ancho de banda por zona.
- Gráficas de consumo ancho de banda.
- Reporte usuarios Concurrentes.
- Establecimiento de hora pico.
- Establecimiento de ancho de banda total requerido.

1.6.2 Limitaciones

El sistema propuesto presenta las siguientes limitaciones;

- No realizará la facturación del consumo de los usuarios ADSL.
- No practicará cortes automáticos.
- No efectuará reconexiones automáticas.
- No se conectará con el sistema de Almacenes para alertar cuando queden pocos módems.

3.1 CAPTURA DE REQUISITOS COMO CASOS DE USO

Para realizar la captura de los requisitos de una manera óptima, se necesita un conjunto de técnicas que ayuden a tener una visión clara del sistema para poder avanzar en los flujos de trabajo subsiguientes. La especificación de requisitos tradicional se reemplaza por tanto por un conjunto de artefactos: el modelo de casos de uso.

3.1.1 Identificación y Breve Descripción de Actores

	Administrador
--	---------------

1. El Administrador representa a la persona que interactúa con el sistema, el cual está registrado previamente, es el encargado también de las tareas administrativas del sistema, realiza copias de seguridad y la restauración de la misma en caso de una eventualidad.

3.1.2 Identificación y Descripción de Casos de Uso

CU – 01	Conectar
ACTORES	Usuario Administrador
OBJETIVO	Ingresar al Sistema
PRECONDICIONES	Ingresar a la página de login del sistema
POSTCONDICIONES	1. Los datos serán enviados para su comprobación
FLUJO DE EVENTOS	Actividades del Actor: 1. Ingresa a la página de login del sistema 2.-Ingresar los datos requeridos en el formulario 3.-Enviar los datos Respuesta del Sistema: 4. El sistema despliega mensaje comprobando
MANEJO DE SITUACIONES EXCEPCIONALES	→ Si se omite información requerida para el ingreso, el sistema mostrará un mensaje indicando el campo faltante y no proseguirá hasta que se complete.

CU – 02	Verificar Usuario y Contraseña
ACTORES	Usuario administrador
OBJETIVO	Comprobar la existencia y veracidad de los datos
PRECONDICIONES	Debe haber introducido y enviado los datos
POSTCONDICIONES	Tendrá acceso al sistema
FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <p>Respuesta del Sistema:</p> <ol style="list-style-type: none"> 1. Verifica la existencia del usuarios 2. Confirma si la contraseña es correcta 3. Despliega mensaje “Has ingresado correctamente”
MANEJO DE SITUACIONES EXCEPCIONALES	<p>→ Si los datos no son correctos, el sistema desplegará un mensaje de error.</p>

CU – 03	Administrar Módem
ACTORES	Usuario administrador
OBJETIVO	<p>Gestión de Módems</p> <ul style="list-style-type: none"> • Adicionar Módem • Modificar Datos Módem • Dar de Baja Módem
PRECONDICIONES	Debe estar logueado en el sistema y seleccionar Administrar Módems.
POSTCONDICIONES	1. Información sobre el Módem queda registrado en el sistema
FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <ol style="list-style-type: none"> 1. Seleccionar Administrar Módems 3. Ingresar la opción deseada 5. Ingresar los datos requeridos en el formulario 4.- Enviar los datos <p>Respuesta del Sistema:</p> <ol style="list-style-type: none"> 2. El sistema despliega la página con las opciones para administrar Módems. 4. El sistema despliega el formulario para llenar la

	información requerida. 5. Confirma la operación con el mensaje: “Operación realizada satisfactoriamente”
MANEJO DE SITUACIONES EXCEPCIONALES	→ Si se omite información requerida , el sistema mostrará un mensaje del campo faltante y no prosigue hasta que se complete.

CU - 04	Asignar Destino Módem
ACTORES	Usuario administrador
OBJETIVO	Administrar el Destino de los Módems <ul style="list-style-type: none"> • Adicionar Destino • Modificar Destino • Eliminar Destino
PRECONDICIONES	Debe estar logueado en el sistema y seleccionar Administrar Destino Módem
POSTCONDICIONES	1. Los datos del destino del Módem quedan actualizados en el sistema
FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <ol style="list-style-type: none"> 1. Seleccionar Administrar Destino Módem 3. Elegir la opción deseada 5. Ingresar los datos requeridos en el formulario 6. Enviar los datos <p>Respuesta del Sistema:</p> <ol style="list-style-type: none"> 2. Despliega página con las opciones para Administrar Destino del Módem. 4.- Despliega formulario para llenar información requerida de acuerdo a la opción deseada 7. Confirma la operación con el mensaje: “Operación realizada con éxito”
MANEJO DE SITUACIONES EXCEPCIONALES	→ Si se omite información requerida , el sistema mostrará un mensaje del campo faltante y no prosigue hasta que se complete.

CU - 05	Generar Reporte Módem
ACTORES	Usuario administrador
OBJETIVO	Generar Reportes sobre destinos del Módem
PRECONDICIONES	Debe estar logueado en el sistema, en la opción Administrar Destino Módem y seleccionar Generar Reportes
POSTCONDICIONES	1. Los datos requeridos de acuerdo a los criterios quedan plasmados en un reporte
FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <ol style="list-style-type: none"> 1. Seleccionar Generar Reportes 3. Ingresar los datos requeridos en el formulario 4. Enviar datos <p>Respuesta del Sistema:</p> <ol style="list-style-type: none"> 2. Despliega el formulario para generar reportes 5. Confirma con el mensaje: “Ha generado el reporte satisfactoriamente”
MANEJO DE SITUACIONES EXCEPCIONALES	→ Si se omite información requerida , el sistema mostrará un mensaje del campo faltante y no prosigue hasta que se complete.

CU - 06	Administrar Tarjeta
ACTORES	Usuario administrador
OBJETIVO	<p>Gestión Tarjetas Adsl</p> <ul style="list-style-type: none"> • Adicionar Tarjeta • Modificar Datos Tarjetas • Dar de Baja Tarjeta
PRECONDICIONES	Debe estar logueado en el sistema, en la opción Administrar Tarjetas
POSTCONDICIONES	Los datos de la tarjeta quedan registrados en el sistema
FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <ol style="list-style-type: none"> 1. Selecciona Administrar Tarjeta 3. Elegir la opción

	<p>5. Llenar Formulario</p> <p>6. Enviar datos</p> <p>Respuesta del Sistema:</p> <p>2. Despliega la página con las opciones para Administrar Tarjeta</p> <p>4. Despliega formulario para llenar la información requerida</p> <p>7. Confirma con el mensaje “Operación realizada exitosamente”</p>
MANEJO DE SITUACIONES EXCEPCIONALES	<p>→ Si se omite información requerida , el sistema mostrará un mensaje del campo faltante y no prosigue hasta que se complete.</p>

CU - 07	Administrar Puertos
ACTORES	Usuario administrador
OBJETIVO	<p>Administración de Puertos para internet</p> <ul style="list-style-type: none"> • Dar de Alta Puerto • Modificar Datos Puerto
PRECONDICIONES	Debe estar logueado en el sistema y seleccionar Administrar Puerto
POSTCONDICIONES	1. Los datos del puerto quedan registrados en el sistema
FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <p>1. Seleccionar administrar puerto</p> <p>3. Seleccionar la opción</p> <p>5. Llenar datos requeridos</p> <p>6. Enviar los datos</p> <p>Respuesta del Sistema:</p> <p>2. Despliega la página con las opciones de administración de puertos</p> <p>4. Despliega un formulario con los datos requeridos</p> <p>7. Confirma con el mensaje: “Operación realizada con éxito”</p>

MANEJO DE SITUACIONES EXCEPCIONALES	→
--	---

CU - 08	Administrar Destino Puerto
ACTORES	Usuario administrador
OBJETIVO	Administración del Destino de Puertos <ul style="list-style-type: none"> • Asignar puerto usuario • Modificar destino puerto usuario • Liberar Puerto
PRECONDICIONES	Debe estar logueado en el sistema y seleccionar Administrar Destino Puerto
POSTCONDICIONES	1. Los datos del puerto quedan actualizado en la base de datos.
FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <ol style="list-style-type: none"> 1. Seleccionar Administrar Destino de Puertos 3. Elegir la opción deseada 5. Llenar los datos requeridos 6. Enviar los datos <p>Respuesta del Sistema:</p> <ol style="list-style-type: none"> 2. Despliega la página con las opciones de administración de puertos 4. Despliega formulario con los datos requeridos 7. Confirma la operación con el mensaje: “Operación realizada con éxito”
MANEJO DE SITUACIONES EXCEPCIONALES	→ Si se omite información requerida para la reasignación del Módem mostrará un mensaje del campo faltante y no prosigue hasta que se complete.

CU - 9	Generar Reporte Puertos
ACTORES	Usuario administrador
OBJETIVO	Información sobre Puertos
PRECONDICIONES	Debe estar logueado en el sistema y Generar Reportes de puertos
POSTCONDICIONES	1. Los datos quedan plasmados en un reporte
FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <ol style="list-style-type: none"> 1. Seleccionar Generar Reportes Puertos 3. Ingresar los datos requeridos en el formulario 4.- Enviar los datos <p>Respuesta del Sistema:</p> <ol style="list-style-type: none"> 2. El sistema despliega el formulario para generar el reporte 5. Confirma la acción: "Reporte Generado Exitosamente"
MANEJO DE SITUACIONES EXCEPCIONALES	

CU - 10	Administrar Usuario
ACTORES	Usuario administrador
OBJETIVO	Administración Usuarios Internet <ul style="list-style-type: none"> • Adicionar Usuario • Modificar Datos Usuario • Dar de Baja Usuario
PRECONDICIONES	Debe estar logueado en el sistema y seleccionar seleccionar generar reportes
POSTCONDICIONES	1. Los datos requeridos son plasmados en reportes
FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <ol style="list-style-type: none"> 1. Seleccionar generar reportes 3. Ingresar los datos requeridos en el formulario 4.- Enviar los datos <p>Respuesta del Sistema:</p> <ol style="list-style-type: none"> 2. El sistema despliega el formulario para reportes 5. Genera el reporte de acuerdo a los criterios

	seleccionados
MANEJO DE SITUACIONES EXCEPCIONALES	→

CU - 11	Administrar Categoría
ACTORES	Usuario administrador
OBJETIVO	Administración de los planes ADSL <ul style="list-style-type: none"> • Adicionar Categoría • Modificar Categoría • Eliminar Categoría
PRECONDICIONES	Debe estar logueado en el sistema y seleccionar Administrar Categorías
POSTCONDICIONES	
FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <ol style="list-style-type: none"> 1. Seleccionar Administrar Categoría 3. Seleccionar la opción 3. Ingresar los datos requeridos en el formulario 4. Enviar los datos <p>Respuesta del Sistema:</p> <ol style="list-style-type: none"> 2. El sistema despliega la página con las opciones de Administrar Categoría 4. El sistema despliega el formulario para llenar los datos 5. Confirma la acción con el mensaje: “Operación realizada exitosamente”
MANEJO DE SITUACIONES EXCEPCIONALES	→ Si se omite información requerida, el sistema mostrará un mensaje del campo faltante y no prosigue hasta que se complete.

CU - 12	Generar Reportes Usuario
ACTORES	Usuario administrador
OBJETIVO	Generar Reportes sobre Usuarios
PRECONDICIONES	Debe estar logueado en el sistema y Generar Reportes Mikrotik
POSTCONDICIONES	1. Se genera un reporte con los datos requeridos

FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <ol style="list-style-type: none"> 1. Seleccionar modificar puerto 3. Ingresar los datos requeridos en el formulario 4. Enviar datos 6.- Modificar datos 5. Enviar los datos <p>Respuesta del Sistema:</p> <ol style="list-style-type: none"> 2. El sistema despliega el formulario la búsqueda del puerto a modificar 5. Despliega formulario con datos del puerto seleccionado 6. Confirma la modificación con el mensaje: “Ha modificado satisfactoriamente los datos del puerto”
MANEJO DE SITUACIONES EXCEPCIONALES	<ul style="list-style-type: none"> → Si no existe datos del puerto buscado, el sistema desplegará un mensaje indicando la no existencia → Si se omite información requerida para el registro de los nuevos datos del puerto, el sistema mostrará un mensaje del campo faltante y no prosigue hasta que se complete.

CU - 13	Realizar Copia Seguridad
ACTORES	Usuario administrador
OBJETIVO	Realizar una copia de Seguridad de la Base de Datos
PRECONDICIONES	Debe estar logueado en el sistema y seleccionar Copia de Seguridad.
POSTCONDICIONES	1. Los datos de la Base de Datos quedan respaldados en un archivo que es la copia de seguridad.
FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <ol style="list-style-type: none"> 1. Seleccionar Realizar Copia de Seguridad 3. Ingresar los datos requeridos en el formulario 4. Enviar datos <p>Respuesta del Sistema:</p> <ol style="list-style-type: none"> 2. El sistema despliega el formulario para realizar la copia

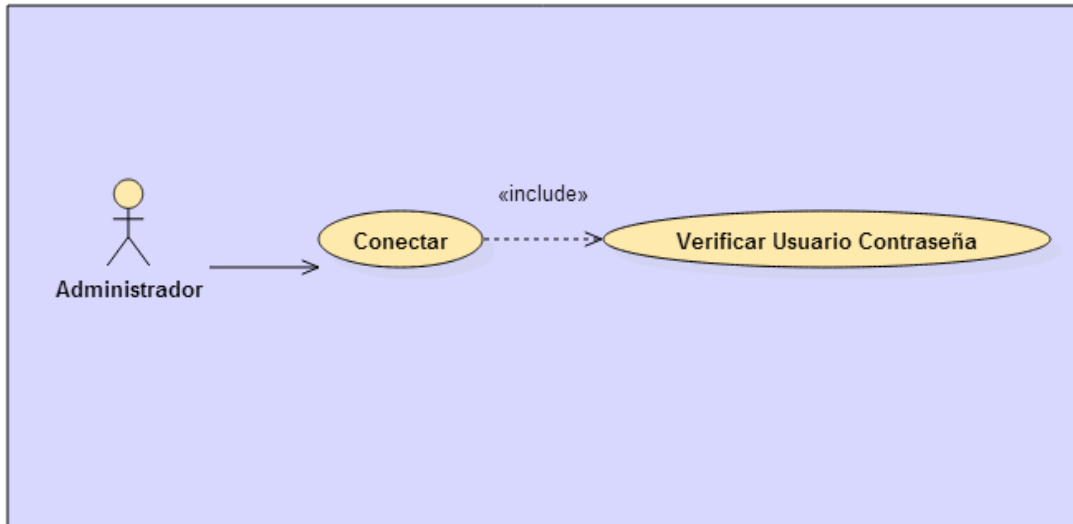
	de seguridad. 8. Confirma la copia de seguridad con el mensaje: “Ha realizado la copia de seguridad satisfactoriamente ”
MANEJO DE SITUACIONES EXCEPCIONALES	→ Si se omite información requerida en el formulario, el sistema mostrará un mensaje del campo faltante y no prosigue hasta que se complete.

CU - 14	Restaurar Base de Datos
ACTORES	Usuario administrador
OBJETIVO	Restaurar la base de datos desde una copia de seguridad
PRECONDICIONES	Debe estar logueado en el sistema y seleccionar Restaurar Base de Datos
POSTCONDICIONES	1. Los datos son recuperados a partir de una copia de seguridad.
FLUJO DE EVENTOS	<p>Actividades del Actor:</p> <ol style="list-style-type: none"> 1. Seleccionar Restaurar Base de Datos 3. Ingresar los datos requeridos en el formulario 4. Adjuntar el archivo con la copia de Seguridad. 5. Enviar Datos <p>Respuesta del Sistema:</p> <ol style="list-style-type: none"> 2. El sistema despliega el formulario para la restauración de la Base de Datos. 6. Confirma la restauración de la Base de Datos con el mensaje: “Se ha restaurado la BD satisfactoriamente ”
MANEJO DE SITUACIONES EXCEPCIONALES	<p>→ Si se omite información requerida en el formulario de restauración , el sistema mostrará un mensaje del campo faltante y no prosigue hasta que se complete.</p> <p>→ Si el archivo con la copia de seguridad no se reconoce, el sistema mostrará un mensaje de error</p>

3.1.3 Descripción del Modelo de Casos de Uso Completo por Escenario

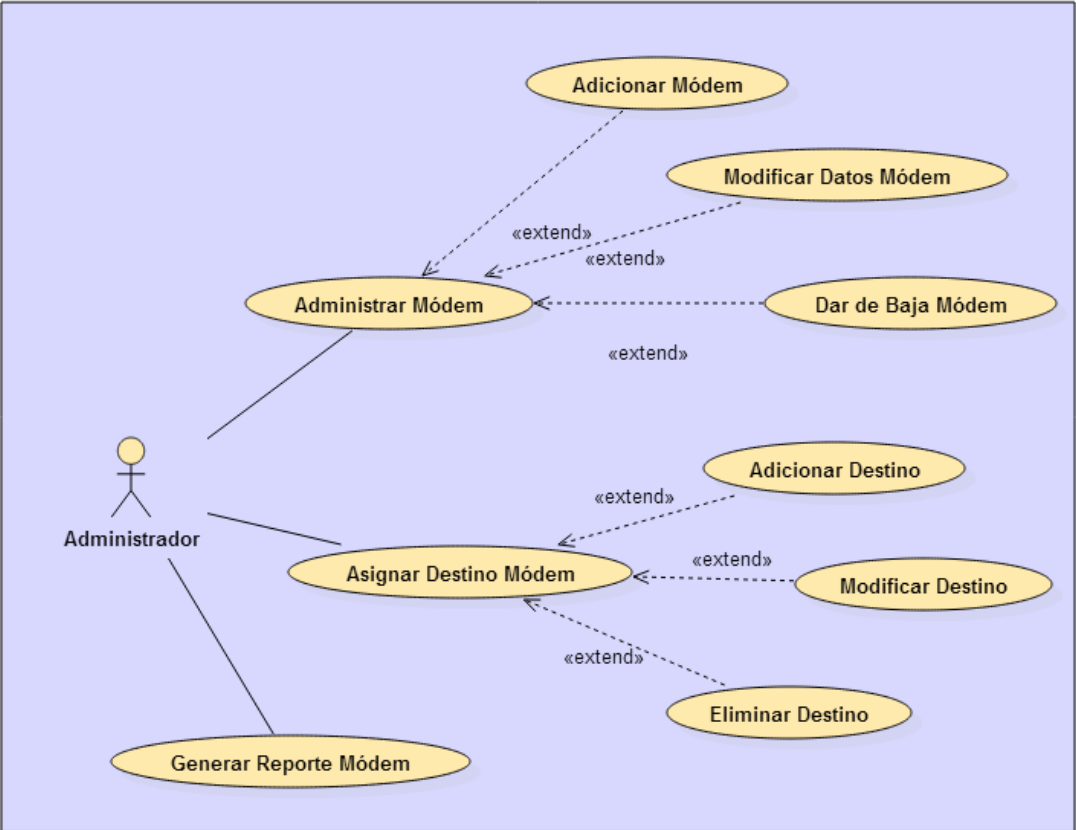
3.1.3.1 Casos de uso para conectarse

El usuario Administrador tiene asignado un nombre de usuario y una contraseña, que le permiten conectar al sistema después de la verificación de los datos en la base de datos.



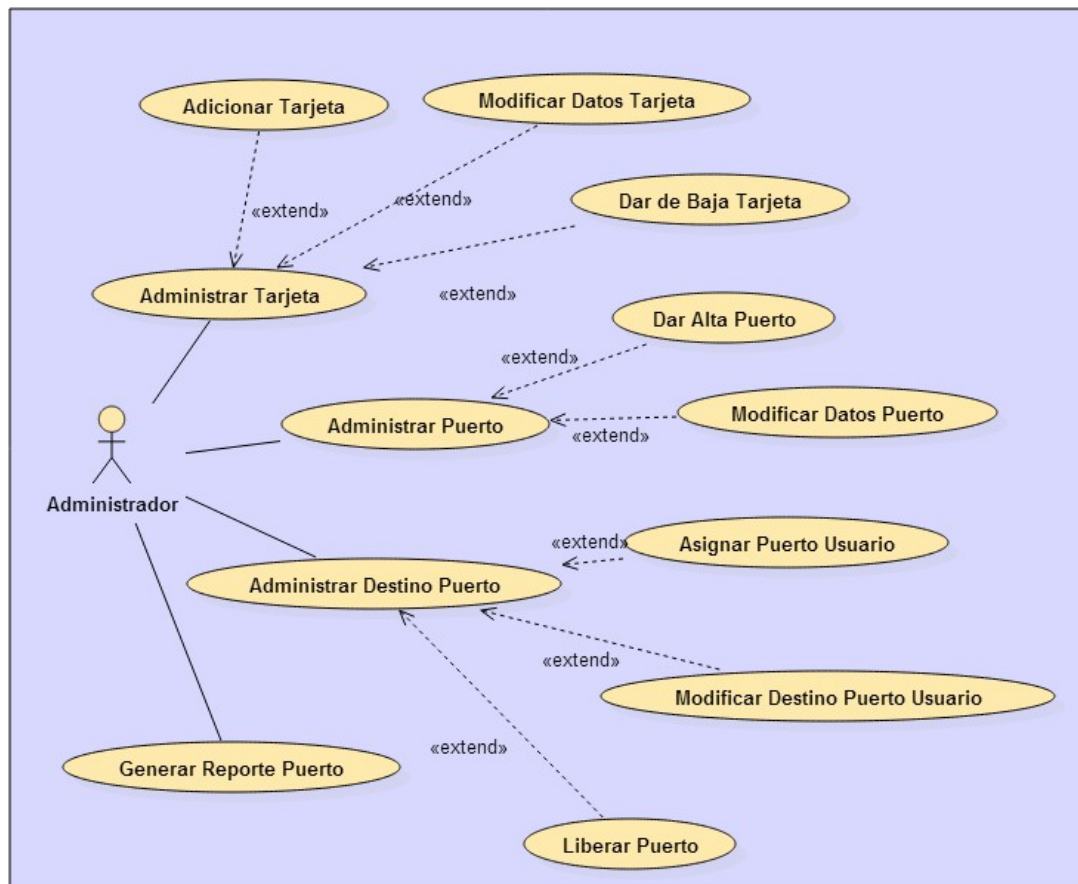
3.1.3.2 Casos de uso para el control de Modems

El usuario Administrador puede adicionar, modificar datos de un módem que se encuentra ya registrado, dar de baja un módem, asignar el módem a un usuario, modificar el destino y eliminar el destino del mismo, también puede generar reportes de acuerdo a los criterios que necesite.



3.1.3.3 Casos de uso control de Puertos

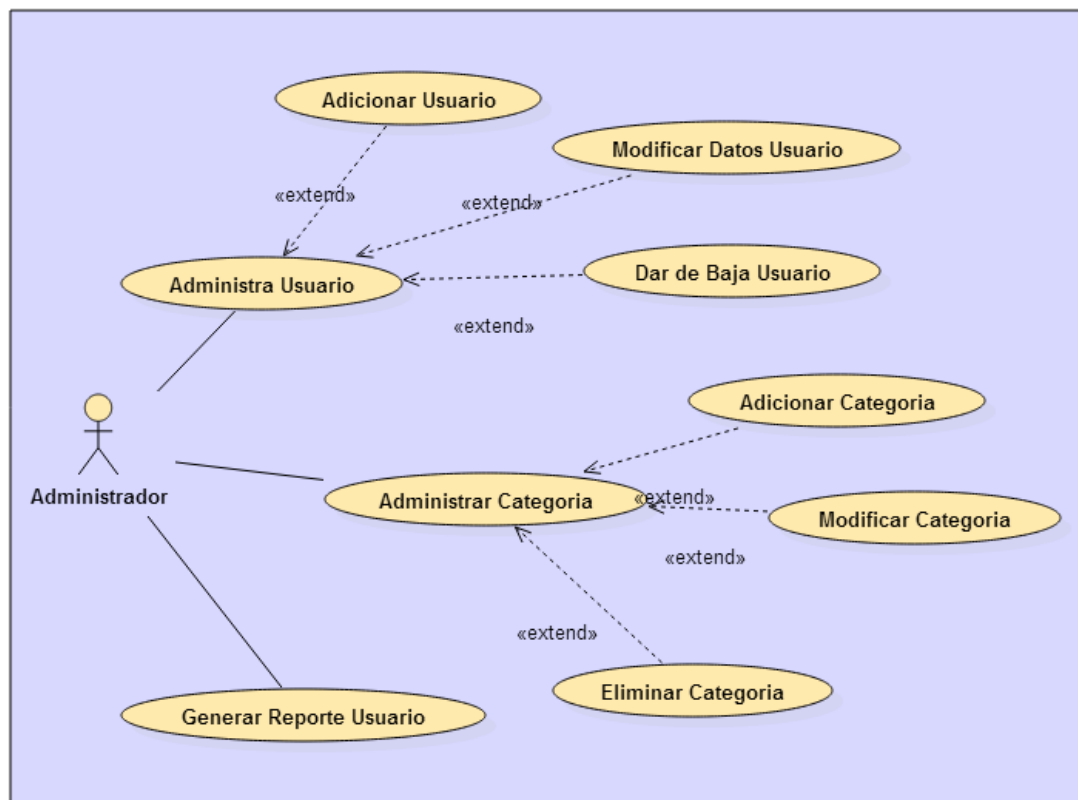
El usuario Administrador, puede adicionar tarjetas, administrar los puertos, adicionar, modificar datos de un puerto ya existente en el sistema, asignarlo a un usuario, modificar el destino del puerto o liberarlo cuando el usuario deja el servicio. También puede generar reportes de acuerdo a los criterios que necesite.



3.1.3.4 Casos de uso control de Mikrotik

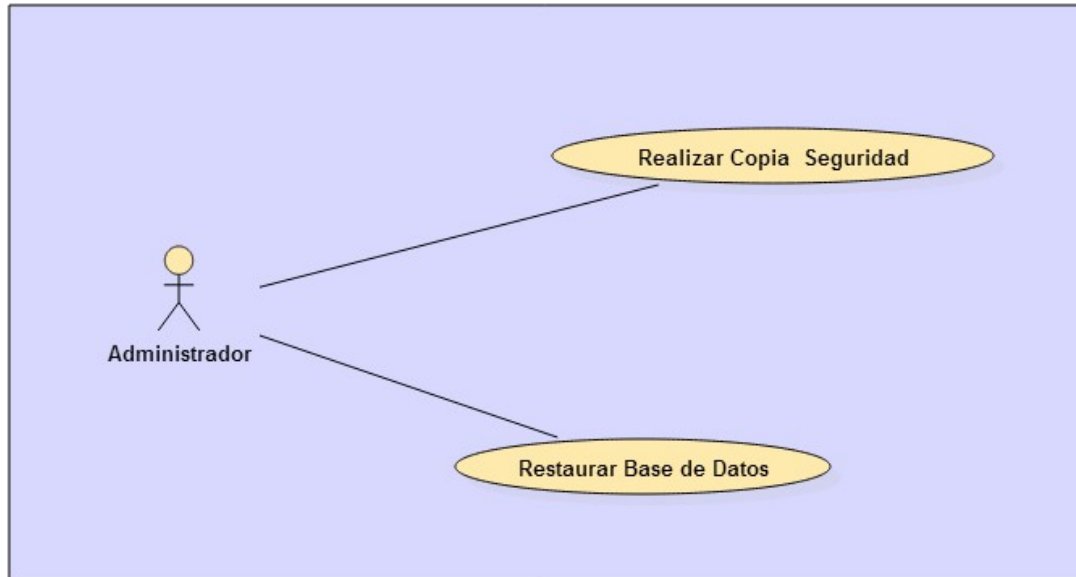
El usuario Administrador, es el encargado de controlar los usuarios que tienen el servicio de Internet, puede adicionar usuarios, modificar datos o eliminar usuarios de la base de datos. Puede adicionar categorías o planes de internet, modificar categoría existente actualizando velocidades, también puede eliminar categorías.

También tiene la opción de generar reportes de acuerdo a los criterios necesarios.



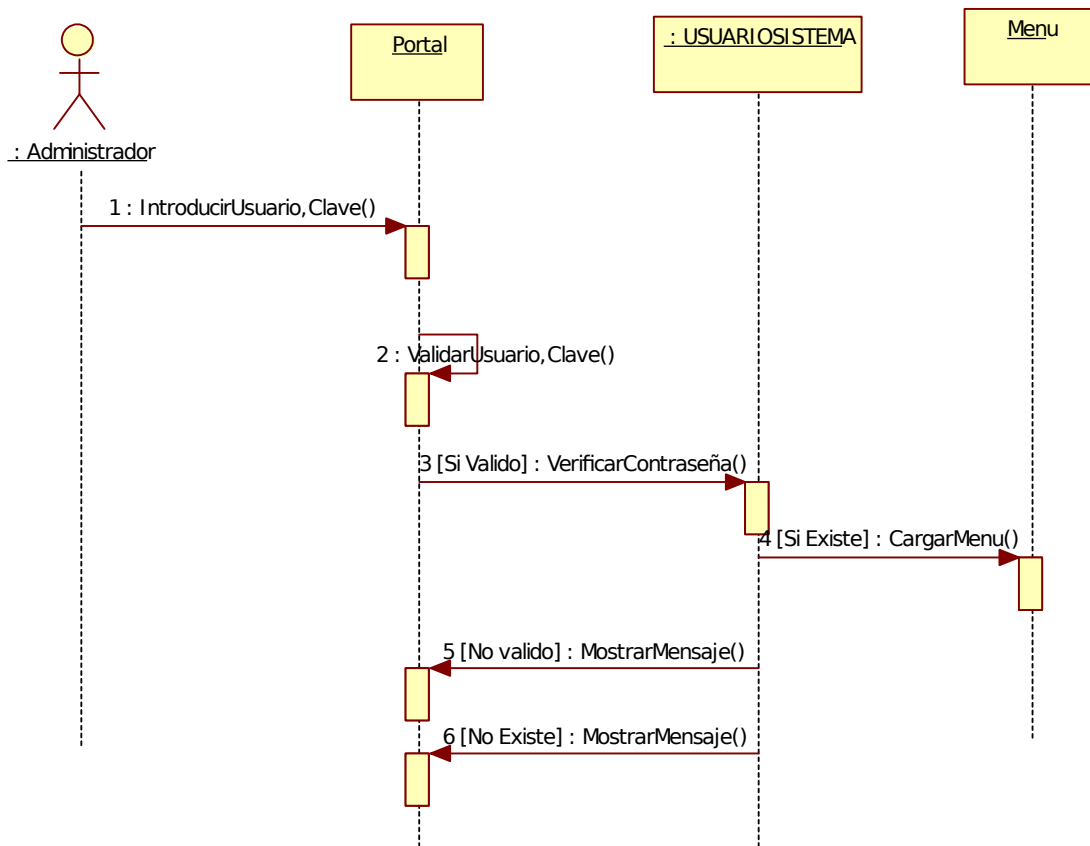
3.1.3.5 Caso de uso para seguridad

El usuario Administrador realiza las tareas administrativas del sistema, puede realizar una copia de seguridad y restaurar la misma, ante cualquier eventualidad.

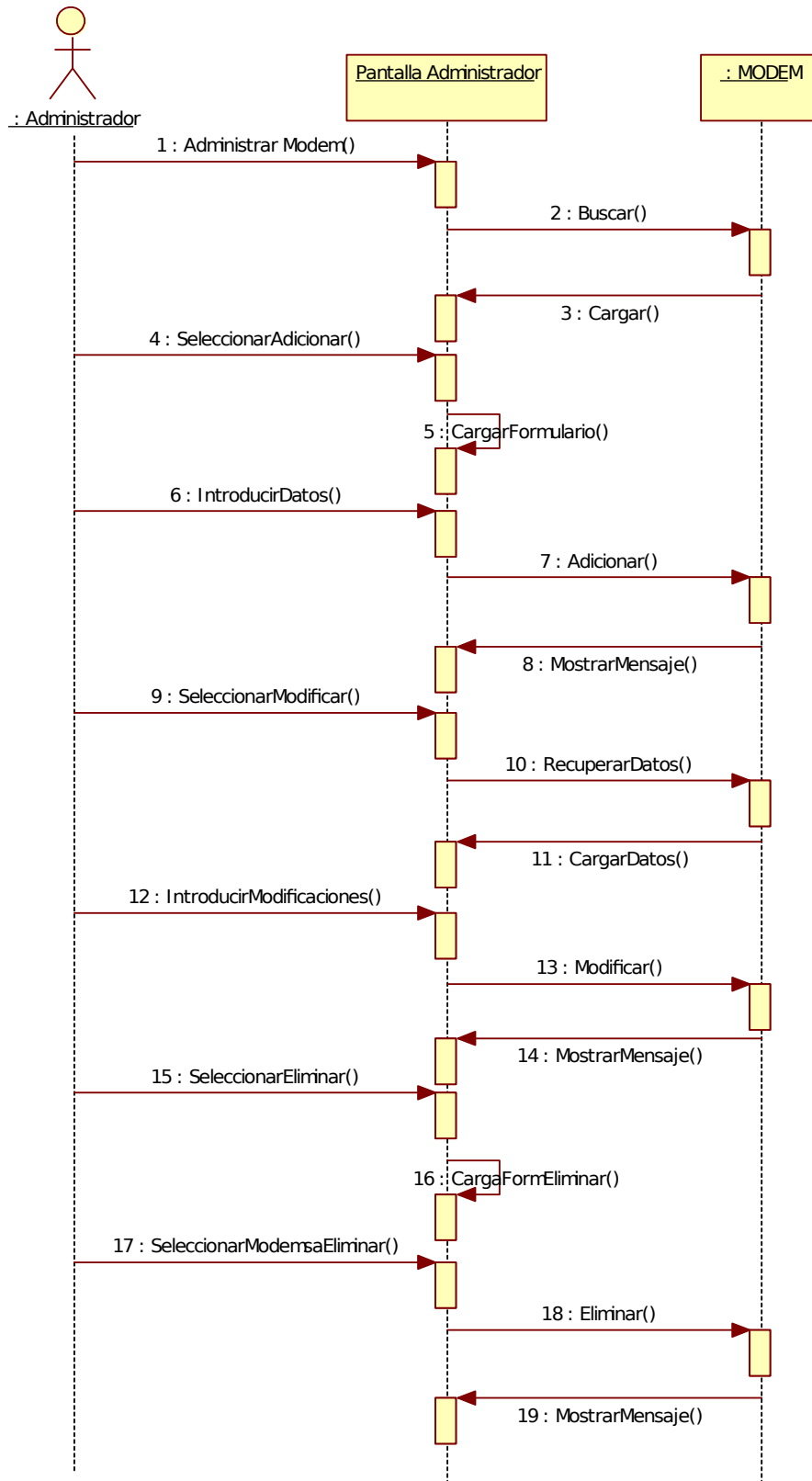


4. DIAGRAMAS DE SECUENCIA

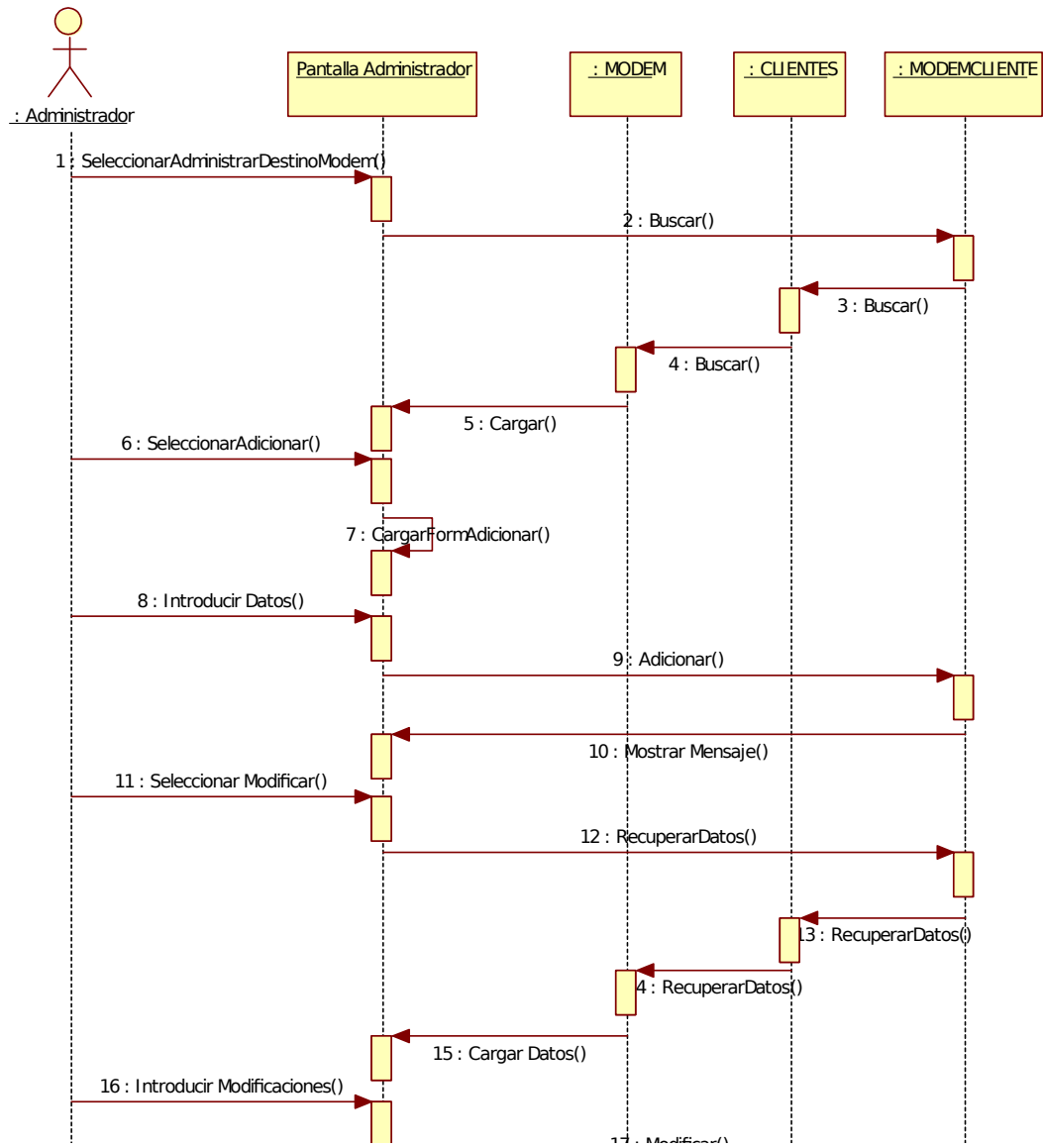
4.1. Diagrama de Secuencia del Caso de Uso Conectar



4.2. Diagrama de Secuencia del Caso de Uso Administrar Modem



4.3. Diagrama de Secuencia del Caso de Uso Administrar Destino Modem



4.5.

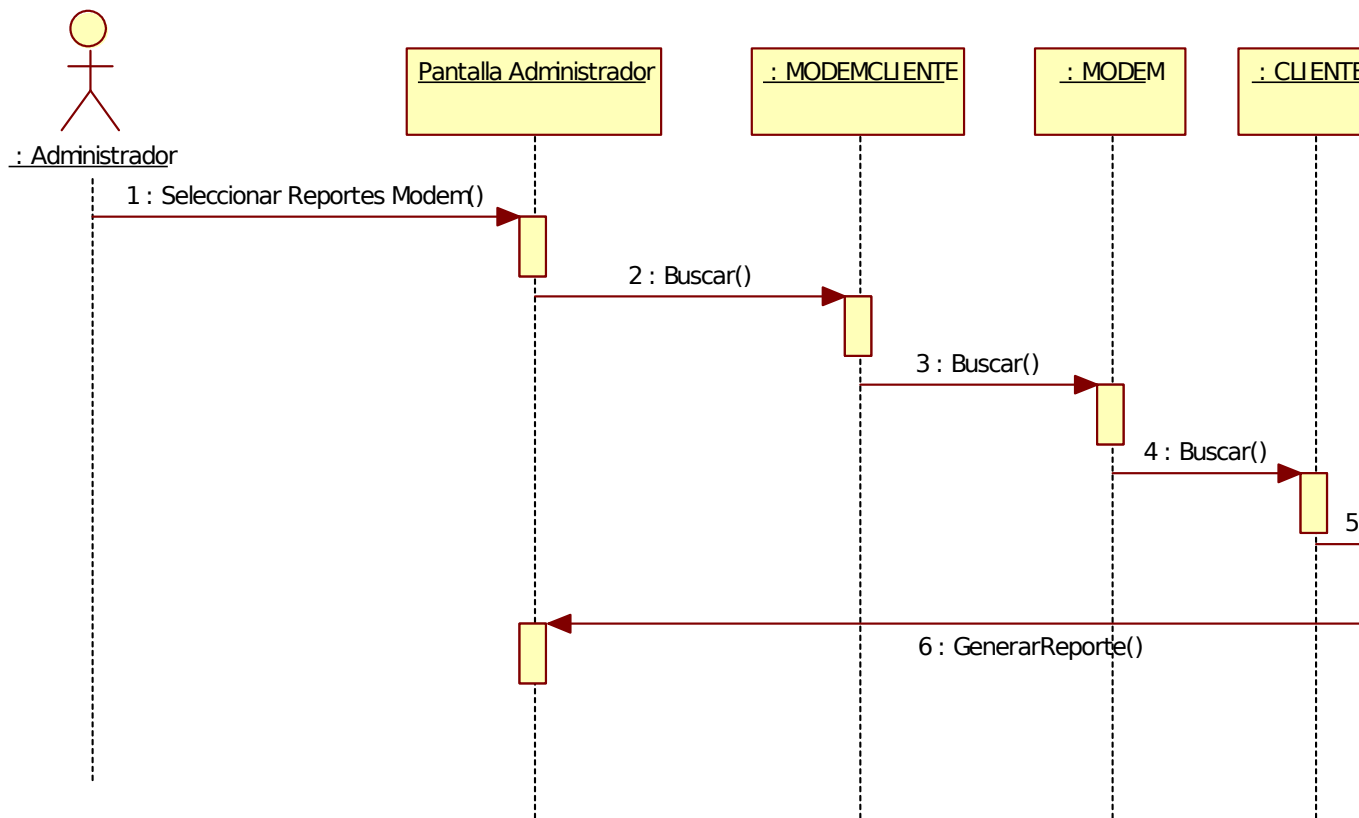
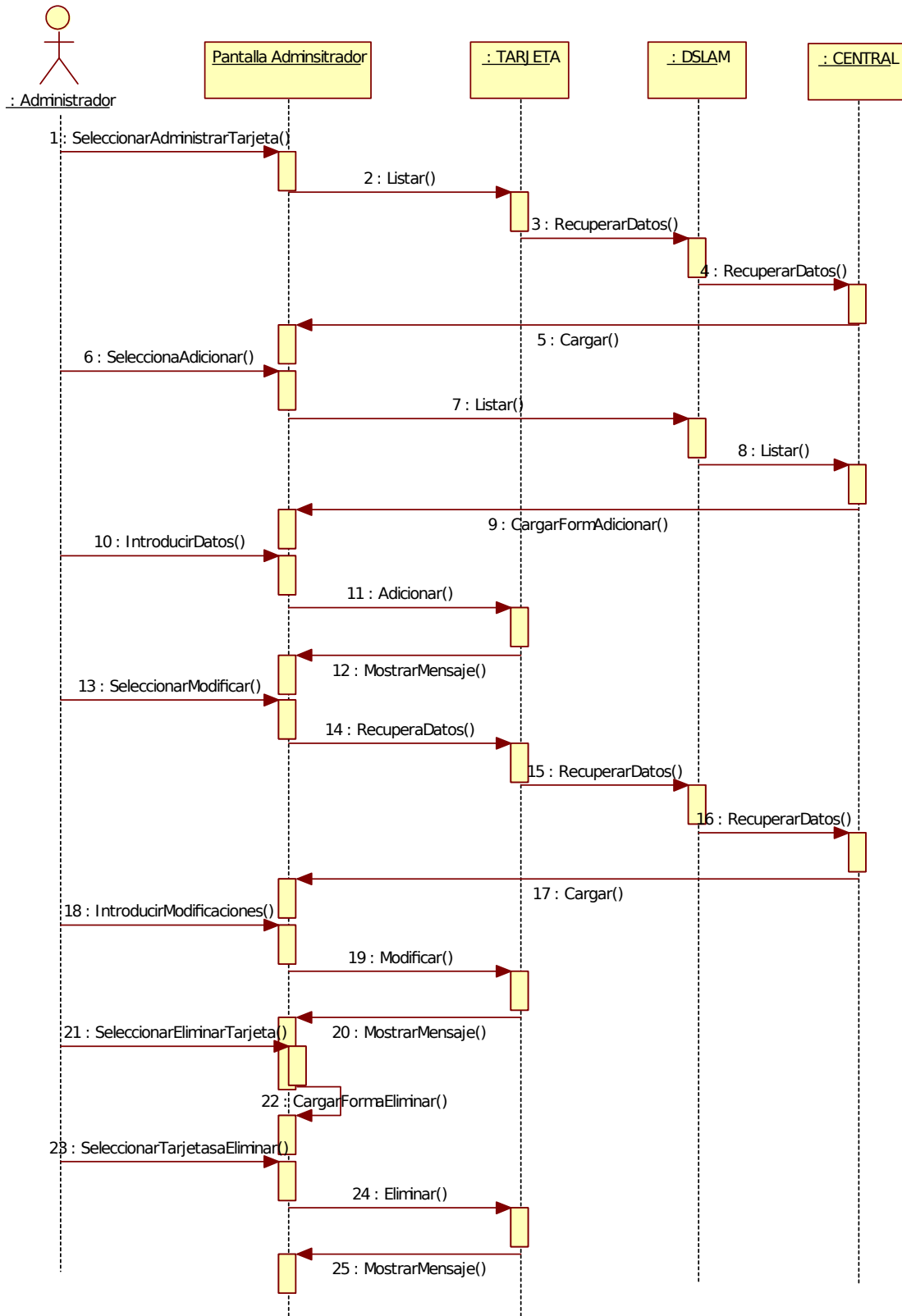
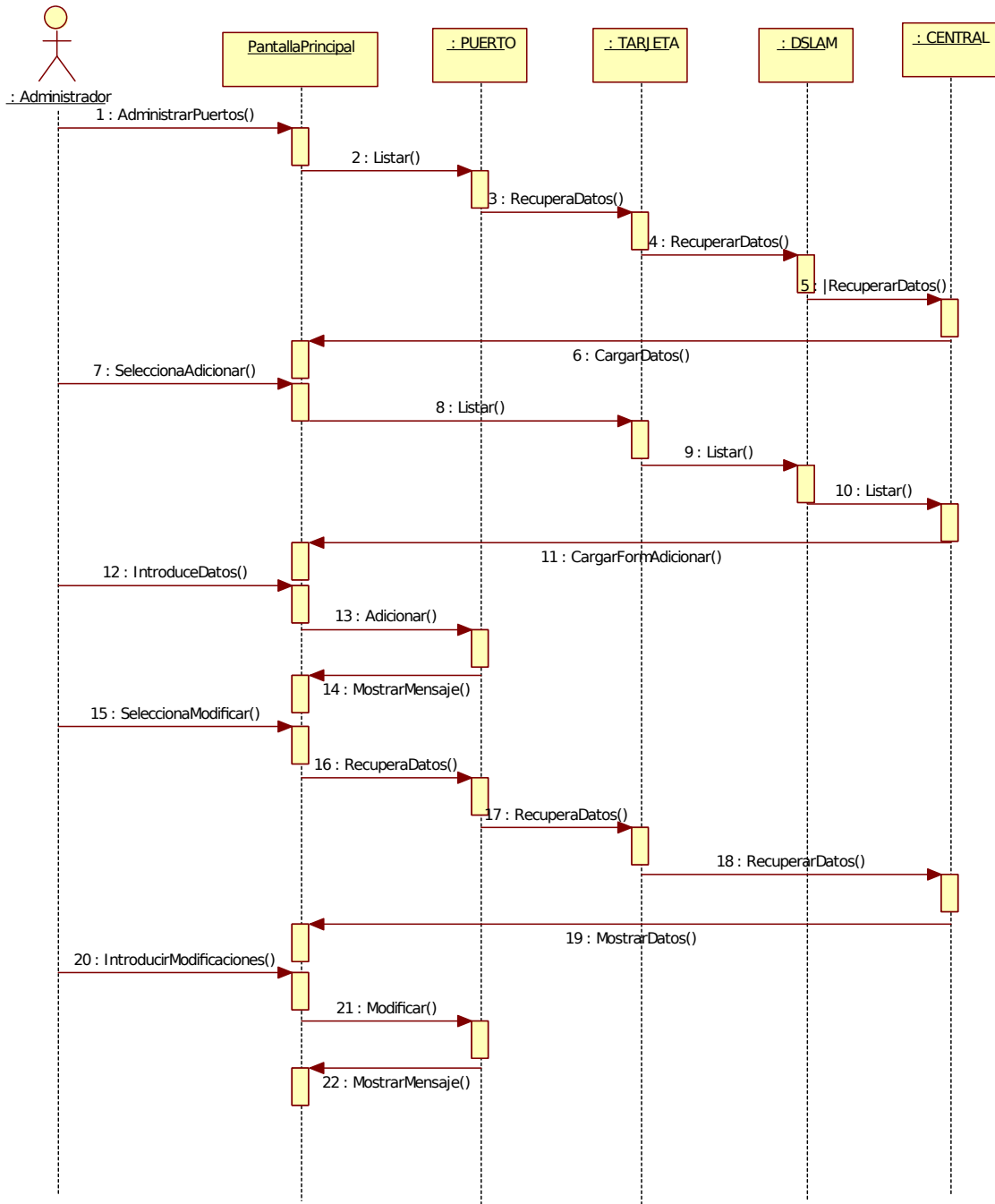


Diagrama de Secuencia del Caso de Uso Administrar Tarjetas



4.6.

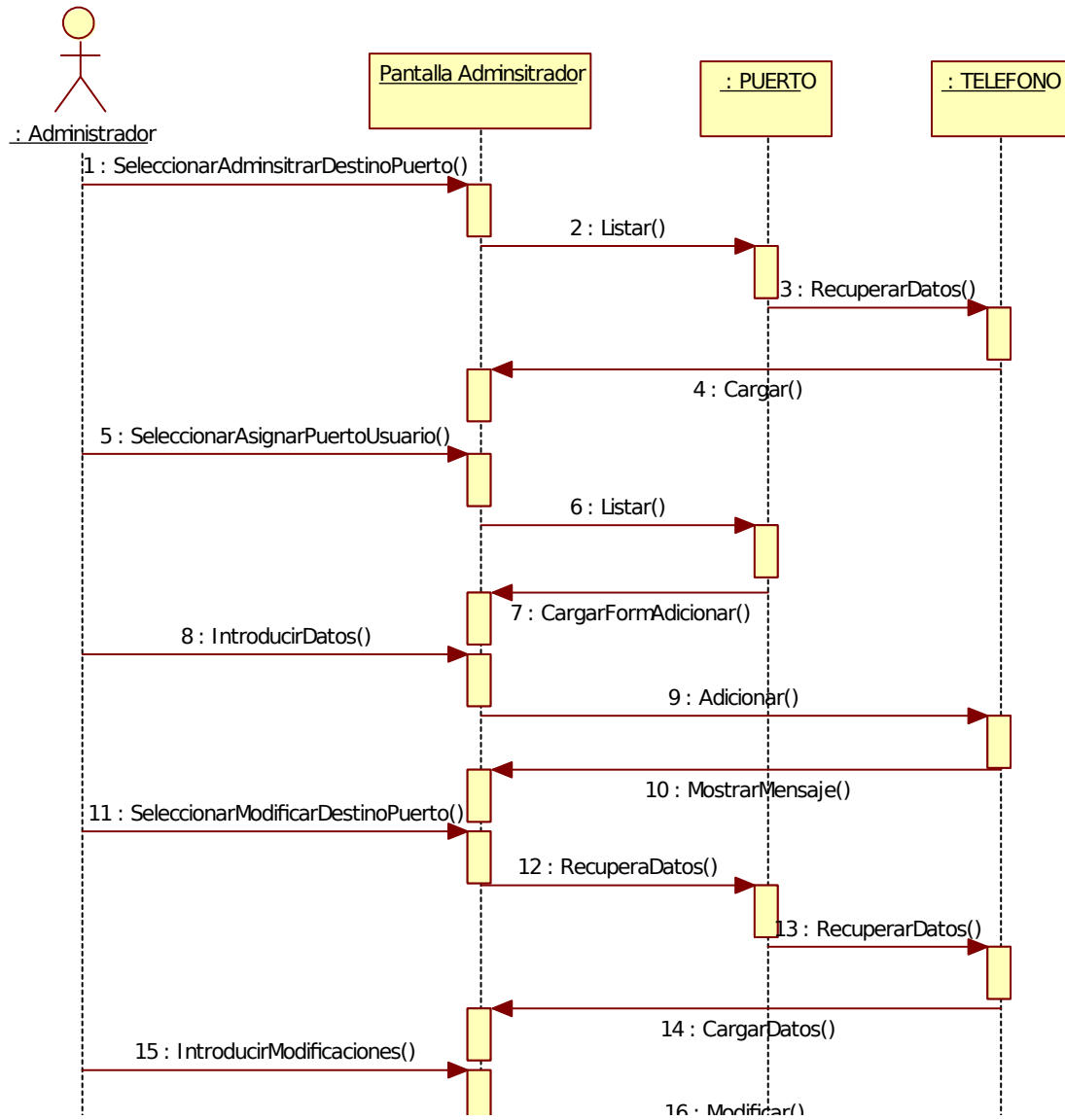
4.7. Diagrama de Secuencia del Caso de Uso Administrar Puertos



4.8.

4.9.

4.10. Diagrama de Secuencia del Caso de Uso Administrar Destino Puerto



4.12.

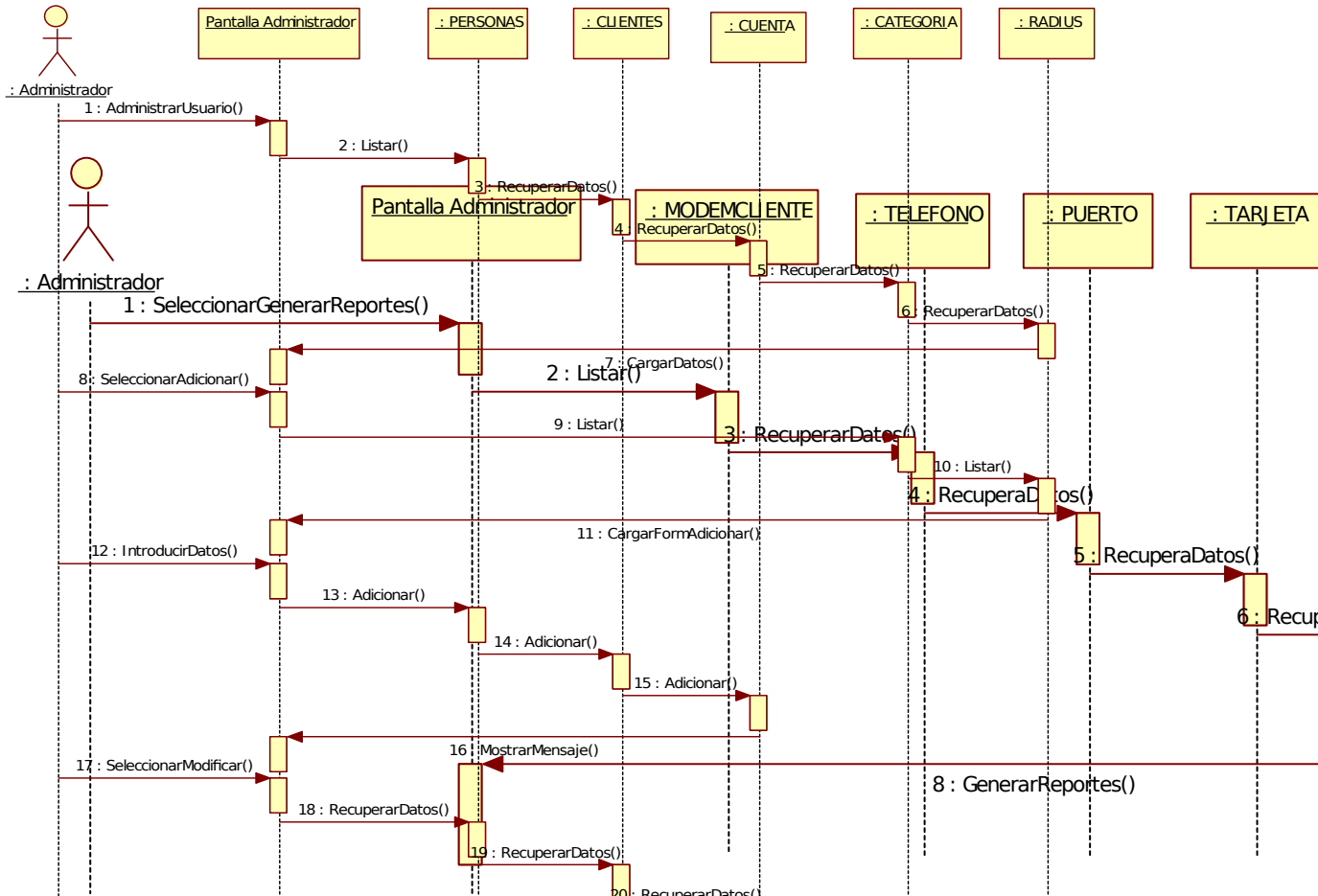
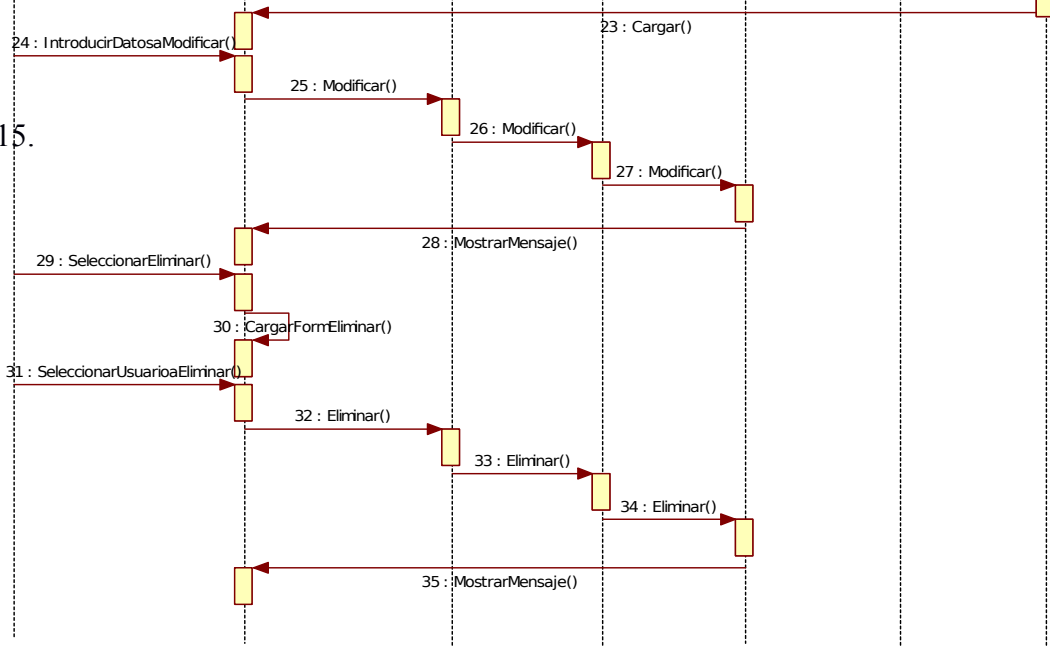


Diagrama de Secuencia del Caso de Uso Administrar Usuario

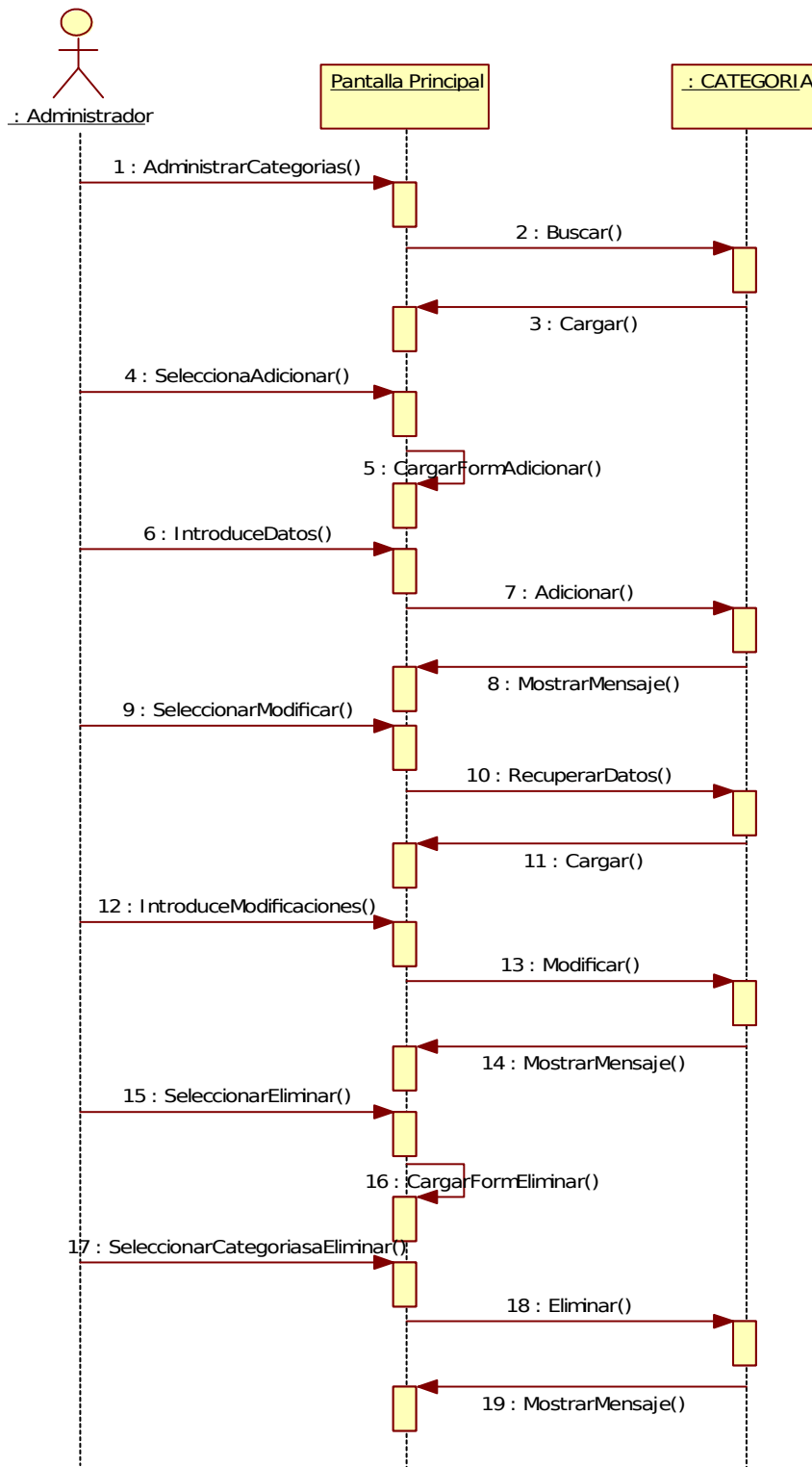
4.13.

4.14.

4.15.



4.16. Diagrama de Secuencia del Caso de Uso Administrar Categoría

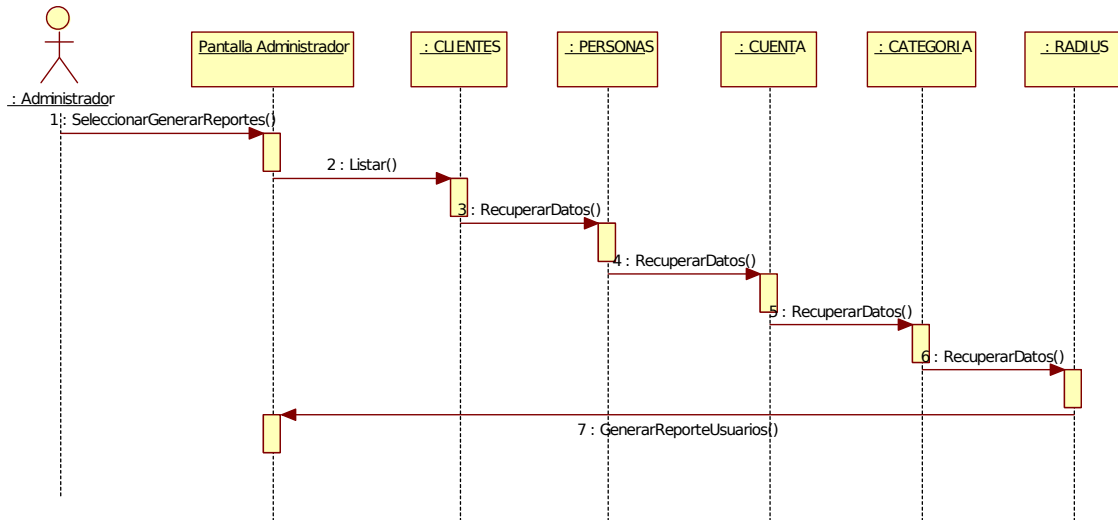


4.17.

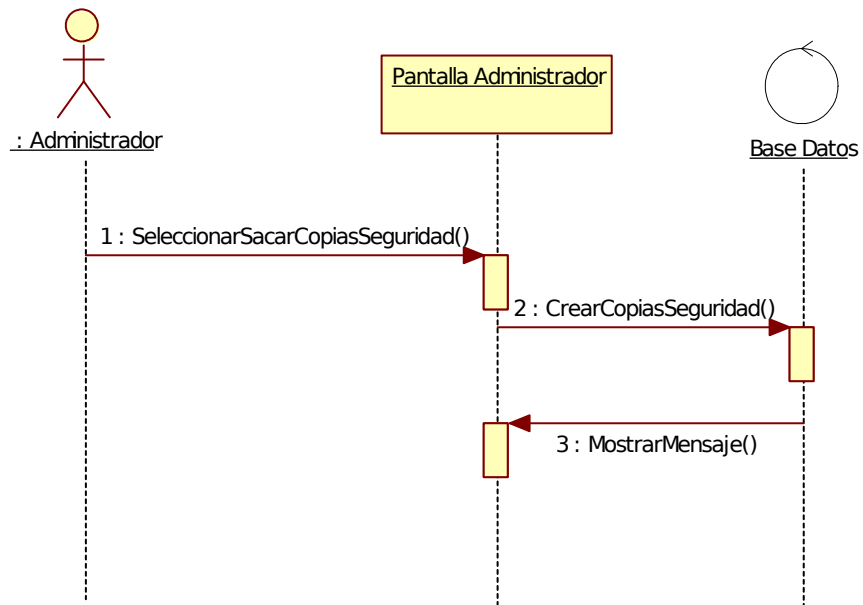
4.18.

4.19. Diagrama de Secuencia del Caso de Uso Generar Reportes Usuario

4.20.



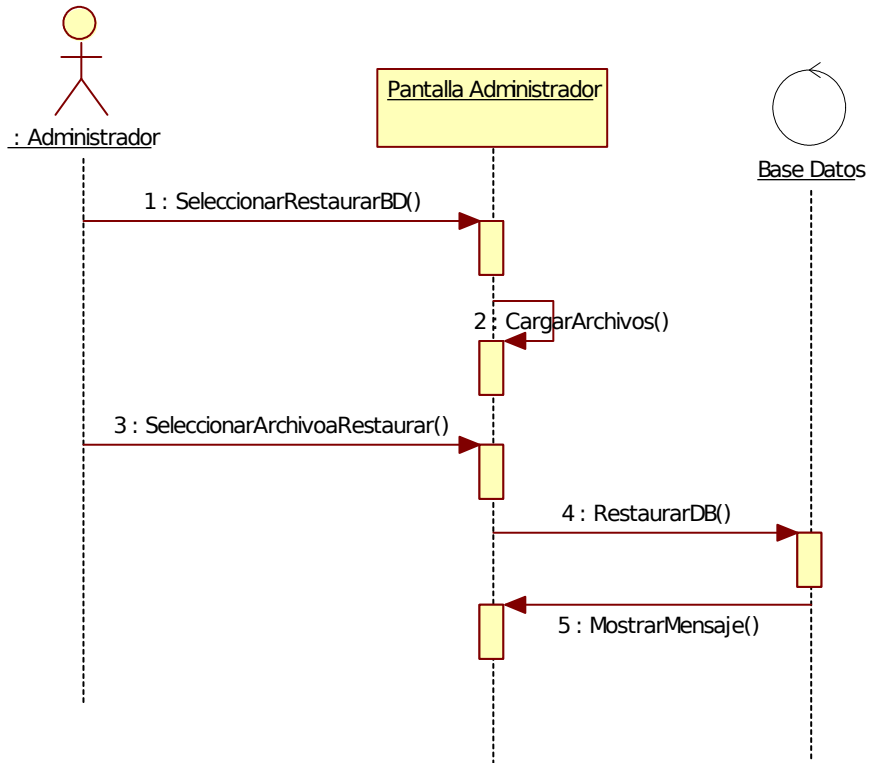
4.21. Diagrama de Secuencia del Caso de Uso Realizar Copia Seguridad



4.22.

4.23.

4.24. Diagrama de Secuencia del Caso de Uso Restaurar Base de Datos

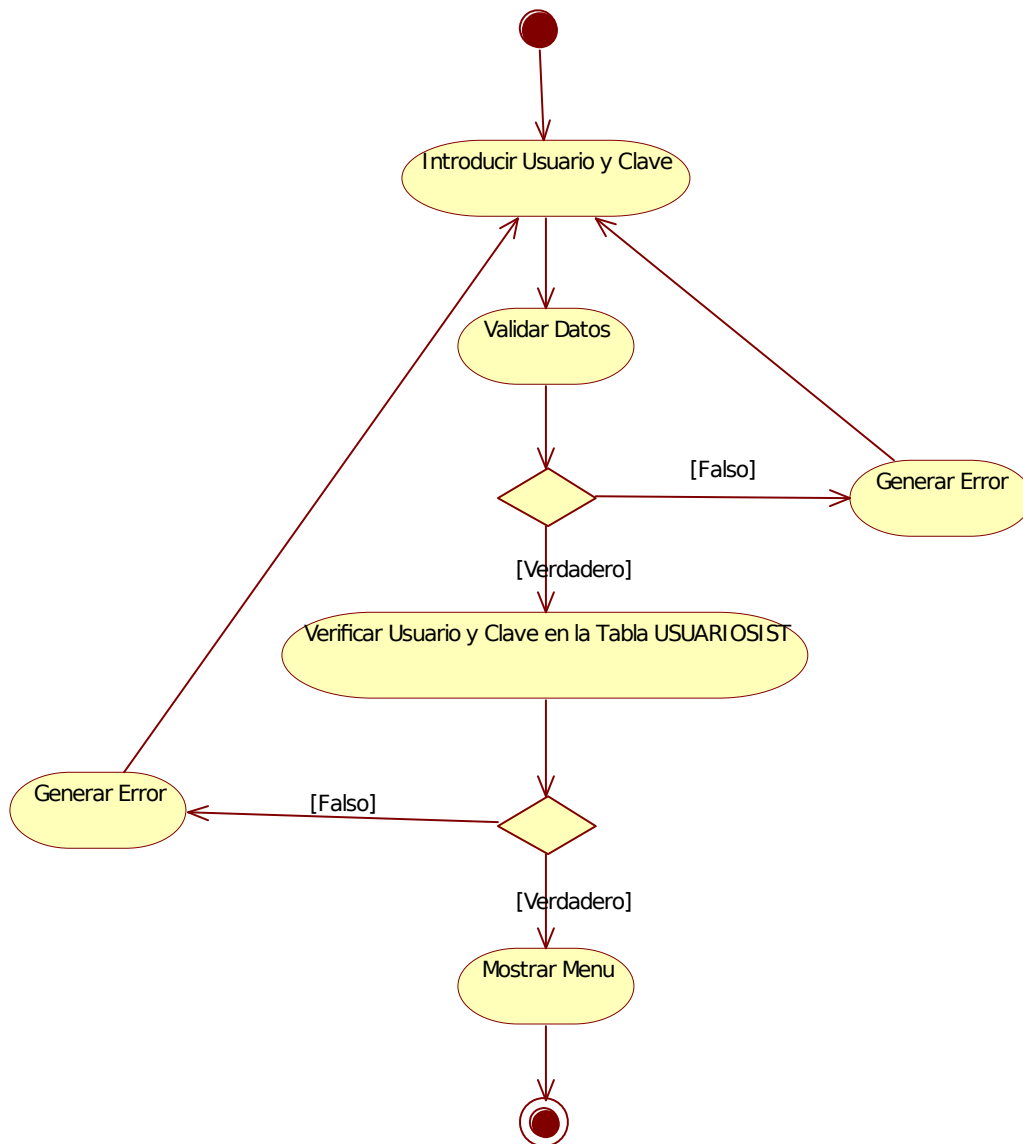


4.25.

4.26.

5. DIAGRAMAS DE ACTIVIDADES

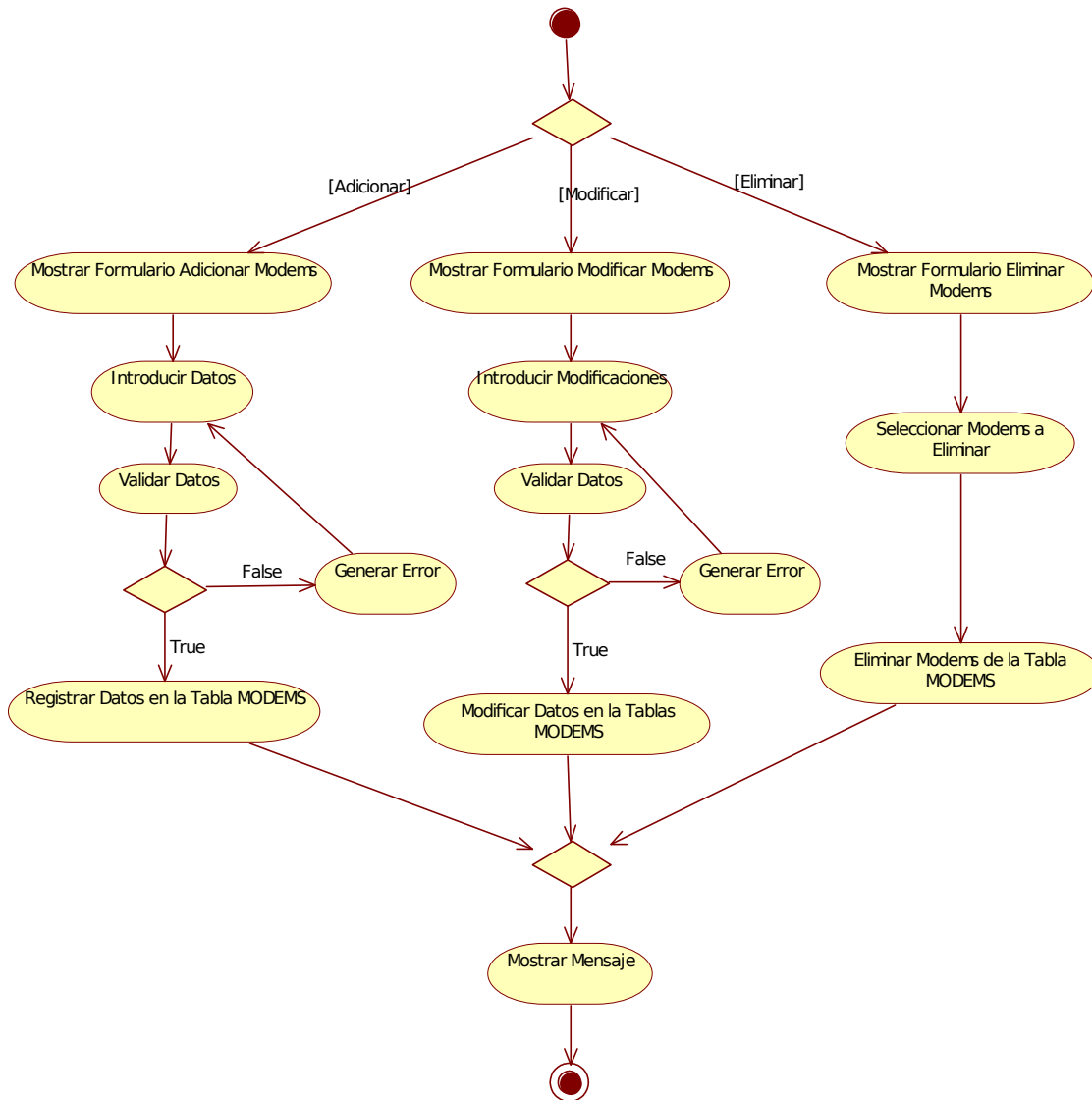
5.1. Diagrama de Actividades del Caso de Uso Conectarse



5.2.

5.3.

5.4. Diagrama de Actividades del Caso de Uso Administrar Modem



5.5.

5.6.

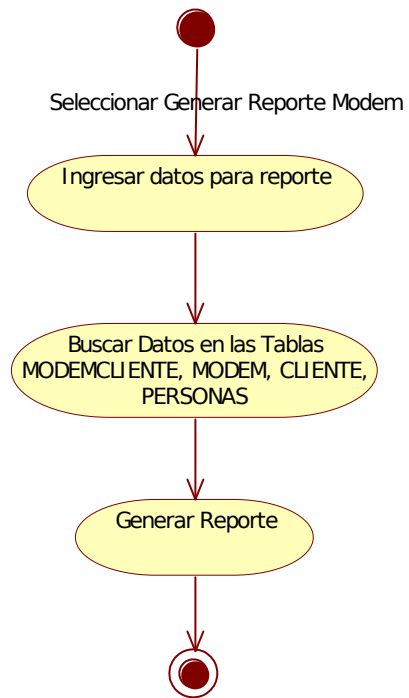
5.7. Diagrama de Actividades del Caso de Uso Administrar Destino Modem



5.8.

5.9.

5.10. Diagrama de Actividades del Caso de Uso Generar Reporte Módem

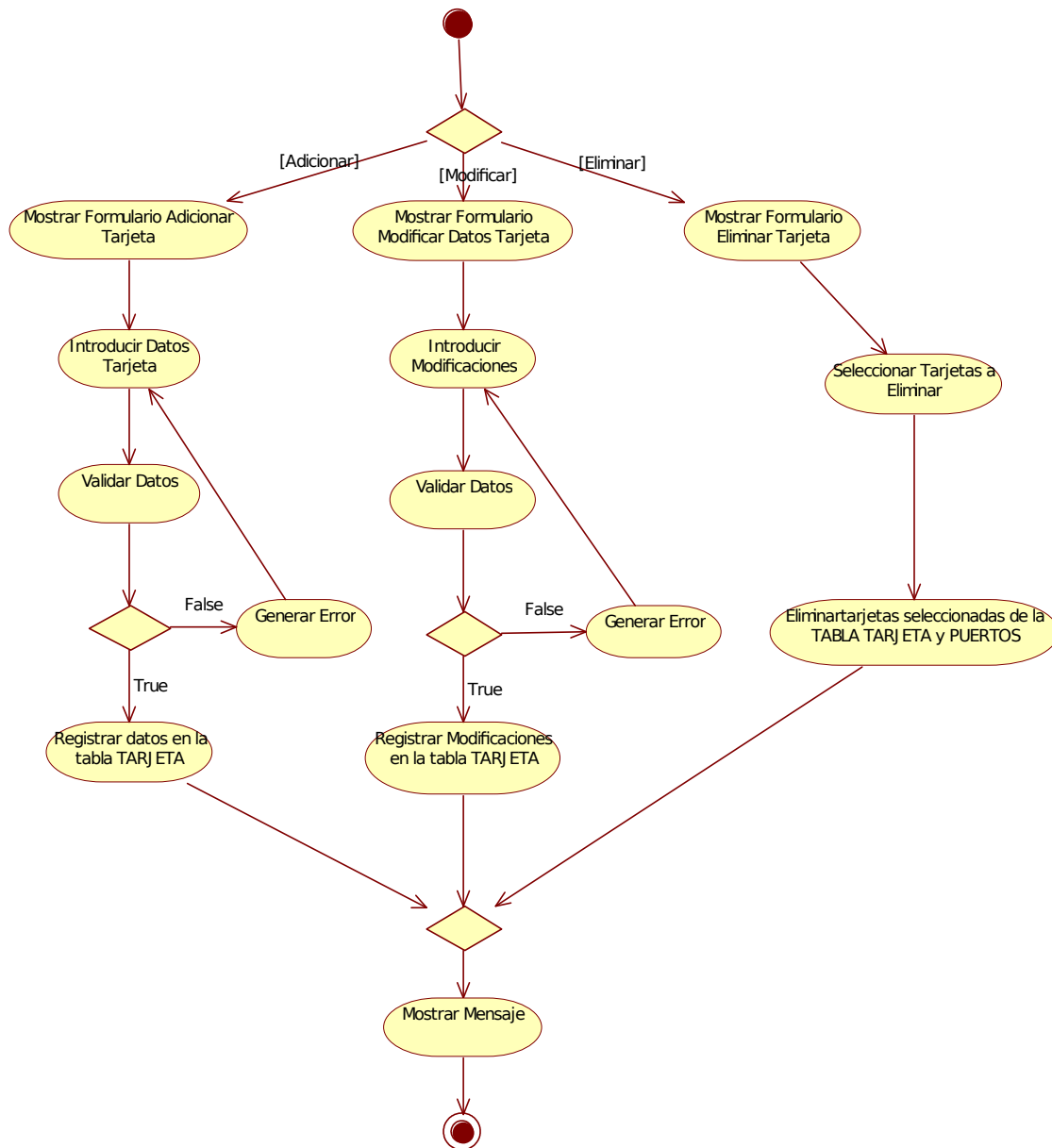


5.11.

5.12.

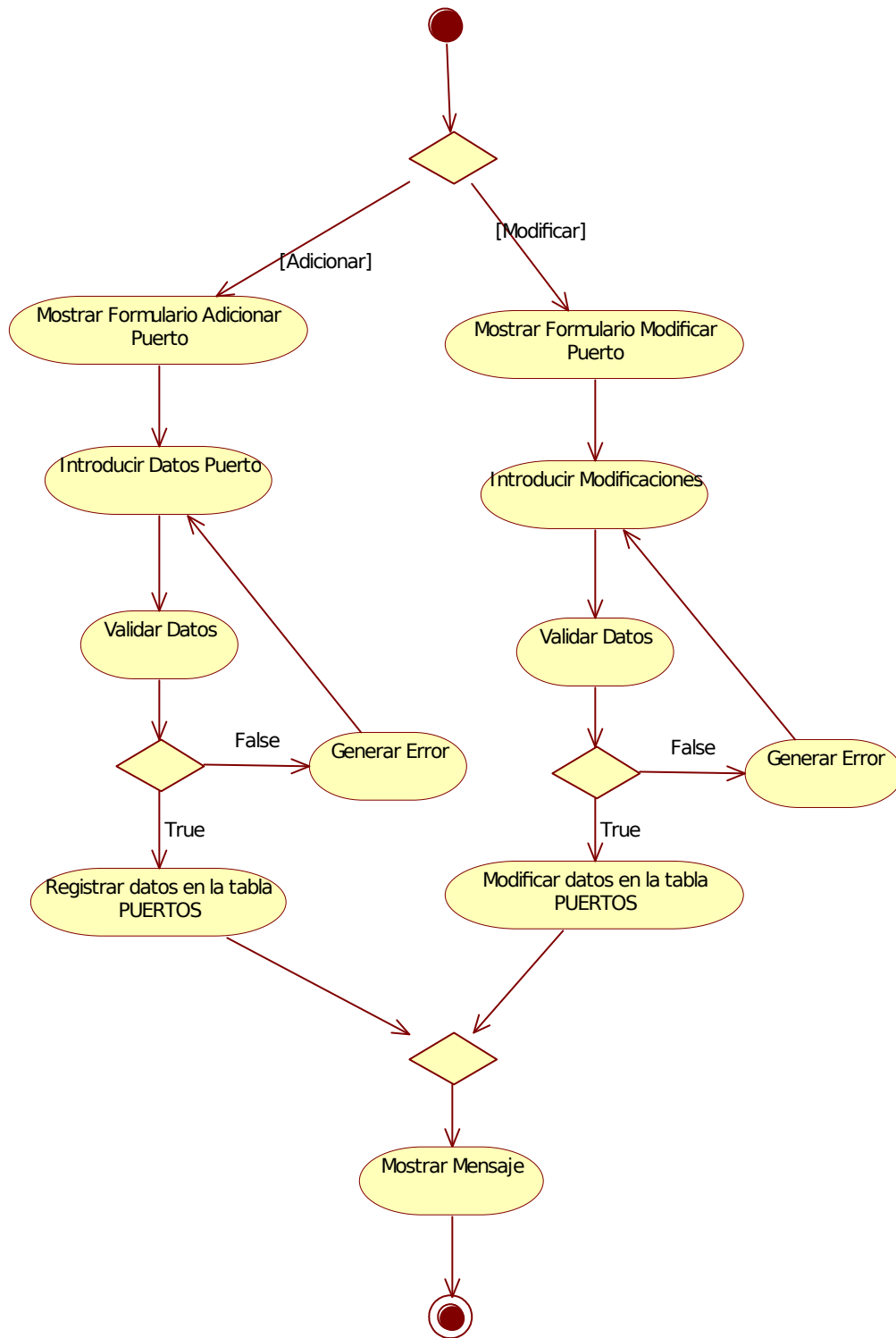
5.13. Diagrama de Actividades del Caso de Uso Administrar Tarjetas

5.14.



5.15.

5.16. Diagrama de Actividades del Caso de Uso Administrar Puertos



5.17.

5.18.

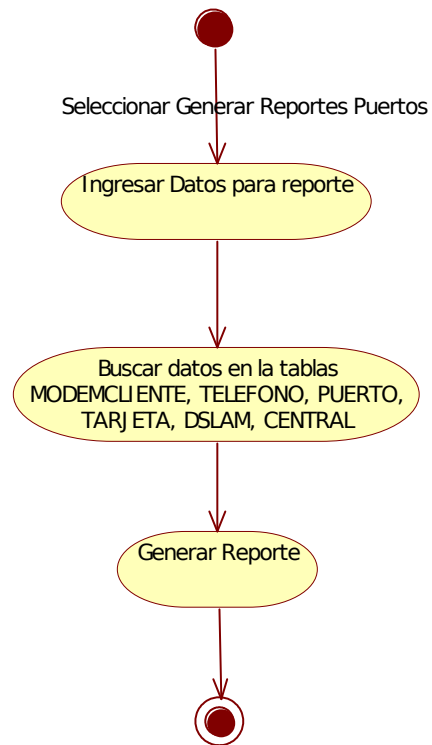
5.19. Diagrama de Actividades del Caso de Uso Administrar Destino Puerto

5.20.



5.21.

5.22. Diagrama de Actividades del Caso de Uso Generar Reporte Puertos



5.23.

5.24.

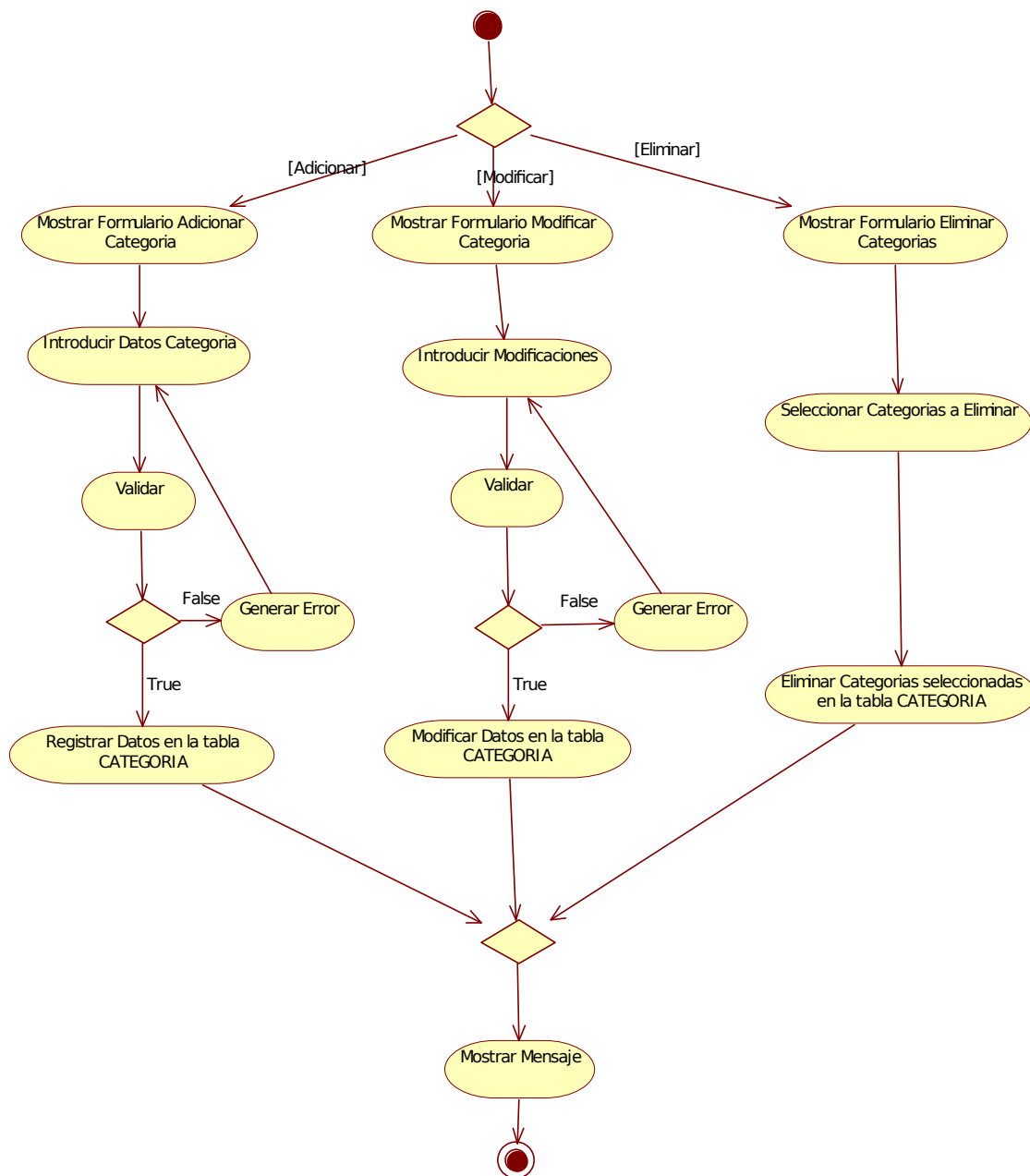
5.25. Diagrama de Actividades del Caso de Uso Administrar Usuario
5.26.



5.27.

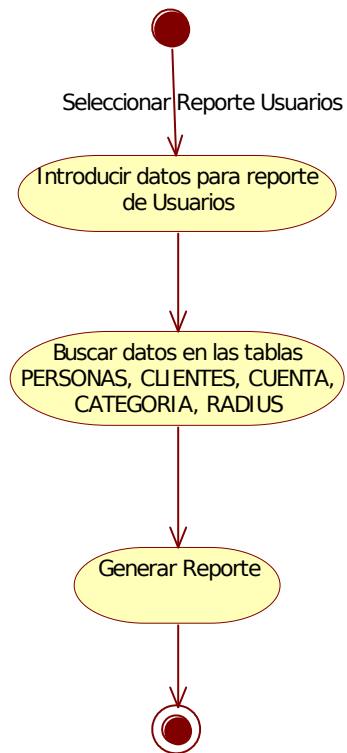
5.28. Diagrama de Actividades del Caso de Uso Administrar Categoría

5.29.



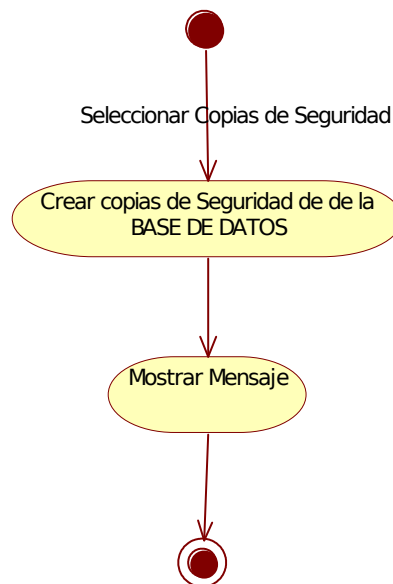
5.30.

5.31. Diagrama de Actividades del Caso de Uso Generar Reportes Usuario



5.32.

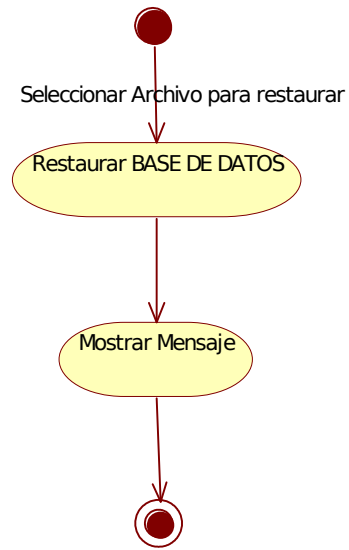
5.33. Diagrama de Actividades del Caso de Uso Realizar Copia Seguridad



5.34.

5.35.

5.36. Diagrama de Actividades del Caso de Uso Restaurar Base de Datos



5.37.

El principal resultado del diseño es el modelo de diseño que se esfuerza en conservar la estructura del sistema impuesta por el modelo de análisis y que sirve como esquema para la implementación.

5.1 DESCRIPCIÓN DE CLASES

Nomenclatura utilizada

_____	Las clases desarrolladas por el sistema son representadas por una línea continua.
-----	Las clases desarrolladas por otros sistemas son representadas por una línea discontinua.
<i>Método()</i>	Los métodos empleados por el sistema son representados con fuente cursiva
Método()	Los métodos empleados por otros sistemas son representados con fuente normal

Nombre de la Clase: MODEM
Descripción: representa al módem para conectarse a internet.
Atributos: Serie Varchar(25) not null Marca Varchar(25) Modelo Varchar(25) Estado Varchar (25)
Operaciones: Adicionar() Modificar() DardeBaja() ListarModems() AdicionarDestino() ModificarDestino() EliminarDestino
Nombre de la Clase: USUARIOSIST
Descripción: respresenta al usuario del sistema
Atributos: CI Varchar(10) not null Nombre Varchar(25) not null Paterno Varchar(25) not null Materno Varchar(25) Cargo Varchar(25)

Operaciones: <i>Buscar()</i> Adicionar() Modificar() ObtenerDatosPersonales() <i>ObtenerNombre()</i>
Nombre de la Clase: LOGIN
Descripción: representa los datos del usuario que accede al sistema
Atributos: NomU Varchar(15) CI Varchar(10) not null Contrasena Varchar(20)
Operaciones: Asignar() <i>Buscar()</i> <i>Obtener()</i> <i>Verificar()</i>

Nombre de la Clase: CLIENTE
Descripción: representa un cliente del servicio de internet
Atributos: Id Integer not null CI Integer not null NombCli Varchar(5) not null ApePat Varchar(25) not null ApeMat Varchar(20) not null Direccion Varchar(20) not null Genero Varchar(1) Ocupación Varchar(25)
Operaciones: Adicionar() Modificar() DardeBaja() <i>Buscar()</i> <i>Listar()</i>

Nombre de la Clase: CUENTA
Descripción: representa a la cuenta con la que el usuario se conecta al internet

<p>Atributos: Name Varchar(8) not null Password Varchar(20) not null Service Varchar (20) CallerId Varchar (20)</p>
<p>Operaciones: <i>Modificar()</i> <i>Buscar()</i> <i>Adicionar()</i> <i>DarDeBaja()</i></p>

<p>Nombre de la Clase: TELEFONO</p>
<p>Descripción: representa una línea telefónica perteneciente a un usuario</p>
<p>Atributos: Num Integer not null Categoria Varchar (20) Estado Varchar (20) FechaInstalacion Date not null</p>
<p>Operaciones: <i>Modificar()</i> <i>Buscar()</i> <i>Adicionar()</i> <i>listar()</i></p>

<p>Nombre de la Clase: CATEGORIA</p>
<p>Descripción: representa un plan ADSL de Internet</p>
<p>Atributos: Id Integer not null NombCat Varchar(20) not null LocalAdress Varchar (20) RemoteAdress Varchar (20) AddressList Varchar (20) RateLimitRx Integer not null RateLimitTx Integer not null</p>
<p>Operaciones: Adicionar()</p>

Modificar() Eliminar() Listar() <i>Buscar()</i>
--

Nombre de la Clase: RADIUS

Descripción: representa un servidor donde las cuentas autentifican

Atributos:

CodRad Integer not null NombreRad Varchar (25) DirecIp Varchar (25) Firmware Varchar (25) Marca Varchar (25) Modelo Varchar (25) FechaInstalacion Date not null Descripción Varchar(50)
--

Operaciones:

Adicionar() Modificar() Eliminar()
--

Nombre de la Clase: CENTRAL

Descripción: representa al MDF de la zona
--

Atributos:

CodCen Varchar(5) not null NombCen Varchar (25) not null Direccion Varchar (25) Descripcion Varchar(200)

Operaciones:

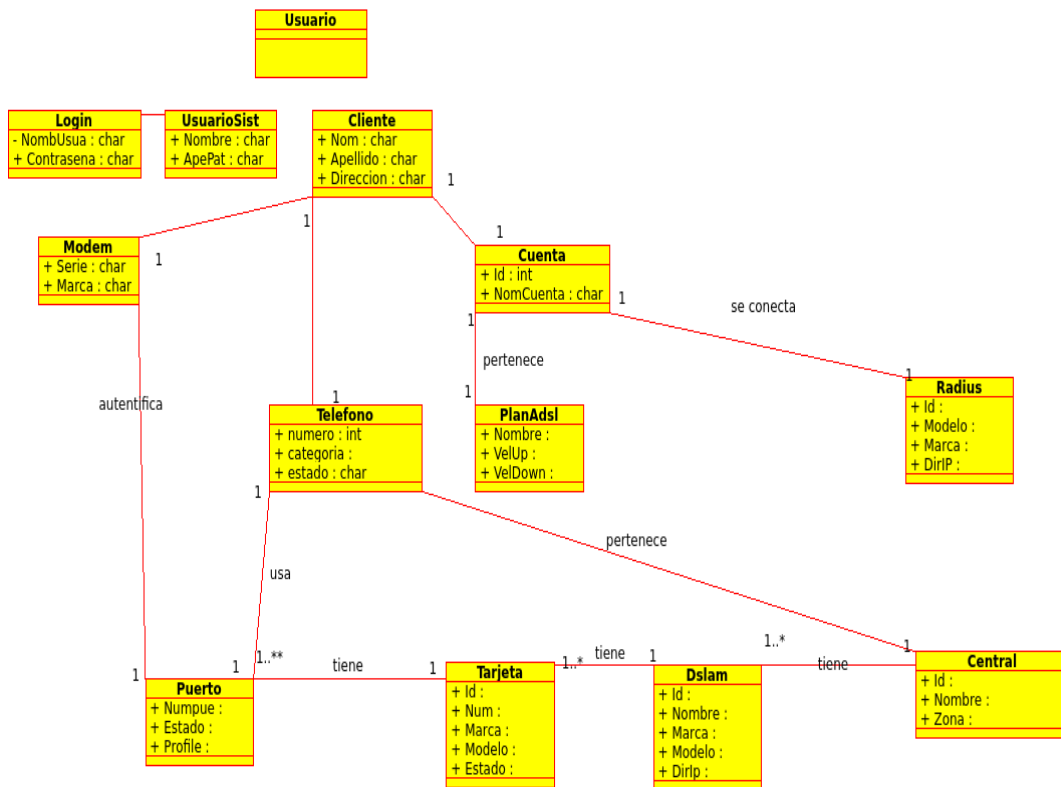
Adicionar() Modificar() Listar()
--

Nombre de la Clase: DSLAM
Descripción: representa el equipo que da señal DSL a la línea telefónica
Atributos: CodDslam Varchar(5) not null NombDslam Varchar (25) not null IpDslam Varchar (25) not null Marca Varchar (25) Modelo Varchar (25) Firmwareslam Varchar (25) CantidadTarje Integer FechaInstalacion Date not null
Operaciones: Adicionar() Modificar() CalcularDisponibilidad()

Nombre de la Clase: TARJETA
Descripción: representa la tarjeta que se inserta en el DSLAM
Atributos: IdTarjeta Varchar (5) not null Marca Varchar (25) Modelo Varchar (25) FirmwareTja Varchar (25) CantidadTarje Integer NomCopia Varchar(2000) not null FechaInstalacion Date not null CantidadPuertos Integer
Operaciones: Adicionar() Modificar() DardeBaja() Listar() CalcularDis()

Nombre de la Clase: PUERTO
Descripción: representa el puerto en una tarjeta disponible para dar ADSL
Atributos: IdPuerto varchar (5) not null numeropuerto Integer Profile Varchar (25) Vlan Integer AdminStatus Varchar (7) MaxMacLear Integer Lastlinkup date not null Estado Varchar (25)
Operaciones: Adicionar() Modificar() Eliminar() <i>Buscar()</i>

5.2 DIAGRAMA DE CLASES



5.3 DICCIONARIO DE DATOS

LP: llave primaria

LF: llave foránea

Tabla MODEM

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
1	CodAct	Varchar(25)	Si	No	Código de la actividad	
2	NombreAct	Varchar(250)	No	No	Nombre de actividad	
3	Avance	Integer	No	No	Porcentaje de avance de la actividad	

Tabla ADJUNTOS

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
4	CodPro	Varchar(20)	Si	No	Código del proyecto	
5	Articulo	Varchar(100)	No	No	Nombre del artículo	
6	Proyectof	Varchar(200)	No	No	Nombre del proyecto	

Tabla ALERTAS

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
7	Id	Integer	Si	No	Código de la alerta	
8	CodPro	Varchar(20)	No	Si	Código de proyecto	Proyecto(CodPro)
9	CodAct	Varchar(25)	No	No	Código de actividad	
10	TipoInf	Varchar(30)	No	No	Tipo de informe	

Tabla CARGO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
11	CodCar	Integer	No	No	Código del cargo	
12	NombreC	Varchar(80)	Si	No	Nombre de cargo	

Tabla CATEGORIAA

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
13	Id	Integer	Si	No	Identificador de categoría	
14	CodPro	Varchar(20)	Si	Si	Código de proyecto	Proyecto(CodPro)
15	CodAct	Varchar(25)	Si	Si	Código de la actividad	Actividad (CodAct)
16	CodItem	Varchar(5)	Si	Si	Código del ítem	Item(CodItem)
17	NombreEta	Varchar(20)	Si	Si	Código de etapa	Etapa (NombreEta)
18	CodMay	Varchar(5)	Si	Si	Código de mayor	Mayor(CodMay)
19	Glosa	Varchar(100)	No	No	Descripción	
20	Numero	Integer	No	No	Número de reg.	
21	Remuneracion/Mes	Float	No	No	Cantidad remunerada por mes	
22	Tiempomes	Varchar(15)	No	No	Tiempo de trabajo al mes	
23	TotalU	Float	No	No	Monto financiado por la universidad	
24	TotalF	Float	No	No	Monto financiado por otra entidad	

Tabla CATEGORIAB

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
25	Id	Integer	Si	No	Identificador de categoría	
26	CodPro	Varchar(20)	Si	Si	Código de	Proyecto(CodPro)

					proyecto	
27	CodAct	Varchar(25)	Si	Si	Código de la actividad	Actividad (CodAct)
28	CodItem	Varchar(5)	Si	Si	Código del ítem	Item (CodItem)
29	NombreEta	Varchar(20)	Si	Si	Código de etapa	Etapa (NombreEtapa)
30	CodMay	Varchar(5)	Si	Si	Código de mayor	Mayor(CodMayor)
31	Glosa	Varchar(100)	No	No	Descripción	
32	Numero	Integer	No	No	Número de reg.	
33	Cantidad	Float	No	No	Cantidad	
34	TotalU	Float	No	No	Total financiado por la universidad	
35	TotalF	Float	No	No	Total financiado por otra entidad	

Tabla CATEGORIAC

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
36	Id	Integer	Si	No	Identificador de categoría	
37	CodPro	Varchar(20)	Si	Si	Código de proyecto	Proyecto(CodPro)
38	CodAct	Varchar(25)	Si	Si	Código de la actividad	Actividad (CodAct)
39	CodItem	Varchar(5)	Si	Si	Código del ítem	Item(CodItem)
40	NombreEta	Varchar(20)	Si	Si	Código de etapa	Etapa(NombreEta)
41	CodMay	Varchar(5)	Si	Si	Código de mayor	Mayor(CodMay)
42	Glosa	Varchar(100)	No	No	Descripción	
43	Numero	Integer	No	No	Número de reg.	
44	Lugar	Varchar(50)	No	No	Destino de viaje	
45	Duración	Varchar(15)	No	No	Tiempo duración del viaje en días	
46	Costo	Float	No	No	Monto necesario para el viaje	

47	Viáticos	Float	No	No	Monto destinado para viáticos	
48	TotalU	Float	No	No	Total financiado por la universidad	
49	TotalF	Float	No	No	Total financiado por otra entidad	

Tabla CATEGORIAD

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
50	Id	Integer	Si	No	Identificador de categoría	
51	CodPro	Varchar(20)	Si	Si	Código de proyecto	Proyecto(CodPro)
52	CodAct	Varchar(25)	Si	Si	Código de la actividad	Actividad (CodAct)
53	CodItem	Varchar(5)	Si	Si	Código del ítem	Item(Coditem)
54	NombreEta	Varchar(20)	Si	Si	Código de etapa	Etapa(NombreEta)
55	CodMay	Varchar(5)	Si	Si	Código de mayor	Mayor(CodMay)
56	Glosa	Varchar(100)	No	No	Descripción	
57	Numero	Integer	No	No	Número de reg	
58	Cantidad	Float	No	No	Cantidad de producto	
59	TotalU	Float	No	No	Total financiado por la universidad	
60	TotalF	Float	No	No	Total financiado por otra entidad	
61	Costo	Float	No	No	Costo	

Tabla CAUSAS

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
62	CodCau	Integer	Si	No	Identificador de causa	
63	Descripción	Varchar(50)	No	No	Descripción de la causa	Proyecto(CodPro)

Tabla COMENTARIO

Nro	Nombre del	Tipo de dato	LP	LF	Descripción	Tabla y campo
-----	------------	--------------	----	----	-------------	---------------

	campo					referente
64	CodCom	Varchar(5)	Si	No	Código del comentario	
65	CodPre	Varchar(5)	No	Si	Código de pregunta	Pregunta(CodPre)
66	Descripcion	Varchar(2000)	No	No	Comentario a la pregunta del momento	
67	Fecha	Date	No	No	Fecha del comentario	
68	Autor	Varchar(10)	No	No	Identificador del usuario	

Tabla CONTADOR

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
69	CodPre	Varchar(5)	No	Si	Código de la pregunta	Pregunta(CodPre)
70	CodRes	Varchar(5)	No	Si	Código de la respuesta	Respuesta (CodRes)
71	Cantidad	Integer	No	No	Número de respuesta del usuario	

Tabla COPIAS

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
72	NomCopia	Varchar(2000)	Si	No	Nombre de la Copia	
73	Fecha	Date	Si	No	Fecha en que se hace la copia	Proyecto(CodPro)

Tabla CUA

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
74	NombreC	Varchar(80)	Si	Si	Nombre de cargo	Cargo(NombreC)
75	CodUni	Varchar(50)	Si	Si	Codigo de unidad academica	UAcademica (CodUni)
76	Ci Ci	Varchar(10)	Si	Si	Cedula de identidad	Ususistem(Ci)

Tabla CUENTA

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
77	NomU	Varchar(15)	No	No	Nombre de Usuario	
78	Contrasena	Varchar(20)	No	No	Clave de la cuenta	

Tabla CURRICULUM

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
79	Ci	Varchar(10)	No	Si	Cedula de identidad del usuario	Ususistem(Ci)
80	EstudiosR	Varchar(3000)	No	No	Títulos obtenidos	
81	Idiomas	Varchar(3000)	No	No	Idiomas que habla	
82	Becas	Varchar(3000)	No	No	Tipo, tema	
83	Antinvest	Varchar(3000)	No	No	Antecedentes	
84	Preproy	Varchar(3000)	No	No	Presentación de proyectos en otros años	
85	Antdoc	Varchar(3000)	No	No	Antecedentes en docencia	
86	Subsidios	Varchar(3000)	No	No	Subsidios conseguidos	
87	Cargos	Varchar(3000)	No	No	Cargos desempeñados	
88	Publicaciones	Varchar(3000)	No	No	Publicaciones realizadas	
89	Presentaciones	Varchar(3000)	No	No	Publicaciones recientes	
90	Patentes	Varchar(3000)	No	No	Realizadas y obtenidas	
91	Actuaciones	Varchar(3000)	No	No	Actuaciones en sociedad científica	
92	Premios	Varchar(3000)	No	No	Premios obtenidos	

Tabla DESEMBOLSO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
------------	-------------------------	---------------------	-----------	-----------	--------------------	--------------------------------

93	Id	Integer	Si	No	Identificador	
94	CodPro	Varchar(20)	Si	Si	Código de proyecto	Proyecto(CodPro)
95	NombreEta	Varchar(20)	Si	Si	Código de etaproy	Etapa(NombreEta)
96	Moneda	Varchar(10)	No	No	Tipo de moneda	
97	Tipo	Varchar(1)	No	No	0=cheque 1=efectivo	
98	Cheque	Varchar(20)	No	No	Nro. de cheque o comprobante	
99	Monto	Float	No	No	Monto desembolsado	
100	Saldo	Float	No	No	Monto restante para desembolsar	
101	Fecha	Date	No	No	Fecha en que se desarrolló el desembolso	
102	FechaAct	Date	No	No	Fecha actual	

Tabla ESTACTIVIDAD

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
103	CodAct	Varchar(25)	Si	Si	Código de actividad	Actividad (CodAct)
104	NombreEs	Varchar(30)	Si	Si	Código de estadoact	Estadoact (NombreEs)
105	Estadoanterior	Varchar(30)	Si	No	El estado anterior de la actividad	
106	Fechaest	Date	No	No	Fecha en la que se guarda el estado de la actividad	
107	TipoPro	Varchar(15)	No	No	Tipo de proyecto (investigación o extensión)	
108	Bandera	Boolean	No	No	True=estado actual	

Tabla ESTADOACT

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
109	NombreEs	Varchar(30)	Si	No	Nombre estado de la actividad	
110	Alias	Varchar(15)	No	No	Otro nombre del estado de la actividad	
111	AfectaPro	Boolean	No	No	Determina si afecta al proyecto	
112	Sw	Char(1)	No	No	1=activado 0=desactivado	
113	Anexo	Boolean	No	No	True=Tiene anexo False=No tiene anexo	
114	TipoPro	Varchar(15)	Si	No	Tipo de proyecto (investigación/extensión)	

TABLA ESTADOPRO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
115	NombreEs	Varchar(30)	Si	No	Nombre del estado del proyecto	
116	Tipopro	Varchar(15)	Si	No	Tipo de proyecto (investigación/extensión)	
117	Alias	Varchar(15)	No	No	Sobrenombre del estado del proyecto	
118	Sw	Char (1)	No	No	1=activado 0=desactivado	
119	AfectaAct	Boolean	No	No	Determina si el estado afecta a las actividades	
120	Anexo	Boolean	No	No	True=Si tiene anexo False=no tiene anexo	

Tabla ETAPA

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
121	Id	Integer	No	No	Identificación de la etapa	
122	NombreEta	Varchar(20)	Si	No	Nombre de la etapa	

Tabla ETAPROEST

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
123	NombreEta	Varchar(20)	Si	Si	Nombre de la etapa	Etapa (NombreEta)
124	NombreEs	Varchar(30)	Si	Si	Nombre de estado del proyecto	Estadoact (NombreEs)
125	CodPro	Varchar(20)	Si	Si	Código del proyecto	Proyecto (NombrePro)
126	Fecha	Date	Si	No	Fecha de cambio de etapa	
127	Tipopro	Varchar(15)	Si	No	Tipo de proyecto (investigación /extensión)	
128	Estadoanterior	Varchar(30)	No	No	Nombre de estado anterior del proyecto	
129	Bandera	Boolean	No	No	True=estado activo	
130	Ctrl	Integer	No	No	Control	

Tabla ETAPROY

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
131	NombreEta	Varchar(20)	Si	Si	Nombre de la etapa	Etapa (NombreEta)
132	CodPro	Varchar(20)	Si	Si	Código del proyecto	Proyecto(CodPro)
133	CostoT	Float	No	No	Costo total	
134	FechaIni	Date	No	No	Fecha de inicio	
135	FechaFin	Date	No	No	Fecha del fin	

136	FechaE	Date	No	No	Fecha cambio de etapa	
137	Hora	Time	No	No	Hora cambio de etapa	

Tabla ETAPROYACT

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
138	CodPro	Varchar(20)	Si	Si	Código de proyecto	Proyecto(CodPro)
139	CodAct	Varchar(25)	Si	Si	Código de la actividad	Actividad (CodAct)
140	FechaIniA	Date	No	No	Fecha de inicio de actividad	
141	FechaFinA	Date	No	No	Fecha de fin de actividad	
142	cirep	Varchar(10)	No	No	Ci del Responsable	
143	NombreEta	Varchar(20)	Si	Si	Código de etapa	Etapa(NombreEta)
144	Total	Float	No	No	Monto asignado a la actividad	
145	Saldo	Float	No	No	Monto pendiente a cancelar por la actividad	
146	Orden	Integer	No	No	Orden de las actividades	

Tabla EVALUACIONACT

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
147	TipoPEA	Varchar(30)	Si	Si	Código de plevact	Plevact(TipoPEA)
148	NombrePEA	Varchar(100)	Si	Si	Código de plevact	Plevact(NombrePEA)
149	CodAct	Varchar(25)	Si	Si	Código de la actividad	Actividad (CodAct)
150	Descripcion	Varchar(20000)	No	No	Contenido de la evaluación del campo	
151	Fecha	Date	No	No	Fecha en que presentó la evaluación	
152	Ban	Char(1)	No	No	Identifica si el	

					registro puede modificarse	
153	TipoPro	Varchar(15)	Si	Si	Tipo de proyecto (investigación /extensión)	Plevact(TipoPro)
154	Padre	Integer	No	No	determina el campo tiene padre	

Tabla EVALUACIONPRO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
155	CodPro	Varchar(20)	Si	Si	Código de proyecto	Proyecto(CodPro)
156	NombrePEP	Varchar(100)	Si	Si	Código de plevpro	Plevpro(NombrePEP)
157	TipoPEP	Varchar(30)	Si	Si	Código de plevpro	Plevpro(TipoPEP)
158	TipoPro	Varchar(15)	Si	Si	Tipo de proyecto (investigación /extensión)	Plevpro(TipoPro)
159	Descripcion	Varchar(20000)	No	No	Observación de evaluación	
160	nro	Integer	No	No	Número de informe parcial	
161	Fecha	Date	No	No	Fecha de la evaluación	
162	Ban	Char(1)	No	No	Bandera	
163	Padre	Integer	No	No	Determina si el campo tiene padre	

Tabla GACADEMICO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
164	Idga	Integer	Si	No	Clave de la tabla gacademico	
165	nombrega	Varchar(80)	No	Si	Nombre del grado académico	
166	Abreviado	Varchar(10)	No	No	Abreviatura	

Tabla INFORCAUSAS

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
-----	------------------	--------------	----	----	-------------	-------------------------

167	CodCau	Integer	No	Si	Código de causas	Causas(CodCau)
168	CodPro	Varchar(20)	No	Si	Código de proyecto	Proyecto(CodPro)
169	NombreEta	Varchar(20)	No	Si	Nombre de la etapa	Etapa(NombreEta)
170	Observacion	Varchar(20000)	No	No	Descripción de la solicitud	

Tabla INFORMEACT

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
171	NombrePIA	Varchar(100)	Si	Si	Nombre del campo de informe de actividad	Plifact (NombrePIA)
172	CodAct	Varchar(25)	Si	Si	Código de la actividad	Actividad (CodAct)
173	TipoPIA	Varchar(30)	Si	Si	Tipo de informe	Plifact(TipoPia)
174	Tipopro	Varchar(15)	Si	Si	Tipo de proyecto (investigación/extensión)	Plifact(TipoPro)
175	Descripción	Varchar(20000)	No	No	Contenido del informe del campo	
176	Fecha	Date	No	No	Fecha en que presentó el informe	
177	Ban	Char(1)	No	No	Identifica si el registro puede modificarse o no	
178	Padre	Integer	No	No	Determina si campo tiene padre	

Tabla INFORMEPRO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
179	CodPro	Varchar(20)	Si	Si	Código de proyecto	Proyecto(CodPro)
180	TipoPIP	Varchar(30)	Si	Si	Código de plinfpro	Plinfpro(TipoPIP)
181	NombrePIP	Varchar(100)	Si	Si	Código de	Plinfpro

					plinfpro	(NomnrePIP)
182	Tipopro	Varchar(15)	Si	Si	Tipo proyecto (investigación/extensión)	PlinfPro(TipoPro)
183	Fecha	Date	No	No	Fecha del informe	
184	Descripcion	Varchar(20000)	No	No	Informe del proyecto	
185	Ban	Char(1)	No	No	Bandera	
186	Padre	Integer	No	No	Determina si el campo tiene padre	
187	Nro	Integer	No	No	Número de informe	

Tabla ITEM

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
188	CodItem	Varchar(5)	Si	No	Código del item	
189	CodMay	Varchar(5)	Si	Si	Código de mayor	Mayor(CodMay)
190	Alias	Varchar(50)	No	No	Nombre detalle del item	
191	NombreI	Varchar(50)	No	No	Nombre del item	
192	Categorial	Varchar(1)	No	No	Categoría a la que pertenece	
193	Sw	Boolean	No	No	1=Activo 0=Inactivo	

Tabla MAYOR

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
194	CodMay	Varchar(5)	Si	No	Código de mayor	
195	Nombre	Varchar(50)	No	No	Nombre detalle de mayor	
196	Sw	Boolean	No	No	1=Activo 0=Inactivo	

Tabla MENSAJES

Nro	Nombre del	Tipo de dato	LP	LF	Descripción	Tabla y campo
-----	------------	--------------	----	----	-------------	---------------

	campo					referente
197	CodMen	Integer	Si	No	Código del mensaje	
198	Descripcion	Varchar(2000)	No	No	Mensaje del usuario	
199	Asunto	Varchar(50)	No	No	Descripción del asunto	
200	FechaM	Date	No	No	Fecha del mensaje	
201	Padre	Integer	No	No	0=Padre CodMen=Hijo	
202	Autor	Varchar(10)	No	No	CI del usuario	
203	CodTem	Varchar(5)	No	Si	Código del tema	Tema(CodTem)

Tabla PLEVACT

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
204	CodPEA	Integer	No	No	Identificador de plantilla	
205	NombrePEA	Varchar(100)	Si	No	Nombre del campo de la evaluación de actividad	
206	TipoPEA	Varchar(30)	Si	No	Tipo de evaluación	
207	CodObj	Integer	No	No	Nombre del objeto	
208	Tipopro	Varchar(15)	Si	No	Tipo de proyecto (investigación/extensión)	
209	Descripcion	Varchar(1000)	No	No	Descripción de evaluación	
210	Obligado	Boolean	No	No	Campo obligatorio o no	
211	Sw	Char(1)	No	No	1=activado 0=desactivado	

212	Padre	Integer	No	No	Determina si el campo tiene padre	
-----	-------	---------	----	----	-----------------------------------	--

Tabla PLEVPRO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
213	CodPEP	Integer	No	No	Código de plevpro	
214	TipoPEP	Varchar(30)	Si	No	Código de plevpro	
215	NombrePEP	Varchar(100)	Si	No	Nombre de plevpro	
216	TipoPro	Varchar(15)	Si	No	Tipo de proyecto (investigación/extensión)	
217	CodObj	Integer	No	No	Nombre de objeto	
218	Descripción	Varchar(1000)	No	No	Descripción de evaluacion	
219	Obligado	Boolean	No	No	Campo obligatorio o no	
220	Sw	Char(1)	No	No	1=activado 0=desactivado	
221	Padre	Integer	No	No	Determina si el campo tiene padre	

Tabla PLINFACT

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
222	CodPIA	Integer	No	No	Identificador de plantilla	
223	NombrePIA	Varchar(100)	Si	No	Nombre del campo de informe	
224	TipoPIA	Varchar(30)	Si	No	Tipo de informe	
225	Tipopro	Varchar(15)	Si	No	Tipo de proyecto (investigación/extensión)	
226	CodObj	Integer	No	No	Nombre del obj.	
227	Descripcion	Varchar(1000)	No	No	Descripción del campo de informe de la actividad	
228	Obligado	Boolean	No	No	Campo obligatorio	

					o no	
229	Sw	Char (1)	No	No	1=activado 0=desactivado	
230	Padre	Integer	No	No	Determina si el campo tiene padre	

Tabla PLINFPRO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
231	CodPIP	Integer	No	No	Identificador de plantilla proyecto	
232	TipoPIP	Varchar(30)	Si	No	Tipo de informe	
233	Tipopro	Varchar(15)	Si	No	Tipo de proyecto (investigación/extensión)	
234	NombrePIP	Varchar(100)	Si	No	Nombre del campo de informe	
235	CodObj	Integer	Si	No	Nombre del obj.	
236	Descripción	Varchar(1000)	No	No	Descripción del campo de informe del proyecto	
237	Obligado	Boolean	No	No	Campo obligatorio o no	
238	Sw	Char(1)	No	No	1=activado 0=desactivado	
239	Padre	Integer	No	No	Determina si el campo tiene padre	

Tabla PRECEDENCIA

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
240	CodPro	Varchar(20)	Si	Si	Código de proyecto	Proyecto(CodPro)
241	CodAct	Varchar(25)	Si	Si	Código de actividad	Actividad (CodAct)
242	CodPre	Varchar(25)	Si	No	Código de precedencia	
243	NombreEta	Varchar(20)	Si	Si	Código de etapa	Etapa(NombreEta)

Tabla PREGUNTA

Nro	Nombre del	Tipo de dato	LP	LF	Descripción	Tabla y campo
------------	-------------------	---------------------	-----------	-----------	--------------------	----------------------

	campo					referente
244	CodPre	Varchar(5)	Si	No	Código de la pregunta	
245	Pregunta	Varchar(1000)	No	No	Pregunta del momento	

Tabla PRESUPUESTO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
246	CodPro	Varchar(20)	Si	Si	Código del proyecto	Proyecto(CodPro)
247	CodAct	Varchar(25)	Si	Si	Código de la actividad	Actividad (CodAct)
248	CodItem	Varchar(5)	Si	Si	Código del item	Item(CodItem)
249	NombreEta	Varchar(20)	Si	Si	Nombre de etapa	Etapa(NombreEta)
250	Codmay	Varchar(5)	Si	Si	Código de mayor	Mayor(CodMay)
251	TotalU	Float	No	No	Monto financiado por la universidad	
252	TotalF	Float	No	No	Monto financiado por otra entidad	
253	Total	Float	No	No	Monto total asignado a la actividad	

Tabla PROYECTO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
254	CodPro	Varchar(20)	Si	No	Código del proyecto	
255	TipoPro	Varchar(15)	No	No	Tipo de proyecto (investigación/extensión)	
256	NombreP	Varchar(200)	No	No	Nombre del proyecto	

257	Avance	Integer	No	No	Porcentaje de avance de proyecto	
258	Nombrel	Varchar(100)	No	Si	Código de lineainvest	LineaInvest (Nombrel)

Tabla RESPUESTA

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
259	CodRes	Varchar(5)	Si	No	Código de la respuesta	
260	Respuesta	Varchar(20)	No	No	Respuestas que podría tener la pregunta	

Tabla ROL

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
261	NombreR	Varchar(20)	Si	No	Nombre del ROL (código)	Causas(CodCau)
262	TipoR	Varchar(1)	No	No	El tipo de rol que tiene el usuario del sistema	Proyecto(CodPro)

Tabla SITUACION

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
263	CodAct	Varchar(25)	Si	Si	Código de la actividad	Actividad (CodAct)
264	CodPro	Varchar(20)	Si	Si	Código de proyecto	Proyecto(CodPro)
265	NombreEta	Varchar(20)	Si	Si	Código de etapa	Etapa(NombreEta)
266	Duracion	Integer	No	No	Duración de la	

					actividad	
267	comienzoA	Integer	No	No	Comienzo lo antes posible	
268	TerminoA	Integer	No	No	Termino lo más antes posible	
269	ComienzoT	Integer	No	No	Comienzo lo más tarde posible	
270	TerminoT	Integer	No	No	Termino lo mas temprano posible	
271	Holgura	Integer	No	No	Holgura de la actividad	
272	Critica	Boolean	No	No	Si es critica o no	

Tabla TELEFONO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
273	NumeroT	Varchar(15)	Si	No	Número de teléfono	
274	LugarT	Varchar(1)	No	No	Ubicacion del telefono	
275	CI	Varchar(10)	Si	Si	Cédula de Identidad del usuario	Ususistem(CI)

Tabla TEMA

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
276	CodTem	Varchar(5)	Si	No	Código del tema	
277	NombreT	Varchar (250)	No	No	Nombre del tema	
278	Autor	Varchar(10)	No	No	Código del usuario del foro	
279	NumPre	Integer	No	No	Nro de preguntas	
280	NumRes	Integer	No	No	Nro de respuestas	
281	FechaT	Date	No	No	Fecha en que se registró el tema	

Tabla TIPO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
282	CI	Varchar(10)	Si	Si	Cedula de	Ususitem(CI)

					Identidad del usuario	
283	NombreR	Varchar(20)	Si	Si	Nombre del rol	Rol(NombreR)

Tabla TRANSICIONA

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
284	Id	Integer nextval	Si	No	Identificador	
285	Precedente	Varchar(30)	No	Si	Nombre del estado de actividad	EstadoAct (Nombrees)
286	Concecuente	Varchar(30)	No	No	Nombre del estado de entrada	
287	TipoPro	Varchar(15)	No	Si	Tipo de proyecto investigación o extensión)	

Tabla TRANSICIONP

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
288	Id	Integer nextval	Si	No	Identificador	
289	Presedente	Varchar(30)	No	Si	Nombre del estado de proyecto	EstadoPro (Nombrees)
290	Consecuente	Varchar(30)	No	No	Nombre del estado de entrada	
291	TipoPro	Varchar(15)	No	Si	Tipo de proyecto (investigación/extensión)	

Tabla UACADEMICA

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
292	CodUni	Varchar(15)	Si	No	Codigo de unidad	
293	Nombre	Varchar(150)	No	No	Nombre de unidad	
294	AliasUa	Varchar(20)	No	No	Alias de unidad	

Tabla UAPRO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
295	CodPro	Varchar(20)	Si	Si	Codigo del proyecto	Proyecto(CodPro)
296	CodUni	Varchar(15)	Si	Si	Codigo de unidad	Uacademica (CodUni)

Tabla USUPRO

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
297	CI	Varchar(10)	Si	Si	Cédula de Identidad del usuario	Ususistem(CI)
298	CodPro	Varchar(20)	Si	Si	Código del proyecto	Proyecto(CodPro)
299	NombreR	Varchar(20)	Si	Si	Código del rol	Rol(NombreR)
300	Cat	Varchar(20)	No	No	Categoría equivalente en investigación del usuario	
301	Ded	Varchar(20)	No	No	Tiempo efectivo en horas/semanas dedicadas al proyecto	
302	Sit	Boolean	No	No	Situación con relación al proyecto a la fecha de presentación	
303	Fechasit	Date	No	No	Fecha	
304	LugarTrab	Varchar(20)	No	No	Lugar donde trabaja el usuario	

Tabla USUSISTEM

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
305	CI	Varchar(10)	Si	No	Cédula de Identidad del usuario	
306	Nombre	Varchar(25)	No	No	Nombre del usuario	
307	Paterno	Varchar(25)	No	No	Apellido paterno del usuario	
308	Materno	Varchar(25)	No	No	Apellido materno del usuario	
309	FechaN	Date	No	No	Fecha de nacimiento del usuario	
310	LugarN	Varchar(40)	No	No	Lugar de nacimiento del usuario	
311	Ecivil	Varchar(14)	No	No	Estado civil del usuario (soltero, casado, viudo, divorciado)	
312	Domicilio	Varchar(50)	No	No	Domicilio del usuario	
313	Mail	Varchar(30)	No	No	Dirección del correo electrónico del usuario	
314	Url	Varchar(30)	No	No	Dirección de página Web del usuario del sistema	
315	Genero	Varchar(1)	No	No	Masculino = M, Femenino = F	
316	nombreGA	Varchar(40)	No	No	Grado académico del usuario del sistema	

Tabla USUWEB

Nro	Nombre del campo	Tipo de dato	LP	LF	Descripción	Tabla y campo referente
317	CI	Varchar(10)	Si	No	Identificador del usuario	
318	NombreCom	Varchar(80)	No	No	Nombre completo	

319	Correo	Varchar(30)	No	No	Correo electrónico	

5.4 ESPECIFICACIÓN DE CASOS DE USO

En la especificación de casos de uso se relaciona el diseño de pantallas, tanto de captura como de salida de datos, la descripción del caso de uso a través de lenguaje estructurado, donde se describen los métodos que se utilizan para realizar el caso de uso y el diseño navegacional.




El formato de la especificación de casos de uso se describe a continuación:

Nombre: el nombre del caso de uso a especificar

Generalidades:

- **Pantalla:** código de la(s) pantalla(s) que intervienen en el caso de uso
- **Usuario:** tipo de usuario que tiene acceso al caso de uso especificado.
- **Pantalla origen:** pantalla anterior, que da inicio al caso de uso.
- **Requerimientos:** variables necesarias para la ejecución del proyecto.

a) Pantallas: diseño de las pantallas que intervienen en el caso de uso, para ello se toma en cuenta las siguientes referencias:

-  Dato recuperado de la base de datos, no modificable
-  Dato recuperado de la base de datos, modificable
-  Dato introducido por el usuario

xxx-999: donde xxx describe el tipo de datos(X(n): Varchar con n caracteres, 999: numérico, dd/mm/aa: fecha, Y:Boolean), y 999 corresponde al número de variable que hace referencia en el diccionario de datos.

- b)* **Descripción:** a través del lenguaje estructurado realizamos la descripción de los casos de uso, explicando en forma breve los métodos que intervienen en el mismo.
- c)* **Navegacional:** diagrama de navegación entre las pantallas que forman parte del caso de uso.

CONCLUSIONES

Para finalizar este proyecto, a continuación se exponen las conclusiones derivadas de la realización del mismo:

- El objetivo propuesto en el presente trabajo a sido alcanzado, en vista a que el sistema automatiza el manejo de la información en el Nodo de Internet en Cosett.
- La tecnología empleada en la instrumentación del sistema es la que responde a los requerimientos del sistema.
- La metodología empleada RUP, mejora considerablemente la calidad de desarrollo del sistema, ya que la misma utiliza el Lenguaje Unificado (UML) para preparar todos los esquemas de un sistema software.
- La manera de estructurar la información es adecuada por lo que los usuarios no tendrán problemas al interactuar con el sistema.
- Contar con información oportuna y certera, es vital para brindar un mejor servicio al socio.

RECOMENDACIONES

- Se recomienda al área de Internet la explotación del sistema aprovechando las ventajas del mismo.
- Se recomienda agregar mas opciones al sistema, para poder abarcar mas aspectos del área de Internet.

2.1 METODOLOGIA EMPLEADA

2.1.1 Proceso Unificado

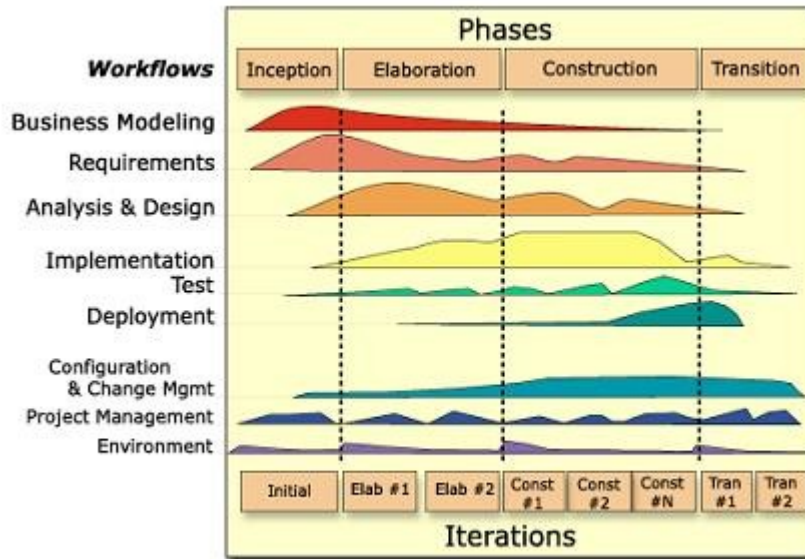
La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, es un proceso de desarrollo de software. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software.

El proceso unificado utiliza el *Lenguaje Unificado de Modelado* UML, para preparar todos los esquemas de un sistema software, este proceso se sostiene en tres ejes deseables en cualquier desarrollo orientado a objetos:

- **Proceso unificado dirigido por casos de uso**, los casos de uso se utilizan como artefacto principal para definir el comportamiento deseado para el sistema, y para comunicar este comportamiento entre las personas involucradas en el sistema. También indica que los casos de uso son la entrada principal para el análisis, diseño, implementación y prueba del sistema, incluyendo la creación, verificación y validación del sistema.
- **Proceso unificado centrado en la arquitectura**, la arquitectura es una vista de diseño completo con las características más importantes resaltadas, dejando los detalles de lado. Debido a que lo que es significativo depende en parte de una valoración, que a su vez, se adquiere con la experiencia, el valor de una arquitectura depende de las personas que se hayan responsabilizado de su creación. No obstante, el proceso ayuda al arquitecto a centrarse en los objetivos adecuados, como la comprensibilidad la capacidad de adaptación al cambio y la reutilización.
- **El proceso unificado es iterativo e incremental**, es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen

2.1.1.1.1 Fases dentro de un ciclo

Cada ciclo se desarrolla a lo largo del tiempo. Este tiempo a su vez, se divide, en 4 fases como se muestra en la figura 4



- **Inicio**, en esta fase se identifican los riesgos mas importantes, se planifica en detalle la fase de elaboración y se estima el proyecto de manera aproximada.
- **Elaboración**, durante esta fase de desarrollo se realizan los casos de uso más críticos, que se identificaron el fase de comienzo. El resultado en esta fase es una **línea base** de la arquitectura.
- **Construcción**, en esta fase la línea base de la arquitectura crecer hasta convertirse hasta el sistema completo. La descripción evoluciona hasta convertirse en producto preparado para se entregado a la comunidad de usuarios.
- **Transmisión**, cubre el periodo durante el cual el producto se convierte en versión beta.

2.1.1.1.2 Flujos de trabajo fundamentales

- **Requerimientos:** Traslado de las necesidades del negocio a un sistema automatizado.
- **Análisis y Diseño:** Traslado de los requerimientos dentro de la arquitectura de software.
- **Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

2.2 DETERMINACIÓN DE REQUERIMIENTOS, ANÁLISIS Y DISEÑO DE SISTEMAS MEDIANTE EL LENGUAJE UML

El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language), es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

2.2.1 Objetivos de UML

- UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.

- UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.
- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial.

2.2.2 Diagramas UML

Se dispone de dos tipos diferentes de diagramas los que dan una vista estática del sistema y los que dan una visión dinámica.

2.2.2.1 Diagramas estáticos

Diagrama de clases: muestra las clases, interfaces, colaboraciones y sus relaciones. Son los más comunes y dan una vista estática del proyecto.

Diagrama de objetos: es un diagrama de instancias de las clases mostradas en el diagrama de clases. Muestra las instancias y cómo se relacionan entre ellas. Se da una visión de casos reales.

Diagrama de componentes: muestran la organización de los componentes del sistema. Un componente se corresponde con una o varias clases, interfaces o colaboraciones.

Diagrama de despliegue: muestra los nodos y sus relaciones. Un nodo es un conjunto de componentes. Se utiliza para reducir la complejidad de los diagramas de clases y componentes de un gran sistema. Sirve como resumen e índice.

Diagrama de casos de uso: muestran los casos de uso, actores y sus relaciones. Muestra quién puede hacer y qué relaciones existen entre acciones (casos de uso). Son muy importantes para modelar y organizar el comportamiento del sistema.

Diagrama de clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento.

Un diagrama de clases está compuesto por los siguientes elementos:

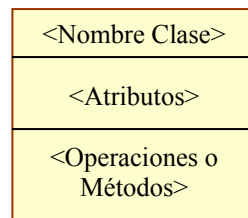
Clase

Relaciones

Clase:

Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio.

En UML, una clase es representada por un rectángulo que posee tres divisiones:



En donde:


Superior: contiene el nombre de la Clase


Intermedio: contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).


Inferior: contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).

Atributos:

Los atributos o características de una Clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno, estos son:

public (+, 


private (-, 


protected (#, 

Métodos:

Los métodos u operaciones de una clase son la forma en como ésta interactúa con su entorno, estos pueden tener las características:

public (+,

private (-, ): indica que el método sólo será accesible desde dentro de la clase (sólo otros métodos de la clase lo pueden acceder).

protected (#, ): indica que el método no será accesible desde fuera de la clase, pero sí podrá ser accesado por métodos de la clase además de métodos de las subclases que se deriven.

Relaciones entre Clases:

Ahora ya definido el concepto de Clase, es necesario explicar cómo se pueden interrelacionar dos o más clases (cada uno con características y objetivos diferentes).

Antes es necesario explicar el concepto de cardinalidad de relaciones: en UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y éstas pueden ser:

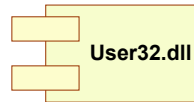
- **uno o muchos:** 1..* (1..n)
- **0 o muchos:** 0..* (0..n)
- **número fijo:** m (m denota el número).⁴

Diagrama de componentes

Se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema. En él situaremos librerías, tablas archivos, ejecutables y documentos que formen parte del sistema.

⁴ <http://www.dcc.uchile.cl/~psalinas/uml/modelo.html>

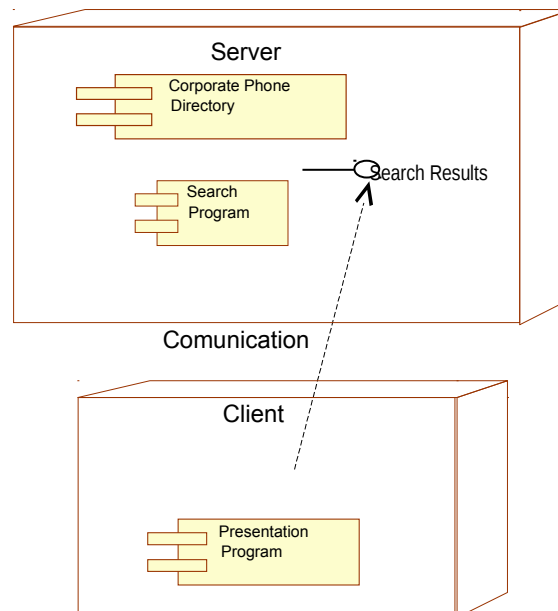
Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.



Aquí tenemos un componente del sistema de Windows. En el diagrama de componentes de Windows debe salir este componente, ya que sin él, el sistema no funcionaría.

Diagrama de despliegue

En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados. Es decir, se sitúa el software en el hardware que lo contiene. Cada Hardware se representa como un nodo.



Un nodo se representa como un cubo, un nodo es un elemento donde se ejecutan los componentes, representan el despliegue físico de estos componentes.

Aquí tenemos dos nodos, el cliente y el servidor, cada uno de ellos contiene componentes. El componente del cliente utiliza un interface de uno de los componentes del servidor. Se muestra la relación existente entre los dos Nodos. Esta relación podríamos asociar a un estereotipo para indicar qué tipo de conexión disponemos entre el cliente y el servidor, así como modificar su cardinalidad, para indicar que soportamos diversos clientes.

Como los componentes pueden residir en más de un nodo podemos situar el componente de forma independiente, sin que pertenezca a ningún nodo, y relacionarlo con los nodos en los que se sitúa.

2.7.2.1.2 Diagrama de casos de uso


El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos:

Actor: un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

Caso de uso: es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

Relaciones:

Asociación 

Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

Extensión (<<Extend>>) 

Es la inserción de comportamiento adicional en un caso de uso base que no tiene conocimiento sobre él. Dicha relación se denota con una flecha punteada.



Inclusión (<<Include>>)

Inserción de comportamiento adicional en un caso de uso base que describe explícitamente la inserción. Dicha relación se denota con una flecha punteada.

Generalización 

Es una relación entre un caso de uso general y un caso de uso más específico, que hereda y añade propiedades de aquel. Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).

Diagramas dinámicos

Diagrama de secuencia, Diagrama de colaboración: muestran a los diferentes objetos y las relaciones que pueden tener entre ellos, los mensajes que se envían entre ellos. Son dos diagramas diferentes, que se puede pasar de uno a otro sin pérdida de información, pero que nos dan puntos de vista diferentes del sistema. En resumen, cualquiera de los dos es un Diagrama de Interacción.

Diagrama de estados: muestra los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos.

Diagrama de actividades: es un caso especial del diagrama de estados. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos.

Diagrama de secuencia

El diagrama de interacción, representa la forma en cómo un Cliente (Actor) u Objetos (Clases) se comunican entre sí en petición a un evento. Esto implica

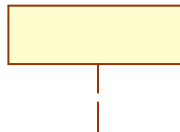
recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades claramente.

Dicho diagrama puede ser obtenido de dos partes, desde el Diagrama Estático de Clases o el de Casos de Uso (son diferentes).

Los componentes de un diagrama de secuencia son:

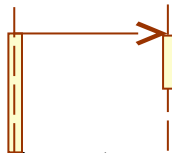
- Un Objeto o Actor
- Mensaje de un objeto a otro objeto.
- Mensaje de un objeto a sí mismo
- Elementos

Objeto/actor:



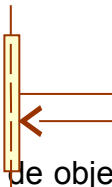
El rectángulo representa una instancia de un Objeto en particular, y la línea punteada representa las llamadas a métodos del objeto.

Mensaje a Otro Objeto:



Se representa por una flecha entre un objeto y otro, representa la llamada de un método (operación) de un objeto en particular.

Mensaje al Mismo Objeto:



No sólo llamadas a métodos de objetos externos pueden realizarse, también es posible visualizar llamadas a métodos desde el mismo objeto en estudio. ⁵

⁵ <http://www.dcc.uchile.cl/~psalinas/uml/interaccion.html>

Diagrama de colaboración

El diagrama de colaboración presenta una alternativa al diagrama de secuencia para modelar interacciones entre objetos en el sistema. Mientras que el diagrama de secuencia se centra en la secuencia cronológica del escenario que estamos modelando, el diagrama de colaboración se centra en estudiar todos los efectos de un objeto dado durante un escenario.

Un diagrama de colaboración está compuesto por los siguientes elementos:

- *Objetos*. Los objetos se representan con un rectángulo, que contiene el nombre y la clase del objeto en un formato nombreObjeto: nombreClase, se conectan por medio de **enlaces**, cada enlace representa una instancia de una asociación entre las clases implicadas. El enlace muestra los mensajes enviados entre los objetos, el tipo de mensaje (sincrónico, asincrónico), y la visibilidad de un objeto con respecto a los otros.
- *Mensajes*. Los mensajes se muestran como flechas etiquetadas unidas a los enlaces. Cada mensaje tiene un número de secuencia, una lista opcional de mensajes precedentes, una condición opcional de guarda, un nombre y lista de argumentos, y un nombre de valor de retorno opcional. El número de serie incluye el nombre(opcional) de un hilo. Todos los mensajes del hilo se ordenan secuencialmente. Los mensajes de diversos hilos son concurrentes a menos que haya una dependencia secuencial explícita. Pueden añadirse varios detalles de implementación como por ejemplo una distinción entre los mensajes síncronos y asíncronos.
- *Flujos de mensajes*. Expresa el envío de un mensaje. Se representa mediante una flecha dirigida cercana a un enlace.

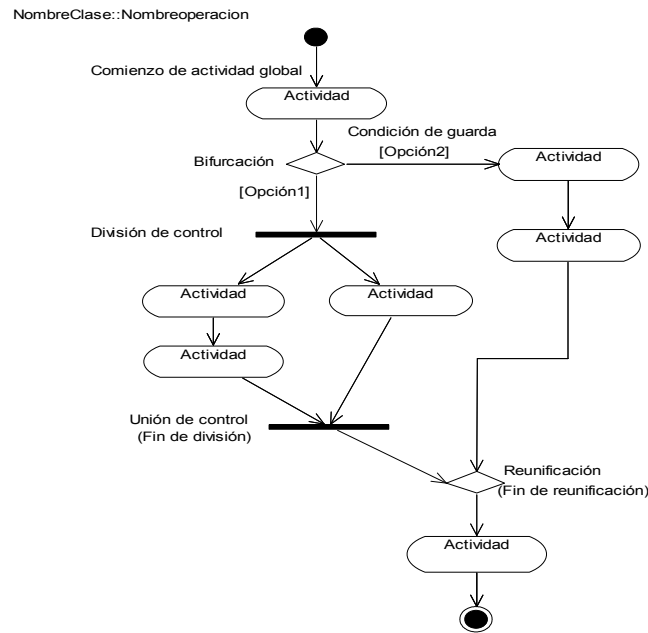
Diagrama de estado

- El Diagrama de Estados modela el comportamiento de una parte del sistema a través del tiempo
- Típicamente se elabora un diagrama de Estados para cada clase que tenga un comportamiento significativo
- El comportamiento es modelado en términos del estado en el cual se encuentra el objeto, qué acciones se ejecutan en cada estado y cuál es el estado al que transita después de un determinado evento

Diagrama de actividades

El Diagrama de Actividad es una especialización del Diagrama de Estado, organizado respecto de las acciones y usado para especificar:

- Un método
- Un caso de uso
- Un proceso de negocio: Workflow



Un estado de actividad representa una actividad: un paso en el flujo de trabajo o la ejecución de una operación. Un grafo de actividades describe grupos secuenciales y concurrentes de actividades. Los grafos de actividades se muestran en diagramas de actividades. Las actividades se enlazan por transiciones automáticas. Cuando una actividad termina se desencadena el paso a la siguiente actividad.

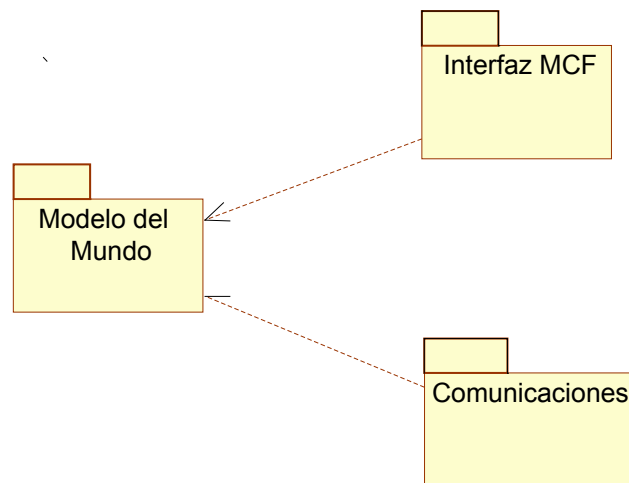
Un diagrama de actividades es provechoso para entender el comportamiento de alto nivel de la ejecución de un sistema, sin profundizar en los detalles internos de los mensajes. Los parámetros de entrada y salida de una acción se pueden mostrar usando las relaciones de flujo que conectan la acción y un estado de flujo de objeto.

Un diagrama de actividades puede dar detalle a un caso de uso, un objeto o un mensaje en un objeto. Permiten representar transiciones internas al margen de las transiciones o eventos externos. En la figura se presenta un ejemplo de diagrama de actividades para un mensaje de un objeto.

Diagrama de paquetes

– Paquete

Un paquete es una forma de agrupar clases (u otros elementos en otro tipo de diagramas) en modelos grandes. Pueden tener asociaciones de dependencia o de generalización entre ellos. Un ejemplo puede ser el siguiente:



En este caso existen tres paquetes (que se muestran vacíos en este caso, con su contenido encapsulado), con dos de ellos dependiendo del Modelo del Mundo

– Dependencia

Denota una relación semántica entre dos elementos (clases o paquetes, por el momento) del modelo. Indica que cambiar el elemento independiente puede requerir cambios en los dependientes. Se muestra como una línea punteada direccional, indicando el sentido de la dependencia. Puede tener por medio de estereotipos una explicación del tipo de dependencia presentada.

En el ejemplo anterior pueden verse dos relaciones de dependencia hacia el paquete Modelo del Mundo.

Como podemos ver, el número de diagramas es muy alto, en la mayoría de los casos excesivos, y UML permite definir sólo los necesarios, ya que no todos son necesarios en todos los proyectos³. En este caso en particular emplearemos los siguientes diagramas: Diagrama de casos de uso, Diagrama de clases, Diagrama de Secuencia, Diagramas de Colaboración, Diagrama de Actividades, Diagrama de Despliegue y Diagrama de Componentes.

2.3 TECNOLOGÍAS WEB

Bajo el concepto de tecnologías Web, se agrupan una serie de nuevas tecnologías y estándares que hacen posible Internet e Intranet:

HTML (HyperText Markup Language) es el lenguaje básico de hacer archivos de texto con hipervínculos, publicarlos en un Servidor Web, y "transportarlos" a diferentes usuarios, que pueden leerlos gracias a los browsers.

Java es un lenguaje de programación diseñado para ser utilizado en la red (es decir, no está instalado localmente), y se utiliza para dar a las páginas Web características extra no disponibles en HTML.

TCP/IP (Transport Control Protocol/Internet Protocol) es el protocolo de comunicación básico de Internet e Intranets, y permite la comunicación punto-a-punto entre los distintos ordenadores de una red.

HTTP (Hypertext Transfer Protocol) es el protocolo Web que controla las peticiones y servicios de documentos HTML

FTP (File Transfer Protocol) es un protocolo utilizado para transmitir archivos a través de la red, desde un disco duro a otro.

SMTP (Simple Mail Transfer Protocol) es un protocolo para el envío y recepción de mensajes de correo electrónico (e-mail)

³ <http://usuarios.lycos.es/oopere/uml.htm>

2.4 APLICACIONES WEB

Una aplicación Web consta de una o más páginas conectadas entre sí. Una página Web es un archivo de texto que contiene lenguaje de marcas de hipertexto (HTML), etiquetas de formato y vínculos a archivos gráficos y a otras páginas Web.

El archivo de texto se almacena en un servidor de Web al que pueden acceder otras computadoras conectadas a ese servidor, vía Internet o una LAN. Al archivo se puede acceder utilizando exploradores Web que no hacen otra cosa que efectuar una transferencia de archivos e interpretación de las etiquetas y vínculos HTML, y muestran el resultado en el monitor.

Otra definición sería que una página Web es un formulario interactivo que utiliza una red de computadoras.

Hay dos propiedades de las páginas Web que las hacen únicas: que son interactivas y que pueden usar objetos multimedia. El término multimedia se utiliza para describir archivos de texto, sonido, animación y vídeo que se combinan para presentar la información, por ejemplo, en una enciclopedia interactiva o juego.

Cuando esos mismos tipos de archivo se distribuyen por Internet o una LAN, se puede utilizar el término hypermedia para describirlos. Gracias al World Wide Web ya es posible disponer de multimedia a través de Internet.

Cada página Web tiene asociada una dirección o URL, un URL es la ruta a una página determinada dentro de Internet, se utiliza de la misma forma que para localizar un archivo en una computadora, en este caso indica que es la página principal que está situada en el servidor de Microsoft que está conectado a la World Wide Web .

La World Wide Web consiste en ofrecer una interfase simple y consistente para acceder a la inmensidad de los recursos de Internet. Es la forma más

moderna de ofrecer información, el medio más potente. La información se ofrece en forma de páginas electrónicas.

El World Wide Web o WWW o W3 o simplemente Web, permite saltar de un lugar a otro en pos de lo que nos interesa. Lo más interesante es que con unas pocas órdenes se puede mover por toda la Internet.⁶

Una de las características más potentes de las aplicaciones Web actuales es su posibilidad de enlazarse con bases de datos, ofreciendo una gama de potentes servicios tales como: comercio electrónico (e-commerce), bibliotecas virtuales, descarga de software, etc. todos ellos basados en consultas y gestión de bases de datos.

2.5 DISEÑO DE PÁGINAS WEB DINÁMICAS

2.5.1 Sitio Web Dinámico

La inmensa mayoría de la información contenida en la WWW, está periclitada o no suficientemente actualizada, y por lo tanto bastante inútil en muchos casos. Una de las principales razones de esa falta de actualización de datos, es la falta de técnicos conocedores de los procesos de mantenimiento. Sabiendo esto, parece de todo punto obvio que para mantener actualizada y viva la información de un sitio Web, hay que diseñar y ofrecer aplicaciones y procesos que permitan que cualquier persona, con un simple conocimiento mecanográfico, pueda mantener y refrescar su propia Web.

Cuando, además, el tema de la Web versa sobre algo tan vivo como la Investigación, unida a la idea de Trabajo en Equipo, no se puede por menos que desarrollar unos Entornos de Trabajo que unan todas las potencialidades que hoy ofrece la tecnología Web, pero además estos recursos tienen que venir a simplificar, más que a complicar y dificultar, los procesos de trabajo.

⁶ <http://www.monografias.com/>

Un sitio Web dinámico sería aquel que, por su concepción como área de trabajo, y como continente de información cambiante, permita su uso como herramienta eficaz para equipos de trabajo online.

2.5.2 Sitio Web de Acceso Privado

Se entiende como aquel sitio Web que sólo permite accesos a la información que contiene sobre la base de permisos concedidos por los responsables del mismo.

Un sitio Web de acceso privado o limitado puede además tener distintos niveles de acceso, en función de los permisos concertados o contratados con sus propietarios.

2.6 USABILIDAD EN LA WEB

Podemos definir la usabilidad de un sistema (producto o servicio) como la medida en la cual puede ser usado para conseguir objetivos concretos con efectividad, eficiencia y satisfacción. Más concretamente, la norma ISO 9241:11 de 1993 define la usabilidad como **"la facilidad de uso de una aplicación informática"**.

Por lo que respecta a la Web, la usabilidad contempla un conjunto de técnicas que ayudan a los seres humanos a realizar tareas en el entorno gráfico de la interfase de usuario de una aplicación Web. Y decimos "aplicación Web" porque la usabilidad no va a afectar sólo a páginas aisladas, sino al conjunto de todas las páginas que componen un sitio Web, abarcando aspectos como la navegación entre ellas (navegabilidad), la facilidad de encontrar un elemento dado dentro de sitio (relacionada con la arquitectura de la información), el número de errores del sistema (relacionado con aspectos de programación), la legibilidad de los textos (correspondiente al diseño Web), la correcta transmisión de la información, el acceso a ésta por todos los posibles usuarios (accesibilidad), etc.

La usabilidad es entonces la última responsable de que un sitio Web cumpla de forma correcta las expectativas con las que ha sido creado, siempre desde el punto de vista del usuario final, de los visitantes de las páginas, ya que son estos los que, al fin y al cabo, van a hacer que un sitio Web sea o no exitoso.

Vista la definición de usabilidad, está claro que la misma es un elemento primordial e imprescindible en la concepción y desarrollo de un sitio Web.

Lamentablemente, muchas han sido y siguen siendo las empresas que desarrollan sitios Web complejos sin tener en cuenta la usabilidad del mismo. Unas veces, por ignorancia, otras por falta de recursos, otras por diseñar los sitios pensando más en la empresa y en sus directivos que en el usuario final. Sea cual sea el motivo, lo que está claro es que, al final, cuando la aplicación Web está en producción, si no se ha contemplado el tema de la usabilidad en todos y cada uno de los pasos de creación de la misma, el sitio Web está condenado al fracaso.

Principios generales de usabilidad en sitios Web

Aunque estos principios pueden parecer muy generales son la base de la disciplina y de su extrapolación y perfecta comprensión permite resolver cualquier cáustica concreta.

El diseño de sitios Web deben seguir los siguientes principios:

1. Anticipación, el sitio Web debe anticiparse a las necesidades del usuario.
2. Autonomía, los usuarios deben tener el control sobre el sitio Web. Los usuarios sienten que controlan un sitio Web si conocen su situación en un entorno abarcable y no infinito.
3. Los colores han de utilizarse con precaución para no dificultar el acceso a los usuarios con problemas de distinción de colores (aprox. un 15% del total).

4. Consistencia, las aplicaciones deben ser consistentes con las expectativas de los usuarios, es decir, con su aprendizaje previo.
5. Eficiencia del usuario, los sitios Web se deben centrar en la productividad del usuario, no en la del propio sitio Web. Por ejemplo, en ocasiones tareas con mayor número de pasos son más rápidas de realizar para una persona que otras tareas con menos pasos, pero más complejas.
6. Reversibilidad, un sitio Web ha de permitir deshacer las acciones realizadas
7. Aprendizaje, los sitios Web deben requerir un mínimo proceso de aprendizaje y deben poder ser utilizados desde el primer momento.
8. Legibilidad, el color de los textos debe contrastar con el del fondo, y el tamaño de fuente debe ser suficientemente grande.
9. Seguimiento de las acciones del usuario. Conociendo y almacenando información sobre su comportamiento previo se ha de permitir al usuario realizar operaciones frecuentes de manera más rápida.
10. Interfaz visible. Se deben evitar elementos invisibles de navegación que han de ser inferidos por los usuarios, menús desplegados, indicaciones ocultas, etc.

Otros principios para el diseño de sitios Web son:

- a) Los usuarios deben ser capaces de alcanzar sus objetivos con un mínimo esfuerzo y unos resultados máximos.
- b) Un sitio Web no ha de tratar al usuario de manera hostil. Cuando el usuario comete un error el sistema ha de solucionar el problema, o en su defecto sugerir varias soluciones posibles, pero no emitir respuestas que meramente informen del error culpando al usuario.

- c) En ningún caso un sitio Web puede venirse abajo o producir un resultado inesperado. Por ejemplo, no deben existir enlaces rotos.
- d) Un sitio Web debe ajustarse a los usuarios. La libertad en el uso de un sitio Web es un término peligroso, cuanto mayor sea el número de acciones que un usuario pueda realizar, mayor es la probabilidad que cometa un error. Limitando el número de acciones al público objetivo se facilita el uso de un sitio Web.
- e) Los usuarios no deben sufrir sobrecarga de información. Cuando un usuario visita un sitio Web y no sabe donde comenzar a leer, existe sobrecarga de información.
- f) Un sitio Web debe ser consistente en todos los pasos del proceso. Aunque pueda parecer apropiado que diferentes áreas tengan diseños diferentes, la consistencia entre los diseños facilita al usuario el uso de un sitio.

Entre los beneficios más evidentes que conlleva la aplicación de la ingeniería de la usabilidad al diseño y desarrollo de un sitio Web podemos citar:

- **Reducción del tiempo final de desarrollo**, con lo que disminuyen los costes y se puede sacar antes una aplicación verdaderamente funcional al mercado. Esto es así porque se evitan posteriores procesos de rediseño del sitio, originados por una mala concepción inicial del mismo.
- **Reducción de los costes de mantenimiento del sitio Web** al ofrecer al público un producto homogéneo y funcional desde el principio.
- Obtención de sistemas más fáciles de entender, usar y recordar, con lo que **se aumenta la productividad de los usuarios y la eficiencia de los procesos**.

- **Mayor calidad del producto final**, con lo que la aceptación del mismo por los usuarios finales es mejor y más rápida, consiguiendo con ello más visitas, y a la vez, una ventaja competitiva de la empresa frente a los competidores.
- **Menor número de reclamaciones por parte de los usuarios**, con lo que disminuyen los costos por atención al cliente.
- **Aumento de la satisfacción de los usuarios** y disminución del estrés ocasionado por el uso de la aplicación Web, consiguiendo con ello fidelizar a los mismos, por lo que se convertirán en visitantes frecuentes de nuestro sitio.
- **Reducción del tiempo de ejecución de tareas** por parte de los usuarios y del número de errores en las mismas, con lo que se consigue un mayor número de procesos finalizados exitosamente y un menor número de abandonos en los mismos.

No debemos olvidar nunca que las personas que acceden a un sitio Web no tienen ni deben porqué ser expertos conocedores de los múltiples sistemas de navegación posibles, de los conceptos que para los desarrolladores son comunes ni de las diferentes tecnologías en uso. Además, el usuario es nervioso e impaciente, y navega persiguiendo objetivos bien definidos, siéndole indiferente cómo se ha desarrollado una aplicación Web, qué cosas puede hacer y los procesos que corran bajo cuerda; a él sólo le importa completar la tarea por la que ha accedido a la aplicación.

Efectivamente, **la usabilidad no es sólo un mero concepto** del que hay que hablar porque está de moda, ni una herramienta para conseguir mejores resultados. **Es una forma de pensar y desarrollar** una concepción lógica y efectiva de diseñar y programar pensando siempre en el usuario final, protagonista absoluto de nuestro producto informático.

Ya dentro del modelo de desarrollo, los principios y técnicas de la ingeniería de la usabilidad se verán involucrados en los siguientes aspectos:

- **Arquitectura de la información:** la información que se facilitará a los usuarios y el sistema de organización de la misma se determinarán de acuerdo a las necesidades de los mismos, no según las preferencias, gustos o comodidades corporativas, de diseño o de programación. El diseño de la información será luego determinante para los sistemas de navegación del sitio Web.
- **Diseño de la interfase de usuario:** elemento fundamental de la aplicación Web, la interfase es la herramienta que facilitamos al usuario para interactuar con el motor de la aplicación, para realizar sus tareas y buscar la información que necesita. Debe ser fácil de usar y de aprender, cómoda y efectiva, usando metáforas de diseño adecuadas, con elementos conocidos por todos. Los procesos de usuario deben ser concebidos e implementados de forma que resulten fáciles de entender y completar por el usuario medio.
- **Programación interna:** los algoritmos de programación y los procesos internos de la aplicación (lenguajes usados, accesos a bases de datos, servidores utilizados, etc.) deben estar orientados en todo momento a facilitar al usuario la realización cómoda de tareas, sacrificando, si es preciso hacerlo, aspectos lógicos del desarrollo, con el fin de construir procesos de usuario, no procesos de programación.
- **Accesibilidad:** la aplicación Web debe posibilitar su uso por todo tipo de usuarios (con problemas de visión, con dificultades auditivas, etc.), en cualquier situación de visualización posible, con cualquier máquina, sistema operativo y navegador Web, sin requerimientos de plugins o software adicional.

2.7 GESTIÓN DEL CONOCIMIENTO

El conocimiento constituye el activo más valioso de cualquier organización en la Sociedad de la Información. La competitividad de las empresas, y por lo tanto su supervivencia, depende de que este conocimiento pueda preservarse y utilizarse en forma eficiente. La Gestión del Conocimiento pretende poner al alcance de cada empleado la información que necesita en el momento preciso para que su actividad sea efectiva.

2.8 HERRAMIENTAS PARA LA IMPLEMENTACIÓN

2.8.1 PHP (Personal Home Page)

Es un lenguaje de programación pensado en la Web de forma que es ideal para la creación de páginas dinámicas.

PHP es un lenguaje encapsulado dentro de los documentos html. De forma que se pueden introducir instrucciones php dentro de las páginas. Gracias a esto el diseñador gráfico del Web puede trabajar en forma independiente al programador. PHP es interpretado por el servidor (apache) generando un HTML con el resultado de sustituir las secuencias de instrucciones PHP por su salida.

Por lo tanto, una Web dinámica con PHP contiene una serie de documentos php que el servidor apache interpreta proporcionando al cliente documentos html con el resultado de las órdenes php.

Ventajas de PHP

- Muy sencillo de aprender.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática. Librándose el usuario de tener que separar las variables y sus valores.
- Se puede incrustar código PHP con etiquetas HTML.

- Excelente soporte de acceso a base de datos.
- La comprobación de que los parámetros son validos se hace en el servidor y no en el cliente (como se hace con javascript) de forma que se puede evitar que chequear que no se reciban solicitudes adulteradas.
- PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.
- Se puede hacer de todo lo que se pueda transmitir por vía HTTP.

Características

- PHP no soporta directamente punteros, como el C, de forma que no existen los problemas de depuración provocados por estos.
- Se pueden hacer grandes cosas con pocas líneas de código.
- Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, gestión de un e-commerce, xml, creación de PDF ...)
- Al poderse encapsular dentro de código html se puede recoger el trabajo del diseñador gráfico e incrustar el código php posteriormente.
- Está siendo utilizado con éxito en varios millones de sitios Web.
- Es multiplataforma, funciona en todas las plataformas que soporten apache.

2.8.2Mysql

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

sa GNU Automake, Autoconf, y Libtool para portabilidad

Uso de multihilos mediante hilos del kernel.

Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice

Tablas hash en memoria temporales

El código MySQL se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL.

Completo soporte para operadores y funciones en cláusulas select y where.

Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación

Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.

Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.

Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).

Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando named pipes y en sistemas Unix usando ficheros socket Unix.

En MySQL 5.0, los clientes y servidores Windows se pueden conectar usando memoria compartida.

MySQL contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.

Disponibilidad en gran cantidad de plataformas y sistemas.

Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones...

Transacciones y claves foráneas.

Conectividad segura.

Replicación.

Búsqueda e indexación de campos de texto.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que

contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.

Características distintivas

Las siguientes características son implementadas únicamente por MySQL:

Permite escoger entre múltiples motores de almacenamiento para cada tabla. En MySQL 5.0 éstos debían añadirse en tiempo de compilación, a partir de MySQL 5.1 se pueden añadir dinámicamente en tiempo de ejecución:

Los hay nativos como MyISAM, Falcon, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example

Desarrollados por partners como solidDB, NitroEDB, ScaleDB, TokuDB, Infobright (antes Brighthouse), Kickfire, XtraDB, IBM DB2). InnoDB Estuvo desarrollado así pero ahora pertenece también a Oracle

Desarrollados por la comunidad como memcache, httpd, PBXT y Revision

Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

Tipos de compilación del servidor

Hay tres tipos de compilación del servidor MySQL:

Estándar: Los binarios estándar de MySQL son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento InnoDB.

Max (No se trata de MaxDB, que es una cooperación con SAP): Los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.

MySQL-Debug: Son binarios que han sido compilados con información de depuración extra. No debe ser usada en sistemas en producción porque el código de depuración puede reducir el rendimiento.

Especificaciones del código fuente

MySQL está escrito en una mezcla de C y C++.

2.8.3 Laravel

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti".

Laravel tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP.²

Gran parte de Laravel está formado por dependencias, especialmente de Symfony, esto implica que el desarrollo de Laravel dependa también del desarrollo de sus dependencias.

Características

Sistema de ruteo, también RESTful³

Blade, Motor de plantillas⁴ ⁵

Peticiones Fluent⁶

Eloquent ORM⁷

Basado en Composer⁸

Soporte para el caché⁹

Soporte para MVC¹⁰

Usa componentes de Symfony¹¹

Patrón MVC

Laravel, propone en el desarrollo usar 'Routes with Closures', en lugar de un MVC tradicional con el objetivo de hacer el código más claro. Aun así permite el uso de MVC tradicional.¹⁰

```
<?php
```

```
Route::get('usuario/listar', function() {  
    //punto de entrada de la petición HTTP  
    ...  
});
```

Modelo

Laravel incluye un sistema de mapeo de datos relacional llamado Eloquent ORM que facilita la creación de modelos. Este ORM se funda en patrón active record y su funcionamiento es muy sencillo. Es opcional el uso de Eloquent, pues también dispone de otros recursos que nos facilitan interactuar con los datos, o específicamente la creación de modelos.⁷

La forma de crear Modelos en Laravel usando Eloquent ORM, es tan simple como:

```
class Libro extends Eloquent {

    //defiendo el nombre de la tabla con la info de los libros

    public static $table = 'tb_libros';

}
```

Incluso el código del modelo puede ser más simple aún, en el caso que el nombre de la tabla coincida con el nombre de la clase. Pues Laravel al igual que Ruby on Rails, usa el paradigma de programación donde se favorece "la convención sobre la configuración". Y si ahora necesitamos disponer un listado, en la ruta: <http://mi-aplicacion.com/libro/listar> Entonces, sólo bastaría crear la Ruta e interactuar con el modelo 'Libro' anteriormente creado, del modo siguiente:

```
Route::get('libro/listar', function() {

    $libros = Libro::all();

    return View::make('mi_vista', $libros); //Muestra los datos

});
```

Vista

Laravel incluye de paquete un sistema de procesamiento de plantillas llamado Blade. Este sistema de plantillas favorece un código mucho más limpio en las Vistas, además de incluir un sistema de Caché que lo hace mucho más rápido. El sistema Blade de Laravel, permite una sintaxis mucho más reducida en su escritura.⁴ Por ejemplo, en vez pintar la vista usando el código PHP:

```
<php echo $mi_nombre; ?>
```

En Blade se escribiría:

```
{{ $mi_nombre }}
```

Lo cual no es una gran ventaja sobre todo cuando siempre es posible usar una expresión resumida en PHP. No obstante, lo que sí es una gran ventaja, es el modo en que Blade maneja las plantillas.

Plantillas

Las plantillas en Blade son archivos de texto plano que contiene todo el HTML de la página con etiquetas que representan elementos o zonas a incluir en la plantilla, o vistas parciales como se conocen en otros frameworks PHP. Sin embargo, en Blade estos elementos incrustados se organizan en un sólo archivo. Esta es una idea muy interesante de Laravel que mejora la organización de las vistas y su rendimiento. Sobre todo cuando las vistas pueden llegar a ser muy complejas incluso con elementos anidados. En el render de una Vista completa en Lavarel se usan dos archivos: la plantilla definiendo el HTML global y las zonas a incluir. Un sólo archivo, la Vista, con los elementos (partial views).

Un ejemplo típico para una plantilla HTML5 de Blade:

```
<!DOCTYPE html>  
  
<html lang="es">  
  
  <head>  
  
    <meta charset="UTF-8">
```

```

    <title>@yield('titulo')</title>
</head>
<body>
    @yield('navegacion')
</body>
</html>

```

En la plantilla presentada, el código `@yield()` identifica al método donde como parámetro se indica el nombre de la zona desplegar. Por otro lado, el código de la vista, donde se define la plantilla a usar y la información de las distintas zonas a desplegar:

```
<!-- identificando la plantilla a utilizar -->
```

```
@extends('template')
```

```
<!-- definiendo una zona -->
```

```
@section('titulo')
```

```
@stop
```

```
<!-- definiendo otra zona -->
```

```
@section('navegacion')
```

```
@stop
```

```
Controlador
```

Los controladores contienen la lógica de la aplicación y permiten organizar el código en clases sin tener que escribirlo todo en las rutas. Todos los controladores deben extenderse de la clase `BaseController`.

Un ejemplo de un controlador en Laravel:

```
class UserController extends BaseController {  
    public function mostrarPerfil($id)  
    {  
        $user = User::find($id);  
        return View::make('user.profile', array('user' => $user));  
    }  
}
```

Estos pueden ser llamados en las rutas de diferentes maneras, pero la más común usando rutas es:

```
Route::get('user/{id}', 'UserController@mostrarPerfil');
```

2.8.4 Xampp

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MySQL, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del

acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.

El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux.

2.9 SEGURIDAD DE LA BASE DE DATOS

Hoy en día, las bases de datos son componentes cardinales de cualquier aplicación basada en Web, permitiendo que los sitios Web provean contenido dinámico. Debido a que información considerablemente sensible o secreta puede ser almacenada en una base de datos, se debe considerar seriamente la protección de la base de datos.

Para recuperar o almacenar cualquier información es necesario conectarse a la base de datos, enviar una consulta válida, recoger el resultado y cerrar la conexión. Hoy en día, el lenguaje de consultas usado comúnmente en estas interacciones es el Lenguaje de Consultas Estructurado (SQL por sus siglas en inglés).

Es importante realizar copias de seguridad periódicamente o cuando el administrador del sistema lo requiera, esto nos permite tener disponible e íntegra la información para cuando sucedan los accidentes. Sin una copia de seguridad es imposible volver la información al estado anterior del desastre.