

I. Capítulo I

1.1 Presentación del Proyecto

1.1.1 Título del proyecto

“Gestión administrativa de la Brigada Parlamentaria de Tarija, mejorada”

1.1.2 Carrera / Unidad

Ingeniería Informática

1.1.3 Facultad

Ciencias y Tecnología

1.1.4 Duración del proyecto

10 meses

1.1.5 Área/línea de investigación priorizado

Tecnologías de la Información y Comunicación - Desarrollo de Sistemas Informáticos – Sistema Web

1.1.6 Responsable del proyecto

Carrera de Ingeniería Informática – Taller III

1.1.7 Entidad asociada (s)

Brigada Parlamentaria de “Tarija”.

1.2 Personal vinculado al proyecto

1.2.1 Director del proyecto

Huanca	Mamani	Antonio	7177505 Tja.
--------	--------	---------	--------------

Apellido Paterno	Apellido Materno	Lujan Nombre	CI
Estudiante Univ. Profesión	Ing. Informática Carrera o Unidad	Ciencias y Tecnología Facultad	
70230805 Celular	tonylujan.07hm@gmail.com Correo electrónico		Firma

Tabla 1 Director del Proyecto

1.2.2 Participantes del equipo de trabajo

Categoría	Nombres y Apellidos	Carrera/Profesión	C.I.	Firma
Director	Antonio Lujan Huanca Mamani	Estudiante en Ing. Informática	7177505 Tja.	
Tutor	Ing. Eduardo Luis Aguirre Gudiño	Ing. Informático	5032287	

Tabla 2 Participantes del equipo de trabajo

1.2.3 Equipo de trabajo de:

**Empresas/Instituciones/Organizaciones
participantes/cooperantes**

Nombre: Brigada Parlamentaria de “Tarija”.			
Dirección: C. Alejandro del Carpio Esq. General Trigo N°383		Teléf.	Oficina: (04)6912334-6917030

Nombre y Apellidos	Cargo	C.I.	Firma
Jacqueline Mendieta	Presidenta de la Brigada Parlamentaria de Tarija		

Tabla 3 Equipo de trabajo de: Empresas/Instituciones/Organizaciones participantes/cooperantes

1.2.4 Actividades previstas para los integrantes del equipo de investigación

Responsable	Actividades
Director	<p>Como Jefe de Proyecto:</p> <ul style="list-style-type: none"> - Planificar y controlar del cronograma del proyecto. - Asignar y gestionar recursos y prioridades a los distintos componentes y actividades del proyecto. - Coordinar las iteraciones entre el equipo de trabajo y los usuarios del proyecto. - Mantener al equipo del proyecto enfocado en los objetivos. - Establecer un conjunto de prácticas que aseguren la calidad e integridad del proyecto. - Supervisar el desarrollo del proyecto. <p>Como Analista de Sistemas:</p> <ul style="list-style-type: none"> - Capturar la especificación y validación de requisitos interactuando con los usuarios. - Elaborar el Análisis y Diseño del Sistema.

	<p>Como Programador:</p> <ul style="list-style-type: none"> - Realizar la Programación del Sistema. - Elaborar las pruebas funcionales del Sistema. <p>Como Capacitador:</p> <ul style="list-style-type: none"> - Formar al personal en el uso de las TIC para el manejo del producto final.
Tutor	<ul style="list-style-type: none"> - Asesoramiento en los aspectos tecnológicos para el desarrollo del Proyecto. - Asesoramiento en el uso de las Metodologías de desarrollo. - Evaluación del documento del proyecto.

Tabla 4 Actividades para los integrantes del equipo de investigación

1.2.4.1 Unidades de Gestión: Organigrama del equipo del proyecto

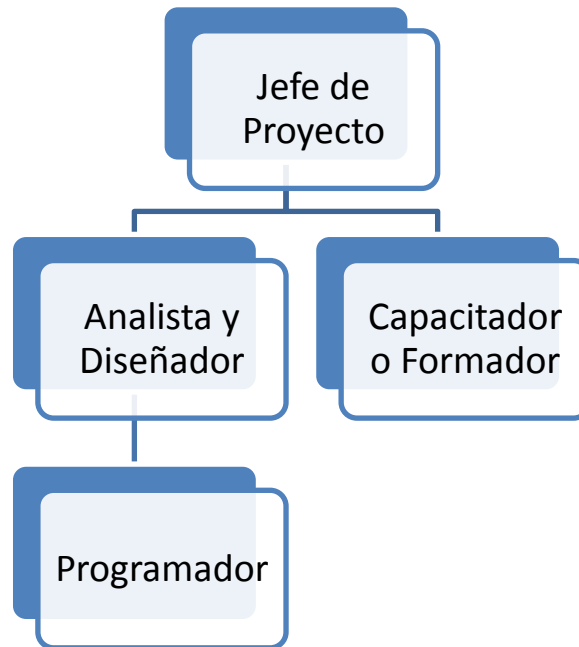


Figura 1 Organigrama del Equipo de Proyecto

1.3 Descripción del Proyecto

1.3.1 Resumen ejecutivo del proyecto

En los últimos años se han incorporado en nuestro entorno numerosos avances tecnológicos que han inundado hogares y oficinas. Son demasiadas aportaciones a la sociedad del bienestar para predecir un retroceso. La preocupación permanente por la mejora de la administración ha conducido a la rápida adopción de sistemas automáticos capaces de facilitar tareas mecánicas y rutinarias, evitar errores y mejorar el control.

En un mundo de constante cambio donde la era de la globalización y el avance tecnológico se hace más evidente, las instituciones públicas no puede quedar al margen de esta situación. Más aún, si este desea transparentar sus actos administrativos y mantener una mejor organización de la información de sus procesos.

El concepto de las TIC se definen como sistemas tecnológicos mediante los que se recibe, manipula y procesa información, y que facilitan la comunicación entre dos o más interlocutores. Por lo tanto las TIC son algo más que informática y computadoras puesto que funcionan en conexión con otras entidades mediante una red. También son algo más que tecnologías de emisión y difusión (como televisión y radio) puesto que no solo dan cuenta de la divulgación de la información, sino que además permiten una comunicación interactiva, donde el usuario goza de diferentes opciones.

Describiendo los importantes avances de los sistemas y tecnología, como también las TIC, mediante estas nuevas tecnologías se permite tener una mejor administración de la información y poder fortalecer el control y la supervisión de los documentos, viendo que cada día el avance tecnológico es constante, el siguiente proyecto presenta la propuesta del desarrollo de un sistema web para un mejor control de la información, en el cual se encargue de la administración de los documentos teniendo una información segura y confiable que beneficiará a toda la población y funcionarios de la Brigada Parlamentaria de Tarija.

El objetivo principal del presente proyecto es Desarrollar e Implementar un Sistema Automatizado para el control de los documentos internos o externos de la Brigada Parlamentaria de Tarija. Además de plantear una socialización de los modelos que permitan crear un sentido de pertenencia para lograr los objetivos propuestos.

1.3.2 Descripción, Fundamentación y Justificación del Proyecto

A pedido de la Presidenta de la Brigada Parlamentaria de Tarija se analizó la problemática y las posibles soluciones para el control de documentación y la comunicación entre los miembros de la Brigada.

En la gestión de documentos, los procesos se realizan de manera manual en unos cuadernos, esto resulta en acumulación de papeleo y causa un difícil seguimiento de la información en la Brigada Parlamentaria de Tarija

Los documentos físicos son almacenados sin usar un modelo de Inventario Documental.

El Gobierno Nacional a través de la AGETIC promueve la aplicación de Tecnologías de la Información y Comunicación en Instituciones Públicas, generando conocimiento, mejorando el servicio público y facilitando la participación para el bienestar de la población. Tomando tres aspectos fundaméntales:

- Simplificación de Trámites
- Datos Abiertos
- Transparencia, Participación y Control Social

En el Reglamento de Brigadas Departamentales Artículo 23 (SISTEMA DE INFORMACION Y DOCUMENTACION) dice lo siguiente: Las brigadas tendrán acceso irrestricto a toda información digital o física de la Asamblea Legislativa, como así también tendrá su propio sistema de conservación de su documentación y archivo.

Para resolver el problema del control de documentación el Presidente y la parte Administrativa de la Brigada Parlamentaria de Tarija decidieron contar con un sistema computarizado para dicho control y para mejorar la parte de la comunicación entre los miembros de la Brigada se sugiere una Capacitación en el área de las TIC.

El proyecto tiene como propósito mejorar la gestión de información que se produce en la Brigada Parlamentaria. Las potencialidades y uso de las TIC son vertiginosos en cuanto a la interactividad, facilidad de la información y comunicación a distancia, numerosas posibilidades colaborativas, accesibilidad de información, etc.

El desarrollo de un Sistema web para un adecuado manejo de la información de documentos de la Brigada Parlamentaria de Tarija aporta al manejo de datos del documento de forma oportuna, ordenada y disponible para realizar reportes.

Las TIC son medios que aportan un flujo ininterrumpido de información, que es esencial para los estilos de vida cotidiana de cada uno de nosotros. Es un instrumento eficaz de comunicación y de expresión que conlleva cambios que alcanzan todos los ámbitos de la actividad humana. Sus efectos se manifiestan de manera muy especial en las actividades laborales.

Con esta herramienta se pretende cambiar la forma de manejar la información que se maneja con lo que respecta la gestión de información de los recursos de la institución, sin dejar atrás las normas y formalidades de trabajar con ellas.

1.3.2.1 Análisis de causas de problemas

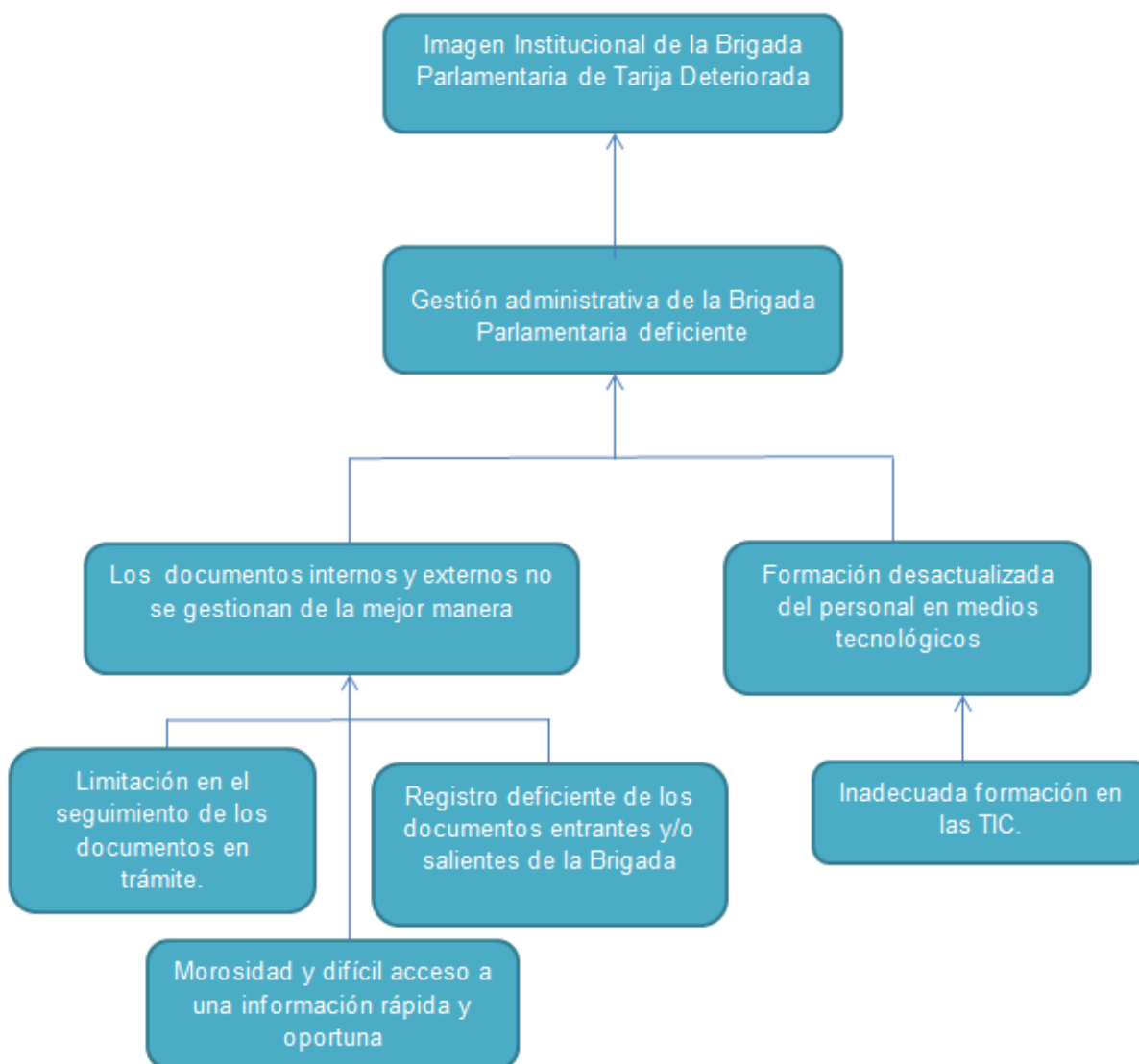


Figura 2 Árbol de problema

1.3.2.2 Análisis de Objetivos

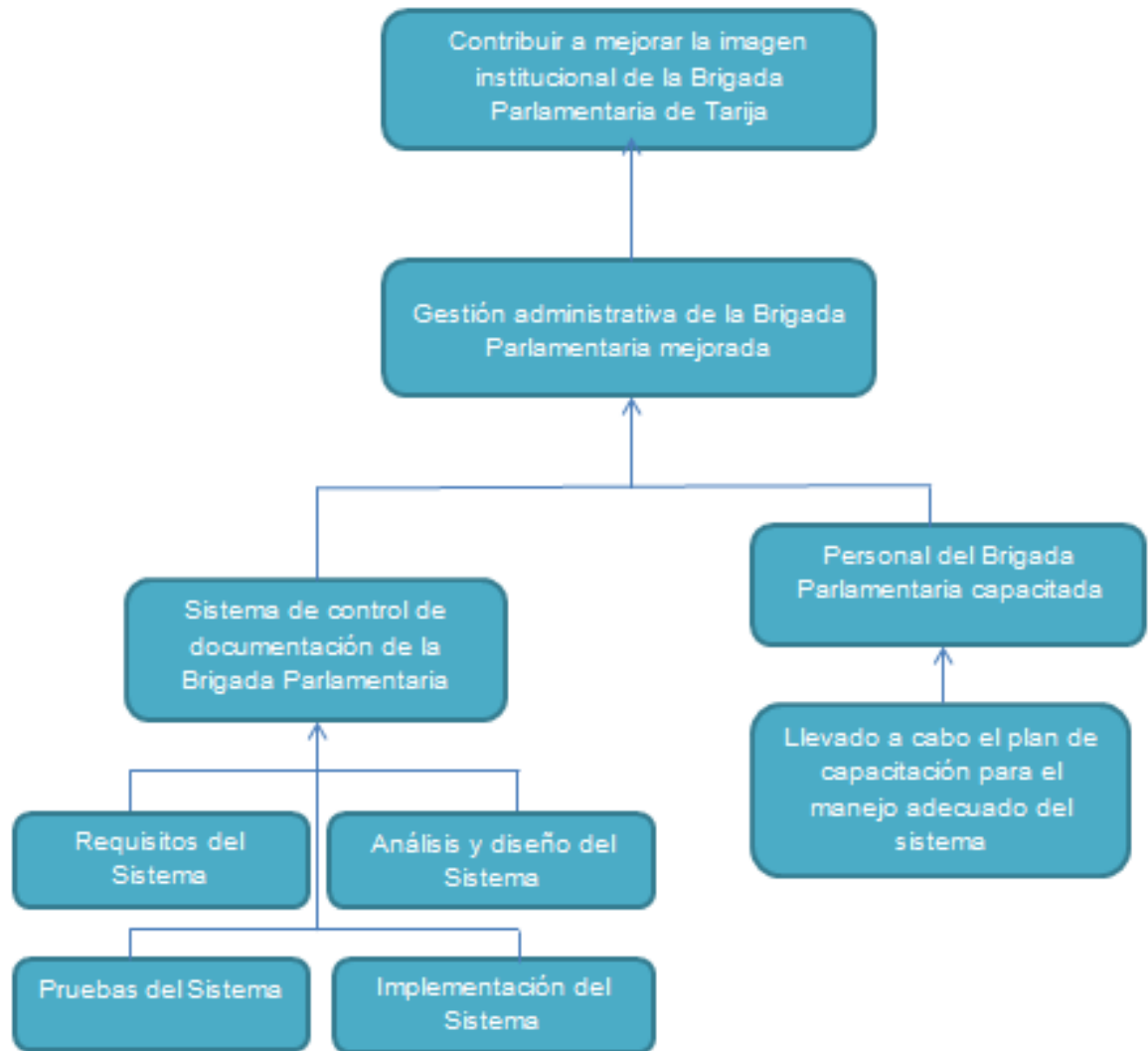


Figura 3 Árbol de Objetivo

1.3.3 Objetivos

1.3.3.1 Objetivo General

Mejorar la gestión documental en la Brigada Parlamentaria de Tarija, a través de las TIC desarrollando e implementando un Sistema Automatizado

1.3.3.2 Objetivo Específico

- Desarrollar un sistema automatizado para un adecuado manejo de documentos en la Brigada Parlamentaria de Tarija.
- Implementar programas de socialización para el manejo y conocimiento del sistema automatizado para poder obtener un mejor nivel de conocimientos en el uso de las nuevas tecnologías.

1.3.4 Metodología, Tecnologías y Herramientas

La metodología adapta para el desarrollo de este proyecto consiste de:

1.3.4.1 Metodología para el desarrollo de las aplicaciones

El Sistema de Gestión de control de documentación del Birgada Parlamentaria de Tarija se desarrollara bajo la *metodología RUP* (Rational Unnified Process) que es un estándar para el análisis, implementación, desarrollo y documentación de sistemas orientados a objetos junto con el lenguaje UML (Unified Modeling Language).

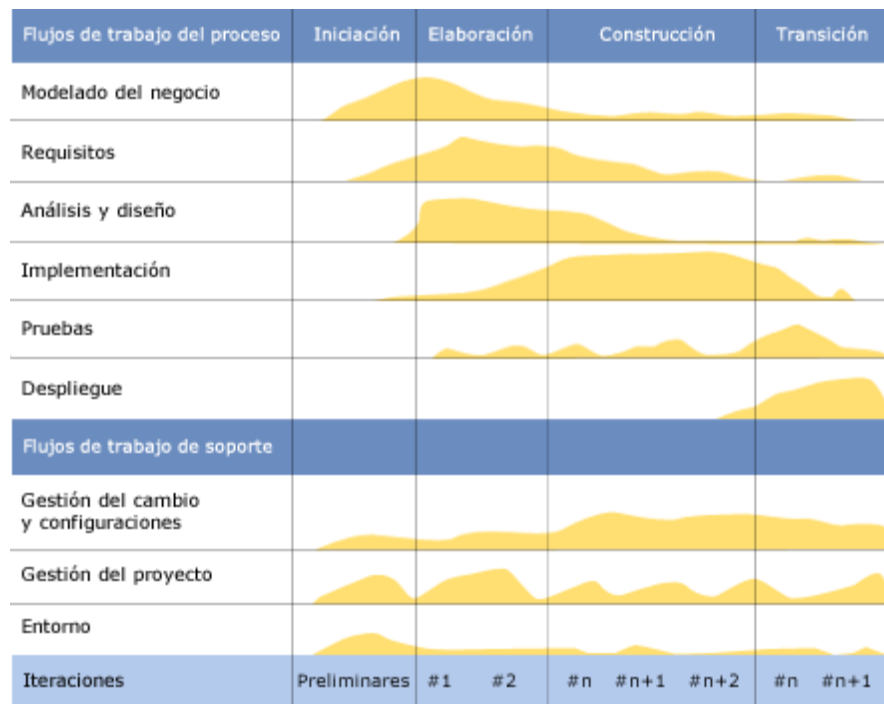


Figura 4 Ciclo de Vida de RUP

El ciclo de vida de RUP es de 4 fases y se implementa el desarrollo en espiral, es decir que se realizan varias iteraciones en cada fase. El número de las iteraciones varían de acuerdo al proyecto.

Las fases de RUP son las siguientes:

- **Inicio:** En esta fase se hace mayor énfasis en las actividades de identificación de requerimientos, riesgos y se modelan los principales casos de uso.
- **Elaboración:** En esta etapa se analizan los requisitos, se desarrolla un prototipo de arquitectura incluyendo las partes más relevantes sistema y se completan los diagramas de Casos de Uso.
- **Construcción:** En esta fase se desarrolla el producto mediante una serie de iteraciones, en las cuales se seleccionan algunos casos de uso, se refina su análisis y diseño y luego se realiza la implementación y sus pruebas. El número de iteraciones depende de que se termine con la implementación de la nueva versión del producto.

- **Transición:** En esta fase se instala el producto y se lo entrega a los usuarios con lo que pueden surgir nuevos requisitos.

UML es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso de construcción del mismo.

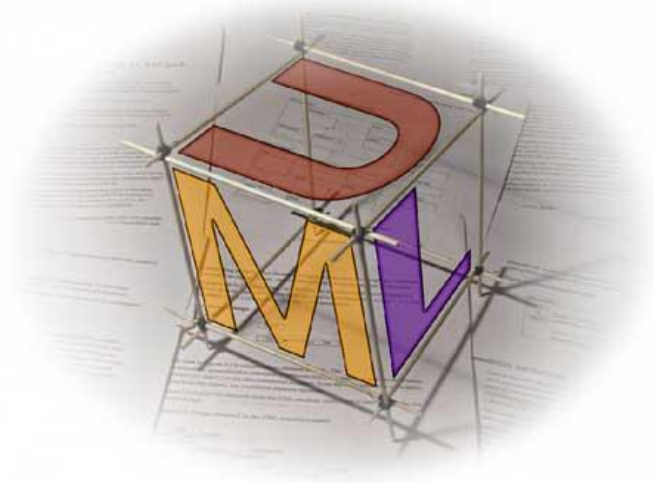


Figura 5 Logotipo de UML

Capas y Tecnologías:

El objetivo principal de la arquitectura seleccionada para el desarrollo de este proyecto, es separar, de la forma más limpia posible, las distintas capas de desarrollo, con especial atención a permitir un modelo de dominio limpio y a la facilidad de mantenimiento y evolución de las aplicaciones.

Para lograr esto se eligió el patrón MVC (Modelo-Vista-Controlador) que permite una separación limpia entre las distintas capas de una aplicación.

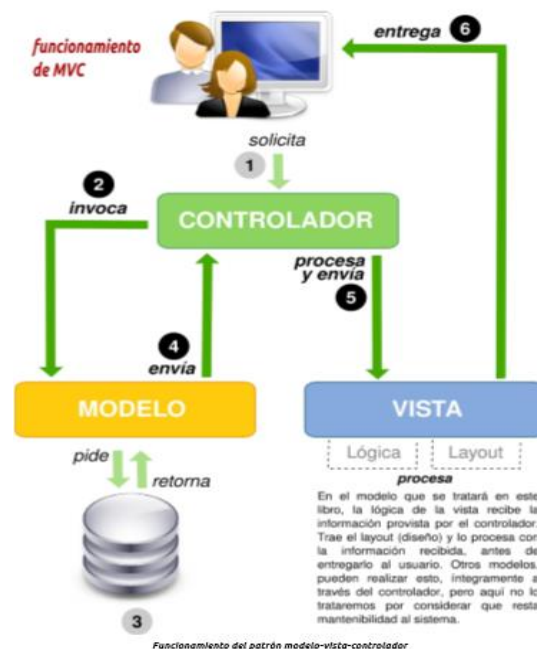


Figura 6 Patrón Arquitectónico MVC

1.3.4.2 Tecnologías para el desarrollo de las aplicaciones

1.3.4.2.1 Tecnologías en el Servidor

Si nos disponemos a realizar una aplicación web, lo primero que habremos de hacer será elegir las tecnologías que usaremos para llevar a cabo el desarrollo. En este punto nos daremos cuenta de que las principales alternativas que nos encontramos son las siguientes:

- JAVA
- .NET
- PYTHON
- RUBY
- PHP
- JAVA y .NET

Tanto en Java como en .Net las curvas de aprendizaje que tiene que sobrellevar el programador inicialmente son bastante fuertes. El caso en el que Java y .Net muestran

su potencial completamente y marcan una gran diferencia con respecto a sus competidores, es en el desarrollo de aplicaciones empresariales.

Nos encontramos por lo tanto, que si lo que nos traemos entre manos es el desarrollo de una aplicación web para una pyme, , una tienda on-line, un blog, o una web personal con cierta funcionalidad, la elección de Java o .Net (aunque sea completamente válida) se nos queda un poco grande, porque no estaremos aprovechando su verdadero potencial, tendremos que hacer frente a un periodo de aprendizaje duro, y nos encontraremos con un proceso de desarrollo considerablemente lento.

En el caso de elegir tecnologías .Net, además, tendremos que tener en cuenta el gasto generado por el licenciamiento de un servidor Windows que aloje la aplicación y tal vez el del gestor de base de datos (en caso de elegir alguna alternativa de pago como SQL Server o similar).

Python, Ruby y PHP (entre otros) son tecnologías de desarrollo muy ágiles, cuyas curvas aprendizaje son muy suaves, y que cubren con solvencia la mayoría de los casos que nos vamos a encontrar. Y aunque «en teoría» no puedan competir con Java y .Net en el desarrollo de grandes aplicaciones, toda regla tiene su excepción, y en este caso nos encontraremos con una lista de excepciones bastante interesante.

También conviene pensar en las facilidades que nos encontraremos a la hora de poner en producción una aplicación desarrollada en php, ruby o Python, ya que existen multitud de servicios de alojamiento (gratuitas y de pago) que disponen de servidores con las especificaciones necesarias para soportar este tipo de aplicaciones.

¿Entonces, sirve o no sirve PHP para desarrollar grandes aplicaciones?

Existe la creencia de que PHP no es válido para desarrollar grandes aplicaciones que tengan que soportar un gran número de usuarios concurrentes y que tenga que cumplir con la necesidad estar en alta disponibilidad. Se suele decir que «una imagen vale más que mil palabras», pero en este caso sería más adecuado decir que «un

ejemplo vale más que mil palabras», y ese ejemplo es nada más y nada menos que Facebook.

Como lenguaje de programación, PHP es simple. Fácil de aprender, fácil de escribir, fácil de leer y fácil de depurar. Somos capaces de conseguir nuevos ingenieros mucho más rápido con PHP que con otros lenguajes, lo que nos permite innovar más rápido.

Otra característica a tener en cuenta sería la conectividad. PHP dispone de una amplia gama de librerías, y agregarle extensiones es muy fácil. Esto le permite al PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML y otras.

Algunas contras:

El manejo de errores no es tan sofisticado como Cold Fusión o ASP.

No existe IDE o Debugger. Una IDE puede no ser importante para la mayoría de los programadores y un debugger ha sido prometido por Zend Tech para un futuro muy cercano.

1.3.4.2.2 Tecnologías en el Cliente

La web es un sistema de distribución de información basada en hipertexto o hipermedia accesible a través del internet.

Un elemento esencial de la web es el HTML el lenguaje de Marcado que se emplea para crear las páginas web.

HTML permite describir la estructura y el contenido de una página web, aunque permite incluir otros objetos como imágenes y videos.

En el proyecto aplicaremos HTML5 que simplemente es la quinta y última versión por ahora del lenguaje de etiquetado de HTML, sin embargo HTML5 no solo es eso sino algo más. Pero las páginas web no solo se desarrollan con HTML, también se emplean otras tecnologías:

- HTML
- CSS (Hojas de estilo en cascada)

- DOM (Jerarquía de objetos)
- JAVASCRIPT (Lenguaje interpretado)

Las últimas tres unidades se las considera DHTML que simplemente es HTML dinámico que permite crear sitios web interactivos. Además del DHTML existen otras tecnologías:

- DHTML
- XHR (API de programación)
- XML (Meta lenguaje)
- JSON (Formato de intercambio Cliente-Servidor)

Los últimos tres combinados se los considera AJAX una técnica de desarrollo web que permite crear aplicaciones interactivas que se ejecutan en el navegador.

1.3.4.3 Herramientas para el desarrollo de las aplicaciones

1.3.4.3.1 Herramientas para el Análisis y diseño

Las herramientas CASE se han venido ampliando y desarrollando constantemente, existe una gran variedad de estas con características específicas, a continuación describiremos algunas de ellas, para llegar a la que más convenga para el proyecto.

- **Racional Rose:** es una herramienta de producción y comercialización establecidas por Rational Software Corporation (actualmente parte de IBM). Rose es un instrumento operativo conjunto que utiliza el Lenguaje Unificado (UML) como medio para facilitar la captura de dominio de la semántica, la arquitectura y el diseño.

Sus características principales:

- No es gratuito, se debe hacer un previo pago para poder adquirir el producto.
- La ingeniería de código (directa e inversa) es posible para ANSI C++, Visual C++, Visual Basic 6, Java, J2EE/EJB, CORBA, Ada 83, Ada

95, Bases de datos: DB2, Oracle, SQL 92, SQL Server, Sybase, Aplicaciones WEB.

- Solamente Ingeniería reversa para COM.

- **Visual Paradigm:** es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Presenta licencia comercial. Es fácil de instalar y actualizar y compatible entre ediciones.

Características principales:

- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
 - Modelado colaborativo con CVS y Subversion (control de versiones).
 - Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI.
 - Ingeniería de ida y vuelta.
 - Ingeniería inversa - Código a modelo, código a diagrama.
 - Ingeniería inversa Java, C++, Esquemas XML, XML, NET exe/dll, CORBA IDL.
 - Diagramas EJB - Visualización de sistemas EJB.
 - Generación de código y despliegue de EJB - Generación de beans para el desarrollo y despliegue de aplicaciones.
- **Enterprise Architect:** es una herramienta CASE de Sparx Systems. Soporta ocho de los nueve diagramas estándares del UML: diagrama de casos de uso, de clases, de secuencia, de colaboración, de actividad, de estados, de implementación (componentes), de despliegue y varios perfiles del UML. Si fuera necesario, el diagrama de objetos se puede crear usando los diagramas de colaboración.

Enterprise Architect tiene un mecanismo de perfil UML genérico para cargar y trabajar con diferentes perfiles UML. En Enterprise Architect, estos perfiles

se especifican en archivos XML con un formato específico. Los perfiles disponibles son:

- Modelado de Procesos de Negocio: Soporta las extensiones de modelado de procesos de negocio de Eriksson-Penker.
- Modelado de Datos.
- Modelado de la Interfaz de Usuario.
- Modelado Web.
- Esquema XSD
- Permite ingeniería de código (directa e inversa) para ANSI C++, Visual Basic 6, Java, PHP, C#, VB.NET, Delphi y Bases de datos: Ingeniería directa desde el modelo de datos al script DDL. La ingeniería reversa usa la fuente de datos ODBC.
- La forma en la que EA trabaja es generando los archivos de código fuente de las clases para aquellas que correspondan al mismo paquete. Adicionalmente, se pueden aplicar los patrones de diseño, el usuario tiene que crear los patrones.

Esta última considerando que la mayoría son con licencia es la de menor limitación de uso a la hora de usarla sin licencia en un determinado tiempo de prueba.

1.3.4.3.2 Herramientas para la Implementación

1.3.4.3.2.1 Herramienta de trabajo

A la hora de la implementación hay que escoger una herramienta de trabajo que bien puede ser un IDE o simplemente un Editor de texto pero como elegir el apropiado, vamos a realizar un análisis breve entre los dos:

IDE.- Es un entorno de desarrollo integrado, es un programa pero, no un simple programa, sino un software especial que trae todo lo necesario para facilitarle lo más posible tu tarea de escribir software, entre las cosas que encontraras en un IDE están:

- Creador de interfaces gráficas
- Completado de código

- Resaltado de sintaxis
- Ayuda contextual
- Sugerencias para codificar mejor
- Plugins para incorporar otras herramientas (un control de versiones por ejemplo)
- Compilar y ejecutar el código

Editor de texto.- Un editor de texto es un programa que permite crear y modificar archivos digitales compuestos únicamente por texto sin formato, conocidos comúnmente como archivos de texto o texto plano.

Por eso son pequeños programas, ligeros, que no consumen muchos recursos y algunos hasta se ejecutan solamente en una terminal de tu sistema operativo. Estas son las características que tiene un editor de texto común

- Buscar y reemplazar
- Copiar, cortar y pegar
- Deshacer y hacer
- Formatear texto
- Seleccionar texto

Ahora que vimos lo que nos ofrece cada herramienta, podemos tomar la mejor decisión de cual utilizar, y la respuesta correcta siempre será: depende.

Depende de la aplicación que estés realizando, depende del lenguaje de programación que utilices, depende de tu experticia en dicho lenguaje, depende de la comodidad que como programador sientas con alguna herramienta. Pero existen algunos criterios que hay que tomar en cuenta para elegir un IDE o un Editor de texto.

Utiliza un IDE cuando:

- El lenguaje de programación sea compilado, Java por ejemplo, eso de estar compilando cada clase por tu cuenta cuando es un proyecto enorme es un gran problema, deja que un IDE lo haga por ti.
- Estas diseñando una interfaz gráfica de usuario, nada mejor que un IDE con un bonito drag and drop, para solo soltar los elementos en el lugar que quieras que salgan.
- Estas aprendiendo un lenguaje nuevo, no sabes la sintaxis, ni los métodos ni nada. La ayuda y completado de código del IDE hará tu aprendizaje más fácil y amistoso.
- Tienes una computadora con suficientes recursos.

Utiliza un Editor de texto cuando:

- Tengas una computadora con muy pocos recursos
- El lenguaje de programación que utilices no sea compilado (php, javascript, ruby, etc)
- Estas creando sitios web dinámicos.
- Eres experto en el lenguaje que estas programando, tan experto que solo con tener la documentación a la mano es suficiente para hacer maravillas.

Tomando en cuenta lo anterior optaremos por un Editor de texto, ahora analizaremos cual es el mejor que se adapte a nuestro entorno de desarrollo.

- Brackets
- Notepad++
- EditRocket
- Sublime Text

Brackets es un editor de texto open source orientado al diseño web. Brackets es mantenido principalmente por Adobe, y está escrito en HTML, CSS y JavaScript, esta liberado bajo una licencia MIT. Se integra con el navegador y cuenta con pre visualización en vivo del HTML y del CSS, y de JavaScript cada vez que guardas los

cambios en tu documento. También cuenta con resaltado, sugerencias y autocompletado de tags, propiedades y valores de la sintaxis.

Brackets es multiplataforma y puedes instalarlo en Windows, OS X y Linux.

Notepad++ es un editor de texto open source para Windows y también es un perfecto reemplazo para Notepad (el simplón bloc de notas de Microsoft). Está bajo una licencia GPL. Notepad++ se basa en el componente de edición de código Scintilla, está escrito en C++ y es sumamente ligero. Cuenta con resaltado de sintaxis, búsqueda, interfaz personalizable, mapa del documento, autocompletado, pestañas para abrir múltiples documentos al mismo tiempo, etc.

EditRocket es un editor de texto y código fuente, que soporta múltiples lenguajes, con funciones básicas de autocompletado, búsqueda y resaltado de sintaxis. Además integra su propio cliente FTP y SFTP, corrector ortográfico y un convertidor de texto a HTML.

EditRocket puede ser instalado en Windows, Linux o Mac gratuitamente por un periodo de prueba de 30 días, posteriormente requiere la compra de una licencia para poder seguir siendo usado.

Sublime Text es una de los editores más populares y completos en la actualidad. Tiene una de las mejores interfaces que existen, con la posibilidad de dividir tu pantalla en grupos de trabajo, y editar varios documentos lado a lado, aprovechando al máximo las pantallas múltiples.

Sublime Text tiene un gran rendimiento y muchas posibilidades de personalización. Soporte para una infinidad de lenguajes de programación, y una enorme selección de plugins a través de su API basada en Python.

Sublime Text es multiplataforma y puedes descargarlo para Linux, Windows y Mac. Aunque es una aplicación de pago, puedes descargar la versión de prueba sin límite de tiempo, con el único inconveniente de que te recordará cada cierto tiempo que compres la licencia del producto pero no limita ninguna de sus funciones.

Esta última tomando en cuenta sus características es la que más se adapta a nuestro proyecto por sus características.

1.3.4.3.2.2 Herramienta para crear un servidor local

Si se está trabajando en una aplicación o página web a modo local (sin conexión a internet) pero emulando un entorno online, tenemos que elegir una herramienta para crear un servidor local.

Las herramientas que analizaremos tendrás incluidas las tecnologías Apache, Mysql, PHP y PhpMyAdmin, de esta manera podremos trabajar con páginas que tengan las tecnologías seleccionadas anteriormente que son PHP, HTML, CSS y Java script, sin necesidad de instalar más de una aplicación o utilidad.

- easyPHP
- Wamp Server
- XAMPP

Comenzaremos con **easyPHP**, un pack completo que incluye Apache, PHP, MySQL, PHPMyAdmin y SQLiteManager, con el cual se podrá desarrollar en PHP.

Desde el panel de control de la herramienta se podrá personalizar las utilidades (iniciando y deteniendo los servicios), y también tendrás acceso a los log.

Con esta aplicación se podrá correr páginas tipo foro o con panel de administración.

easyPHP es una herramienta gratuita para sistemas operativos Windows.

Wamp Server es una herramienta con la cual, a la hora de instalar, estarás incorporando de forma automática los siguientes componentes: servidor web Apache, entorno PHP5, base de datos MySQL, y los gestores PHPmyadmin y SQLitemanager.

Con Wamp Server se dispondrá de una barra del sistema, para acceder a la carpeta donde tiene alojada tu o tus páginas web, ficheros log, configuración de los servicios web Apache y base de datos MySQL.

Wamp Server es un servidor local muy estable y con muchas prestaciones.

XAMPP un GPL disponible para Windows, Mac OS X, Linux y Solaris. Te permite tener de forma simple y local Apache, MySQL, PHP y phpMyAdmin.

Con este potente pack de instalación se podrá desarrollar páginas web en un entorno seguro e idéntico al online.

La utilidad más importante de la herramienta es su panel de control (de diseño propio) pero también se destacan PEAR, MiniPerl, mod_ssl, OpenSSL, PHPMyAdmin, Webalizer, FileZilla FTP Server, SQLite, Zend Optimizar, Mercury Mail.

XAMPP es un servidor local muy potente que te ofrece una gran colección de librerías, una base de datos MySQL, un servidor de correo, servidor FTP. Y las últimas versiones de MySQL y PHP; Apache y MySQL.

XAMPP es idóneo para trabajar con páginas web, especialmente CMS.

Como se puede apreciar la mejor elección para este proyecto es XAMPP.

1.3.4.4 Entorno de Trabajo para el desarrollo de las aplicaciones

Un Framework es un entorno o ambiente de trabajo para desarrollo; dependiendo del lenguaje normalmente integra componentes que facilitan el desarrollo de aplicaciones como el soporte de programa, bibliotecas, plantillas y más.

Lo más importante de todo este entorno de trabajo es que contiene máquinas virtuales, compiladores, bibliotecas de administración de recursos en tiempo de ejecución y especificaciones de lenguajes, haciendo nuestro trabajo más eficiente y recursivo.

La arquitectura más utilizada en casi todos los frameworks es conocida como MVC (Controlador, Modelo, Vista), esta arquitectura divide el desarrollo en tres grandes partes:

- **Modelo:** Son los datos de la aplicación y su reglamentación.
- **Vista:** Es la presentación de los datos.

- **Controlador: Procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.**

Existen muchos Framework por eso más que difícil es casi imposible cuantificarlos y su uso va de acuerdo a las necesidades de cada proyecto y al gusto del freelance y/o desarrollador; algunos de ellos pueden ser:

- .NET: El Framework de Microsoft.
- Ruby on Rails (RoR): es el Framework de aplicaciones web de código abierto de un lenguaje de programación llamado Ruby.
- Genesis: Framework para WordPress
- Zengrid: Framework para Joomla
- PhoneGap: Permite crear aplicaciones móviles multiplataforma a partir de tecnologías web como HTML5, CSS3 y Javascript.
- CodeIgniter: Permite el desarrollo web MVC escrito en PHP de última generación
- JQuery Mobile: basado en HTML5 optimizado para todas las plataformas de dispositivos móviles.
- Titanium: Framework Open Source con licencia Apache, genera aplicaciones nativas para iPhone, Android y Blackberry.

La lista se haría interminable por tal motivo opinar sobre cuál es el mejor framework para desarrollar es verdaderamente inútil ya que los encuentras incluso tanto para desarrolladores expertos como para principiantes.

Elegir un framework va de acuerdo a la necesidad del proyecto a desarrollar así como también la decisión de no usarlos, consideración que valdría la pena reevaluar si tenemos en cuenta que el desarrollo con framework es menos propenso a sufrir errores. Lo que sí puedo decir es que hoy día la utilización de frameworks es casi del 100% porque permite a los desarrolladores freelance y/o autónomos ser más productivos, es decir más proyectos, más ingresos y menos tiempo de trabajo.

Como otros muchos framework MVC, **CodeIgniter** fue diseñado siguiendo la filosofía que prima la convención sobre la configuración.

CodeIgniter tiene una forma rígida como se debe escribir, como llamar a los modelos, vistas y controladores por ejemplo, pero siguiéndolas hace que el framework ya sepa en muchas ocasiones que es lo que debe hacer sin necesidad de escribirlo el código. CodeIgniter aporta mejoras, aumenta su inteligencia respecto a otros framework MVC, hace mucho más flexible y adaptarse prácticamente a cualquier proyecto que se quiera realizar.

1.3.4.5 Metodología para las Capacitaciones y/o Talleres

Las capacitaciones son muy importantes para el proyecto porque permiten acelerar el proceso de aprendizaje de los usuarios, adquirir habilidades y ayudar al usuario a entender el funcionamiento del Sistema de gestión de documentos.

Para este componente se eligió el modelo ACER. Un modelo para elaborar una sesión de capacitación.

El modelo propuesto es un modelo sistémico; todas las variables identificadas están relacionadas entre sí. Aunque el modelo se presente y se explique por etapas no se trata de un modelo lineal. Este modelo posee tres dimensiones principales, que son:

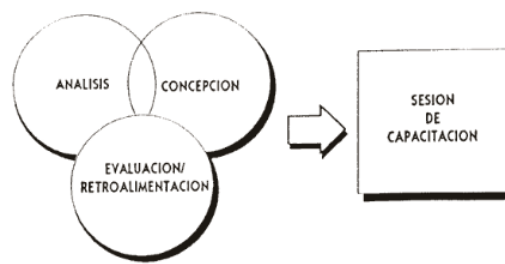


Figura 7 Modelo ACER

- Análisis
- Concepción
- Evaluación –
- Retroalimentación

Para poder aplicar este modelo hay que comprender cada una de sus dimensiones y captar de qué forma se enriquecen entre sí.

1.3.5 Matriz de Marco Lógico

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Fin Contribuir a mejorar la imagen institucional de la Brigada Parlamentaria de Tarija	Al final del 1° año de ejecución del proyecto, el 51% de la población atendida por la brigada parlamentaria de Tarija manifiesta un alto grado de satisfacción por los servicios de la Institución $\frac{(51 \text{ afirman que el trabajo mejoro}}{100\%)*100$ 100 encuestas realizadas	Resultado de la encuesta anual realizada por el Secretario Técnico de la Institución, referente al trabajo que desempeña la Brigada Parlamentaria.	La Institución cuenta con medios tecnológicos suficientes para el desarrollo de sus funciones.
Objetivo General (Propósito) Gestión administrativa de la Brigada Parlamentaria de Tarija, mejorada	Al concluir con el proyecto el 60% de los procesos asociados la Brigada Parlamentaria se han automatizado, a partir del 2018 con relación al año base 2017. $\left(\frac{\text{NroPerosnasporHora}}{\text{NroPerosnas porDia}}\right) \times 100$	Cuadro comparativo de la cantidad de procesos del control de documentación vs. Cantidad de procesos automatizados avalado por la Presidenta de la Brigada Parlamentaria.	Aceptación de la implementación del sistema automatizado por parte de la Presidenta máxima autoridad de la Brigada Parlamentaria
Objetivos Específicos (Componentes) 1. Sistema de control de documentación de la Brigada	Al finalizar el proyecto se ha desarrollado el sistema certificado por los estándares en desarrollo de aplicaciones web y que cumplan	Carta de conformidad firmada por la Docente de la Materia Taller III	El personal involucrado proporciona toda la información necesaria para una correcta recolección de datos

<p>Parlamentaria de Tarija</p> <p>2. Socialización del sistema automatizado y las TIC ejecutado e implementado</p>	<p>con la norma IEE830.</p> <p>Al terminar el proyecto el personal involucrado se encuentra capacitado un 75% para el manejo del sistema web.</p> <p>(26PersonasCapacitadas/35Total Personas) X 100%</p>	<p>Fotos de la capacitación.</p> <p>Planilla de asistencia con firmas de los participantes.</p> <p>Certificados de la capacitación.</p>	<p>Coordinación entre el equipo de trabajo del proyecto y el usuario final.</p> <p>Los miembros de la brigada Parlamentaria apoyan para realizar la capacitación.</p> <p>Se dispone con ambientes y material necesario para la capacitación.</p>
<p>Actividades</p> <p>Componente 1:</p> <p>Determinación de Requerimientos</p> <p>Análisis y Diseño del Sistema</p> <p>Desarrollo del Sistema(implementación)</p> <p>Ejecución de Pruebas</p> <p>Componente 2:</p> <ul style="list-style-type: none"> - Análisis - Concepción - Evaluación - Retroalimentación 	<p>De acuerdo al presupuesto/justificación del proyecto.</p>	<p>Informes del presupuesto de gastos.</p> <p>Cronogramas e informes de la capacitación.</p> <p>Manuales de usuario y de sistema.</p> <p>Testimonios de los involucrados en el desarrollo del sistema.</p>	<p>Se dispone de hardware de última generación.</p> <p>Se acepta el presupuesto necesario para la implementación del sistema automatizado.</p> <p>Uso apropiado de los manuales de usuario.</p> <p>Uso apropiado de la socialización por parte de los involucrados</p>

1.4 Presupuesto / Justificación

Estimaciones del Proyecto con partida de la gestión 2012.

ITEM	RUBROS	Aporte Universidad	Otro Aporte	TOTAL (Bs.)
20000	SERVICIOS NO PERSONALES			
	21000. Servicios Básicos			420
	22000. Servicios de transporte			100
	25000. Servicios Profesionales y Comerciales			12600
	Sub total rubro			13120
30000	MATERIALES Y SUMINISTROS			
	31000. Alimentos y Productos Forestales			80
	32000. Productos de Papel, Cartón e Impresos			180
	39000. Productos Varios.			50
	Sub total rubro			310
	TOTAL			13430
	TOTAL + 40% Incentivo			

Tabla 5 Presupuesto / Justificación

1) GRUPO 20000. SERVICIOS NO PERSONALES

b) SUB GRUPO 21000. Descripción de los gastos de servicios básicos

Partida	Tipo de servicio básico *	Costo	Tiempo mes	Costo Total
21100	Comunicación	20	3	60
21200	Energía Eléctrica	15	8	120
21400	Servicios Telefónicos	30	8	240
Total				420

* Se refiere principalmente a los gastos por servicios; como: servicio de correo, radiogramas, servicio telefónico, fax, Internet.

c) SUB GRUPO 22000. Descripción de los gastos de viajes y transporte de personal

Partida	Personal	Lugar	Duración (días)	Costo unitario*	Costo total
22600	Transporte de Personal	Tarija	50	2	100
Total					100
Total sub grupo 22000					100

* En el caso de los viáticos, debe considerarse la escala establecida por la UAJMS.

f) SUB GRUPO 25000. Descripción de los gastos en servicios profesionales y comerciales

Partida	Tipo de servicio profesional y comercial *	Cantidad	Costo unitario	Tiempo mes	Costo total
25200	Estudios e Investigaciones				12000
25600	Imprenta				600
Total					12600

* Se refiere a gastos por servicios profesionales de asesoramiento especializado, se incluyen, estudios, investigaciones, publicidad, imprenta, fotocopias, capacitación de personal y otros ejecutados por terceros.

2) GRUPO 30000. MATERIALES Y SUMINISTROS

g) SUB GRUPO 31000. Descripción de los gastos Alimentos y Productos Agroforestales

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
31110	Refrigerios y Gastos Administrativos	10	8	80
Total				80

* Se refiere a la adquisición de materiales y bienes como: alimentos y productos agroforestales, alimentos y bebidas para personas (indicar el total de refrigerios), alimentos para animales, productos pecuarios.

h) SUB GRUPO 32000. Descripción del gasto de Productos de Papel, Cartón e Impresos

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
32100	Papel de Escritorio	6	30	180
Total				180

* Se refiere a la adquisición de; papel y cartón en sus diversas formas y clases, impresos y publicaciones, periódicos, revistas, libros, fotocopias, etc.

k) SUB GRUPO 39000. Descripción del gasto en productos varios

Partida	Productos de cuero y caucho	Cantidad	Costo/Unitario	Total
39500	Útiles de Escritorio y de Oficina			50
Total				50

*Se refiere principalmente a los gastos por productos de limpieza, todo lo referente a la funcionamiento de la oficina en material de escritorio.

1.4.1 Curriculum Viate

1.4.1.1 Antecedentes personales

Huanca Apellido Paterno	Mamani Apellido Materno	Antonio Lujan Nombre	7177505 Tja C.I.
22/09/1990 Fecha de Nacimiento	M...X... F..... Sexo	B/ Senac C/Lomas Patriotas Nro. 0432 Dirección	

Tabla 6 Antecedentes Personales

1.4.1.2 Antecedentes académicos

Título obtenido	Universidad	País	Año
Ing. Informática - 1º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2009

Ing. Informática - 2º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2010
Ing. Informática - 3º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2010
Ing. Informática - 4º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2011
Ing. Informática - 5º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2012
Ing. Informática - 6º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2012
Ing. Informática – 7º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2013
Ing. Informática – 8º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2014
Ing. Informática – 9º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2015
Ing. Informática – 10º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2016

Tabla 7 Antecedentes Académicos

1.4.1.3 Participación en proyectos de Investigación

Título proyecto	Institución	Cargo	Año

Tabla 8 Participación en Proyectos de Investigación

1.4.1.4 Publicaciones realizadas (libros, revistas, compendios y otros)

Autor	Tipo de publicación, Año, título, volumen, páginas, editorial

Tabla 9 Publicaciones Realizadas

1.4.1.5 Antecedentes en Ayudantías

Carrera	Semestre	Año
Ing. Informática	Laboratorio	2016

Tabla 10 Antecedentes en Ayudantías

1.4.1.6 Cursos Realizados

Curso	Institución	Duración en Hrs	Año
MANT. Y REPARACION DE COMPUTADORAS Y REDES	Instituto Domingo Sabio.	747	2010
Desarrollo de Aplicaciones WEB con Spring MVC	Universidad Autónoma Juan	20	2013

	Misael Saracho.		
Curso Seminario JavaScript	SOLOLEARN Discover your Potential	10	2017
Curso Fundamentos HTML	SOLOLEARN Discover your Potential	10	2017
Curso Fundamentos de SQL	SOLOLEARN Discover your Potential	10	2018

Tabla 11 Cursos Realizados

II. Capítulo II

2.1 Componente 1: Sistema de control de documentación de la Brigada Parlamentaria de Tarija

2.1.1 Marco Teórico

2.1.1.1 Introducción

Una organización es un sistema compuesto por tres elementos: personas, materiales e información. Los sistemas de información, por su parte, surgen como sistemas complejos y abiertos que interactúan con otros sistemas y subsistemas como parte de su actuación. Por los años 90, una de las concepciones más defendidas por la gestión de la información fue que las organizaciones son sistemas de información.

El uso de ciertos conceptos tomados de la teoría de sistemas y del campo de la informática llevó a un alto grado de desarrollo entre los sistemas de información. Aunque existen diversas definiciones, hechas desde diferentes enfoques, sobre los sistemas de información, en su gran mayoría tienen puntos en común. El análisis realizado sobre las definiciones más frecuentes efectuadas en la década de los años 90 revela que constituyen un conjunto integrado de procesos, elementos o componentes que –según las estrategias y necesidades de una organización– recopilan, elaboran y distribuyen la información necesaria.

Un sistema moderno de gestión de información exige la aplicación de nuevas tecnologías de información; sin embargo, la tecnología por sí sola no es suficiente para lograr una buena gestión de información. Son diversos los procesos que conforman los sistemas de gestión de información; ellos generan las entradas y salidas del sistema o de otros procesos relacionados; también pueden identificarse, controlarse, corregirse o actualizarse en la medida en que se producen las transformaciones del entorno y evoluciona la organización, como vía incuestionable para garantizar su calidad, eficiencia y mejora continua.

A modo de resumen de este antecedente de marco teórico, puede decirse que los sistemas de gestión de información, en su definición más general, se refieren al

conjunto de todos los componentes necesarios que se interrelacionan, con el objetivo de tramitar y facilitar la información sobre el tema de interés para su consumo en cualquier medio, momento y lugar.

2.1.1.2 Estudio de la variable Independiente: Gestión de la Información

2.1.1.2.1 Brigada Parlamentaria de “TARIJA”

2.1.1.2.1.1 Misión

Ejercer la soberanía y representación del pueblo, legislando y fiscalizando los poderes públicos, con calidad y transparencia en la gestión legislativa

2.1.1.2.1.2 Vision

Una Brigada Parlamentaria fortalecida como órgano intercultural que consolida el desarrollo pleno de la democracia y sus valores

2.1.1.2.1.3 Objetivo General

Objetivo estratégico de la función legislativa: Realizar la labor legislativa a través del fortalecimiento de las funciones de legislar, fiscalizar y gestionar para generar un trabajo óptimo y con resultados esperados en el plazo de cinco años.

Objetivo estratégico de la calidad legislativa: Trabajar en la satisfacción de expectativas y las necesidades latentes del pueblo mediante las herramientas de calidad de deliberación, técnicas legislativas, lista de verificación y tecnología informática, en un periodo legislativo, contribuyendo de esa manera a la calidad legislativa.

Objetivo estratégico de transparencia: Exponer y someter al análisis del pueblo la información relativa a la gestión legislativa con transparencia para una mejor imagen y percepción de la población en un periodo legislativo.

Objetivo estratégico de Interculturalidad: Desarrollar un trabajo Intercultural generando respeto, reciprocidad, cooperación y solidaridad para la

construcción de una sociedad democrática, mejorando la generación del valor organizativo en el plazo de cinco años.

Objetivo estratégico de Gestión de Calidad Administrativa: Promover la gestión administrativa mediante una gestión de calidad para la satisfacción de los usuarios y la ciudadanía así como de la misión de la C.D., promoviendo el hacer bien lo que tenemos que hacer, fortaleciendo la institucionalidad, en un plazo de cinco años.

2.1.1.2.1.4 Matriz Foda

El FODA se representa a través de una matriz de doble entrada, llamada matriz FODA, en la que el nivel horizontal se analiza los factores positivos y los negativos.

En la lectura vertical se analizan los factores internos y por tanto controlables del programa o proyecto y los factores externos, considerados no controlables.

Las Fortalezas son todos aquellos elementos internos y positivos que diferencian al programa o proyecto de otros de igual clase.

Las Oportunidades son aquellas situaciones externas, positivas, que se generan en el entorno y que una vez identificadas pueden ser aprovechadas

Las Debilidades son problemas internos, que una vez identificados y desarrollando una adecuada estrategia pueden y deben eliminarse.

Las Amenazas son situaciones negativas, externas al programa o proyecto, que pueden atentar contra éste, por lo que llegado al caso, puede ser necesario diseñar una estrategia adecuada para poder sortearla.



Figura 8 Matriz FODA

2.1.1.2.1.5 Diagnóstico Estratégico

El análisis realizado permite detectar que la Brigada Parlamentaria de Tarija depende de varios factores muy sensibles para poder realizar un trabajo óptimo.

El análisis es útil para que la Institución conozca qué características deba tomar en cuenta para afrontar estratégicamente y cuáles son sus principales fortalezas que motiven a la institución seguir adelante y mejorar, optimizar y estandarizar los servicios que ofrece.

Es importante recordar que los parlamentarios son representantes directos de la población, por lo tanto se espera que el trabajo de los legisladores sea en beneficio de la población a la cual representan.

Es imprescindible que la institución cuente con talento humano calificado para el apoyo a los legisladores, logrando optimizar el trabajo que desempeñan.

Debido a la coyuntura política actual, el trabajo de la institución se ve truncado por lo tanto se debe mejorar este aspecto, para mejorar la imagen institucional hacia la ciudadanía.

2.1.1.3 Metodología del Desarrollo

2.1.1.3.1 Metodología RUP (Rational Unified Process)¹

a) Definición.- El Proceso Unificado Rational (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

También se conoce por este nombre al software desarrollado por Rational, hoy propiedad de IBM, el cual incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades.

b) Características esenciales.- El RUP presenta tres características muy marcadas e importantes que son las siguientes:

- **Está dirigido por los Casos de Uso:** Un sistema de software se crea para servir a sus usuarios. Por lo tanto, para construir un sistema exitoso se debe conocer qué es lo que quieren y necesitan los usuarios prospectos. El término usuario se refiere no solamente a los usuarios humanos, sino a otros sistemas. En este contexto, el término usuario representa algo o alguien que interactúa con el sistema por desarrollar.

Un caso de uso es una pieza en la funcionalidad del sistema que le da al usuario un resultado de valor. Los casos de uso capturan los requerimientos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso el cual describe la funcionalidad completa del sistema.

Aún y cuando los casos de uso dirigen el proceso, no son elegidos de manera

¹ Ingeniería de Software – Universidad Autónoma Del Perú - Material de Lectura Sesión N° 2

aislada. Son desarrollados a la par con la arquitectura del sistema, esto es, los casos de uso dirigen la arquitectura del sistema y la arquitectura del sistema influencia la elección de los casos de uso. Por lo tanto, la arquitectura del sistema y los casos de uso maduran conforme avanza el ciclo de vida.

- **Está Centrado en su Arquitectura:** El papel del arquitecto de sistemas es similar en naturaleza al papel que el arquitecto desempeña en la construcción de edificios. El edificio se mira desde diferentes puntos de vista: estructura, servicios, plomería, electricidad, etc. Esto le permite al constructor ver una radiografía completa antes de empezar a construir. Similarmente, la arquitectura en un sistema de software es descrita como diferentes vistas del sistema que está siendo construido.

El concepto de arquitectura de software involucra los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, tal y como las interpretan los usuarios y otros stakeholders, y tal y como están reflejadas en los casos de uso. Sin embargo, también está influenciada por muchos otros factores, tales como la plataforma de software en la que se ejecutará, la disponibilidad de componentes reutilizables, consideraciones de instalación, sistemas legados, requerimientos no funcionales (ej. desempeño, confiabilidad).

La arquitectura es la vista del diseño completo con las características más importantes hechas más visibles y dejando los detalles de lado. Ya que lo importante depende en parte del criterio, el cual a su vez viene con la experiencia, el valor de la arquitectura depende del personal asignado a esta tarea. Sin embargo, el proceso ayuda al arquitecto a enfocarse en las metas correctas, tales como claridad (understandability), flexibilidad en los cambios futuros (resilience) y reusó.

¿Cómo se relacionan los casos de uso con la arquitectura? Cada producto tiene función y forma. Uno sólo de los dos no es suficiente. Estas dos fuerzas deben estar balanceadas para obtener un producto exitoso. En este caso función corresponde a los casos de uso y forma a la arquitectura. Existe la

necesidad de intercalar entre casos de uso y arquitectura. Es un problema del “huevo y la gallina”. Por una parte, los casos de uso deben, cuando son realizados, acomodarse en la arquitectura. Por otra parte, la arquitectura debe proveer espacio para la realización de todos los casos de uso, hoy y en el futuro. En la realidad, ambos arquitectura y casos de uso deben evolucionar en paralelo.

- **Es Iterativo e Incremental:** Desarrollar un producto de software comercial es una tarea enorme que puede continuar por varios meses o años. Es práctico dividir el trabajo en pequeños pedazos o mini-proyectos. Cada mini-proyecto es una iteración que finaliza en un incremento. Las iteraciones se refieren a pasos en el flujo de trabajo, los incrementos se refieren a crecimiento en el producto. Para ser más efectivo, las iteraciones deben estar controladas, esto es, deben ser seleccionadas y llevadas a cabo de una manera planeada.

Los desarrolladores basan su selección de qué van a implementar en una iteración en dos factores. Primero, la iteración trata con un grupo de casos de uso que en conjunto extienden la usabilidad del producto. Segundo, la iteración trata con los riesgos más importantes. Las iteraciones sucesivas construyen los artefactos del desarrollo a partir del estado en el que fueron dejados en la iteración anterior.

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean el diseño usando la arquitectura como guía, implementan el diseño en componentes y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple sus metas – y usualmente lo hace – el desarrollo continúa con la siguiente iteración. Cuando la iteración no cumple con sus metas, los desarrolladores deben revisar sus decisiones previas y probar un nuevo enfoque.

Otras características del RUP son las siguientes:

- Administración eficiente y eficaz de requisitos
- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué,

cuándo y cómo)

- Pretende implementar las mejores prácticas en Ingeniería de Software
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software
- Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

c) Ciclo de Vida.- El ciclo de vida del RUP, como se conoce al trazado de las actividades de desarrollo en el tiempo, está dividido en 4 fases que son: Inicio, Elaboración, Construcción y Transición, que corresponden a los 4 hitos principales de RUP: Proyecto, Arquitectura, Versión Beta (β) y Entregable (release).

En términos de habilidades y conocimiento, el RUP está dividido en principios clave. Cada uno de ellos corresponde a distintos aspectos del desarrollo de software que generalmente requieren habilidades específicas; esto se refleja en los roles y las actividades definidas para cada principio.

Cada fase cambia el foco del equipo de trabajo para alcanzar cada uno de los hitos y es llevada a cabo en forma iterativa. Esto quiere decir que la fase se fragmenta en pequeños proyectos que recorren todas las disciplinas y producen un ejecutable —en el sentido de software. Dicho producto es la forma más efectiva de verificar el progreso del proyecto y de reducir los riesgos inherentes.

El proceso se puede describir en dos dimensiones:

- El eje horizontal representa tiempo y demuestra el aspecto dinámico del proceso, se expresa en términos de ciclos, de fases, de iteraciones, y de hitos o milestones.

- El eje vertical representa el aspecto estático del proceso: cómo se describe en términos de actividades, de dispositivos, de trabajadores y de workflows. .

El ciclo de vida del software está particionado en ciclos, cada ciclo trabaja en una nueva generación del producto. El RUP divide un ciclo de desarrollo en cuatro fases consecutivas.

- Fase de inicio
- Fase de elaboración
- Fase de construcción
- Fase de transición

Cada fase constituye un eslabón bien definido, un punto en el tiempo en el cual ciertas decisiones críticas deben tomarse, y por lo tanto afinar metas debe haber sido alcanzadas.

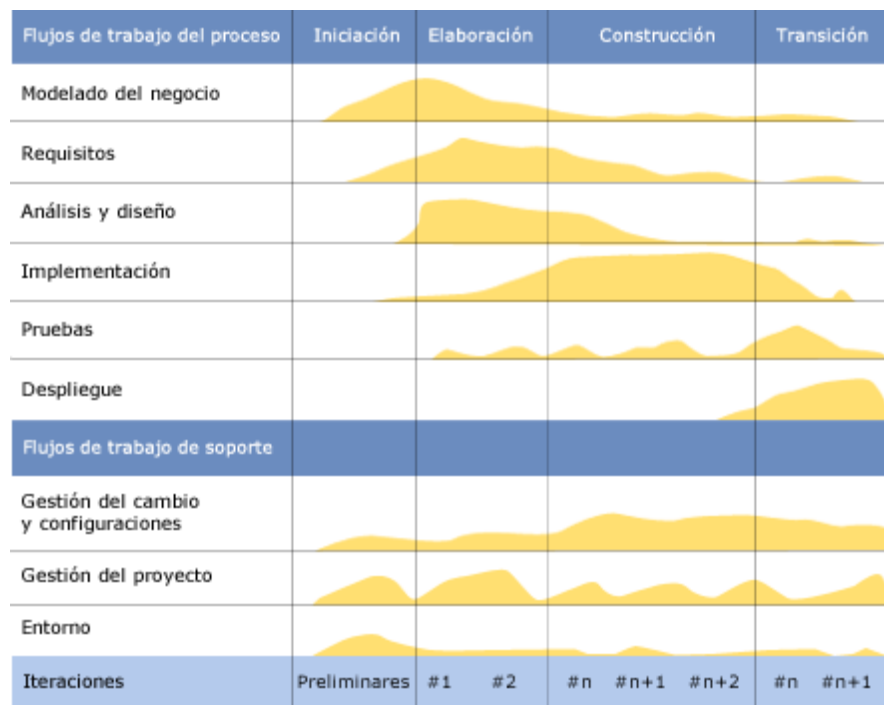


Figura 9 Fases en el Ciclo de Vida del RUP

- **Fase de Inicio:** Durante la fase del inicio, se establece el caso de negocio para

el sistema y delimita el alcance del proyecto. Para lograr esto debe identificar todas las entidades externas con las cuales el sistema interactúe (los actores) y definir la naturaleza de esta interacción a un nivel alto. Esto implica identificar todos los casos de uso y describir sólo los más significativos. El caso de negocio incluye criterios de éxito, la evaluación de riesgos, y la estimación de los recursos necesarios, y un plan de la fase que muestre las fechas previstas e hitos importantes.

El resultado de la fase del inicio es:

- Un documento de la visión: una visión general de los requerimientos básicos del proyecto, de las características dominantes, y de las restricciones principales.
 - Un modelo inicial de casos de uso (10%-20% completo).
 - Un glosario inicial del proyecto (opcionalmente puede ser expresado como modelo de dominio).
 - Un caso inicial de negocio, que incluye contexto del negocio, los criterios del éxito (proyección del rédito, reconocimiento del mercado, etcétera), y pronóstico financiero.
 - Una estimación de riesgo inicial.
 - Un plan de proyecto, demostrando fases e iteraciones.
 - Un modelo de negocio, en caso de necesidad.
 - Uno o más prototipos.
- **Fase de elaboración:** El propósito de la fase de elaboración es analizar el dominio del problema, establecer una fundación arquitectónica sana, desarrollar el plan del proyecto, y eliminar los elementos del riesgo más alto del proyecto. Para lograr estos objetivos, usted debe tener una visión completa del sistema. Las decisiones arquitectónicas tienen que tomarse con una comprensión cabal del sistema: su alcance, funcionalidad importante y requerimientos no funcionales tales como requerimientos de performance. Es fácil argumentar que la fase de elaboración es la más crítica de las cuatro fases. En el final de esta fase, la “ingeniería dura” se considera completa y el

proyecto experimenta su día más importante: la decisión sobre si o no confiar en las fases de la construcción y de la transición.

Para la mayoría de los proyectos, esto también corresponde a la transición de una fase de operatoria móvil, ligera y ágil, poco arriesgada, a una de alto costo, riesgo elevado con una inercia substancial. Mientras que el proceso siempre debe acomodarse a los cambios, las actividades de la fase de elaboración aseguran que la arquitectura, los requerimientos y los planes sean bastante estables, y que los riesgos se atenúan lo suficiente, así usted puede determinar el costo y fecha de terminación del desarrollo en forma bastante certera.

Durante fase de elaboración, se construye un prototipo ejecutable de la arquitectura en unas o más iteraciones, dependiendo del alcance, del tamaño, del riesgo, y de la novedad del proyecto. Este prototipo debe tratar por lo menos los casos de uso más críticos identificados en la fase del inicio, que exponen típicamente los mayores riesgos técnicos del proyecto. Mientras que un prototipo evolutivo de un componente de calidad es siempre la meta, no excluye el desarrollo de unos o más prototipos exploratorios, desechables, para atenuar riesgos específicos.

El resultado de la fase de elaboración es:

- Un modelo de caso de uso (por lo menos 80% completo) - todos los casos de uso y actores deben haber sido identificados-, y se han desarrollado la mayoría de las descripciones de casos de uso.
- Requerimientos suplementarios que capturan los requerimientos no funcionales o cualquier requerimiento que no se asocie a un caso de uso específico.
- Una descripción de la arquitectura del software.
- Un prototipo arquitectónico ejecutable.
- Una lista revisada del riesgo y un caso de negocio revisado.
- Un plan de desarrollo para el proyecto total, incluyendo el plan de grano grueso del proyecto, demostrando iteraciones “y los criterios de

la evaluación para cada iteración.

- Un caso actualizado del desarrollo que especifica el proceso que se utilizará.
- Un manual preliminar del usuario (opcional).

- **Fase de Construcción:** Durante la fase de la construcción, todos los componentes y características restantes se desarrollan, se integran en el producto, y se prueban a fondo. La fase de la construcción es, en cierto sentido, un proceso de fabricación donde el énfasis se pone en manejar los recursos y controlar las operaciones para optimizar costos, tiempos y calidad. Una arquitectura robusta y un plan comprensible están íntimamente relacionados. Es decir, una de las cualidades críticas de la arquitectura es su facilidad de la construcción. Ésta es una razón por la que durante la fase de elaboración se pone el énfasis en el desarrollo equilibrado de la arquitectura y del plan.

El resultado de esta fase es un producto listo para poner en las manos de los usuarios finales. Como mínimo, consta de:

- El producto de software integrado en las plataformas adecuadas.
- Los manuales del usuario.
- Una descripción de la versión o release actual.

- **Fase de Transición:** El propósito de la fase de la transición es justamente la transición del producto de software al ambiente de producción. Una vez que el producto se haya entregado al usuario final, surgen algunos temas que llevan al desarrollo de nuevas versiones, a corregir errores, o a terminar algunas características que habían sido pospuestas.

Se ingresa a esta fase cuando el producto está lo suficientemente maduro para comenzar a pasar a producción. Esto requiere que un cierto subconjunto del sistema se encuentre en un nivel aceptable de la calidad y que la documentación del usuario está disponible de modo que la transición proporcione resultados positivos para todas las partes. Esto incluye:

- La “prueba beta” para validar el nuevo sistema contra las expectativas

del usuario

- Operación en paralelo con un sistema anterior que el nuevo sistema esté sustituyendo
- La conversión de las bases de datos operacionales
- Entrenamientos y capacitación de los usuarios y la gente de mantenimiento
- Lanzar el producto a los equipos de marketing, distribución y ventas

La fase de transición se centra en las actividades requeridas para poner el software en manos de los usuarios. Típicamente, esta fase incluye varias iteraciones, incluyendo lanzamientos beta, lanzamientos de disponibilidad general, así como la reparación de errores y el lanzamiento de versiones mejoradas.

Un esfuerzo considerable se realiza en la documentación orientada al usuario final, en entrenar a los mismos, en brindar apoyo en las primeras etapas del uso, y en reaccionar al feedback que generen los mismos usuarios. En este punto del ciclo de vida, sin embargo, el feedback del usuario se debe centrar sobre todo en el ajuste fino del producto, la configuración, instalación, y a las cuestiones de usabilidad.

Esta fase puede variar -según el proyecto- de ser muy simple a muy compleja. Por ejemplo una nueva versión de un procesador de texto puede ser muy simple, mientras que substituir el sistema de control de tráfico aéreo de un país sería muy complejo.

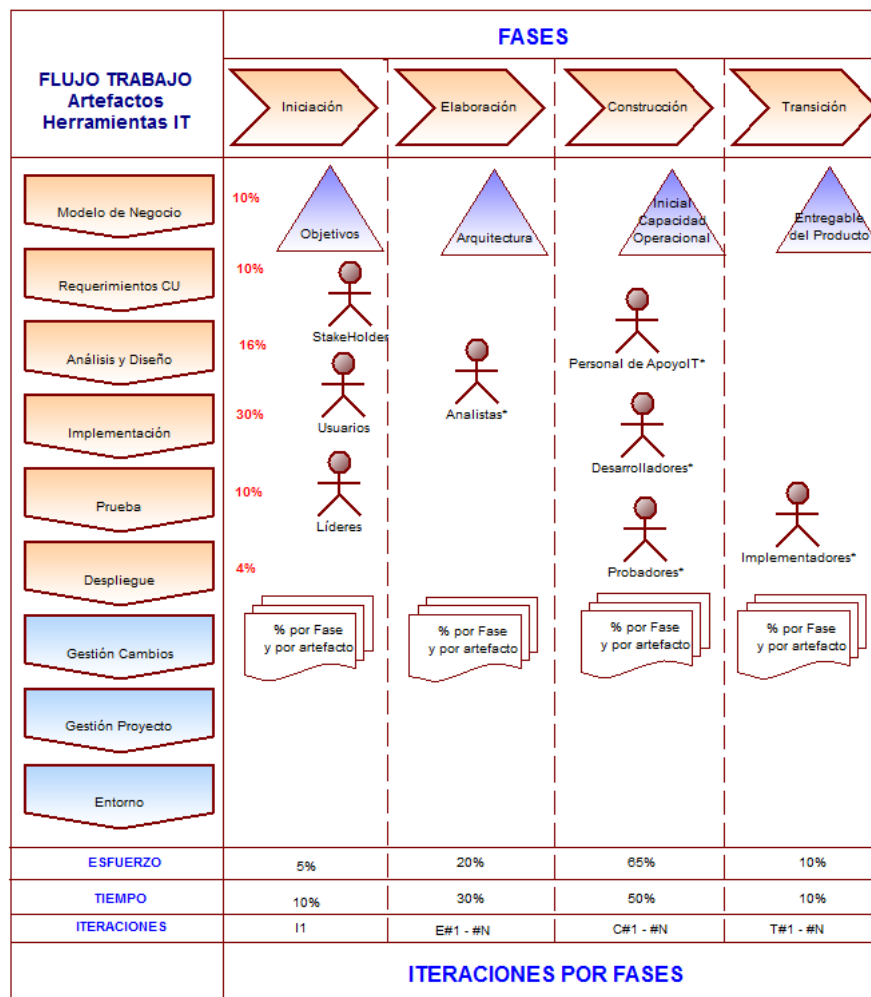


Figura 10 Flujo de Trabajo del RUP

2.1.1.3.2 UML (Lenguaje Unificado de Modelado)²

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).

Es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de

² Ingeniería de Software – Universidad Autónoma Del Perú - Material de Lectura Sesión N° 2

facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

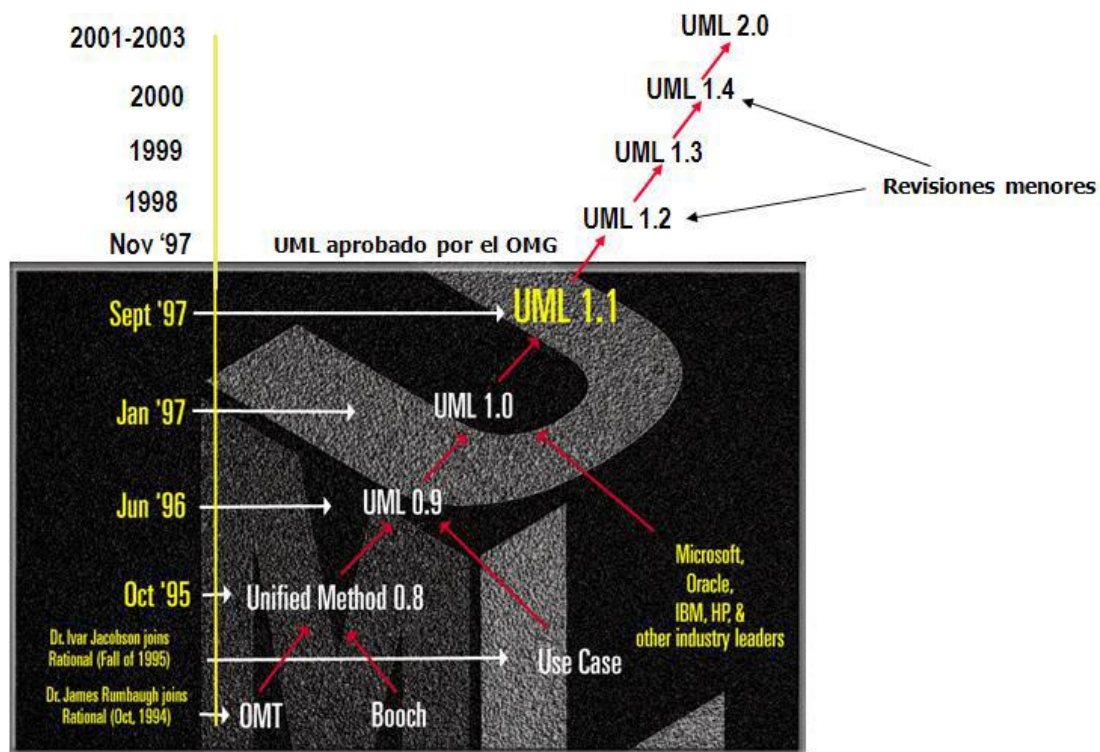


Figura 11 Fases históricas del UML

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. El UML no es una metodología de desarrollo de sistemas de información.

Uno de los objetivos principales de la creación de UML era posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Para ello era necesario definir una notación y semántica común.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representa

- a) **Vistas en UML:** Un enfoque en la presentación de un sistema en UML es conocida como 4+1 vistas. Esta forma de documentar nuestros modelos y divide lo que sabemos de él en cinco áreas:



Figura 12 Vistas en UML

- **Vista de Casos de Uso.** Que contiene requisitos desarrollados en las restantes vistas.
- **Vista Lógica.** Muestra la estructura estática del sistema.
- **Vista Física.** Muestra el despliegue de la aplicación en la red de computadoras.

- **Vista de Procesos.** Muestra los hilos y procesos de ejecución así como la comunicación entre estos.
- **Vista de Desarrollo.** Muestra la estructura en modelos del código del sistema.

Estas vistas se presenta tradicionalmente en una figura de cuatro cajas con un ovalo central que representa al modelo de casos de uso. Dicho gráfico no es UML pero al ser tan conocido no se puede dejar de incluir en este marco teórico.

- b) **Diagramas en UML:** A continuación se muestra los diagramas que existen en para el modelamiento de sistemas mediante UML.

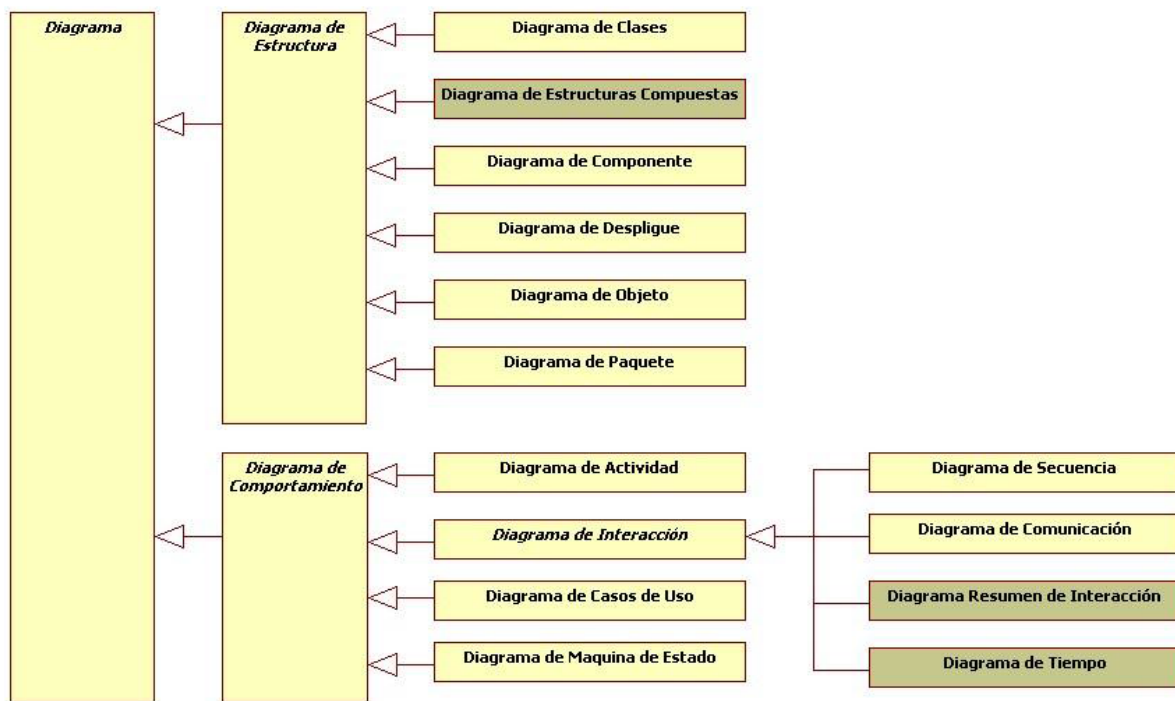


Figura 13 Diagramas en UML

- c) **Que no es UML:**

- UML no es un método de desarrollo. No te va a decir cómo pasar del análisis al diseño y de este al código. No son una serie de pasos que te llevan a producir código a partir de unas especificaciones.

- UML al no ser un método de desarrollo es independiente del ciclo de desarrollo que vayas a seguir, puede encajar en un tradicional ciclo en cascada, o en un evolutivo ciclo en espiral o incluso en los métodos ágiles de desarrollo.

d) Relación entre Modelos y Diagramas en UML: Un modelo captura una vista de un sistema del mundo real. Es una abstracción de dicho sistema, considerando un cierto propósito. Así, el modelo describe completamente aquellos aspectos del sistema que son relevantes al propósito del modelo, y a un apropiado nivel de detalle.

Por su parte un diagrama, es una representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo con vértices conectados por arcos.

Un proceso de desarrollo de software debe ofrecer un conjunto de modelos que permitan expresar el producto desde cada una de las perspectivas de interés. El código fuente del sistema es el modelo más detallado del sistema (y además es ejecutable).

Sin embargo, se requieren otros modelos. Cada modelo es completo desde su punto de vista del sistema, sin embargo, existen relaciones de trazabilidad entre los diferentes modelos. Por otro lado, los diagramas expresan gráficamente partes de un modelo.

2.1.1.3.2.1 Tipos de Diagramas Utilizados

2.1.1.3.2.1.1 Diagrama de Caso de Uso del Sistema

El modelo de caso de uso es un modelo de las funciones deseadas del sistema y su entorno, y hace las funciones de un contrato entre el cliente y los desarrolladores.

Es utilizado como una introducción esencial para las actividades de análisis, diseño, y prueba del sistema.

Un modelo de caso de uso muestra a los actores, casos de uso, paquetes de caso de uso, y sus relaciones.

El modelo de caso de uso es un modelo que describe los requisitos de un sistema en términos de casos de uso. Los casos de uso y los actores definen el alcance del sistema que usted construye. Los casos de uso incluyen cualquier cosa que está dentro del sistema; los actores incluyen cualquier cosa que es externa para el sistema.

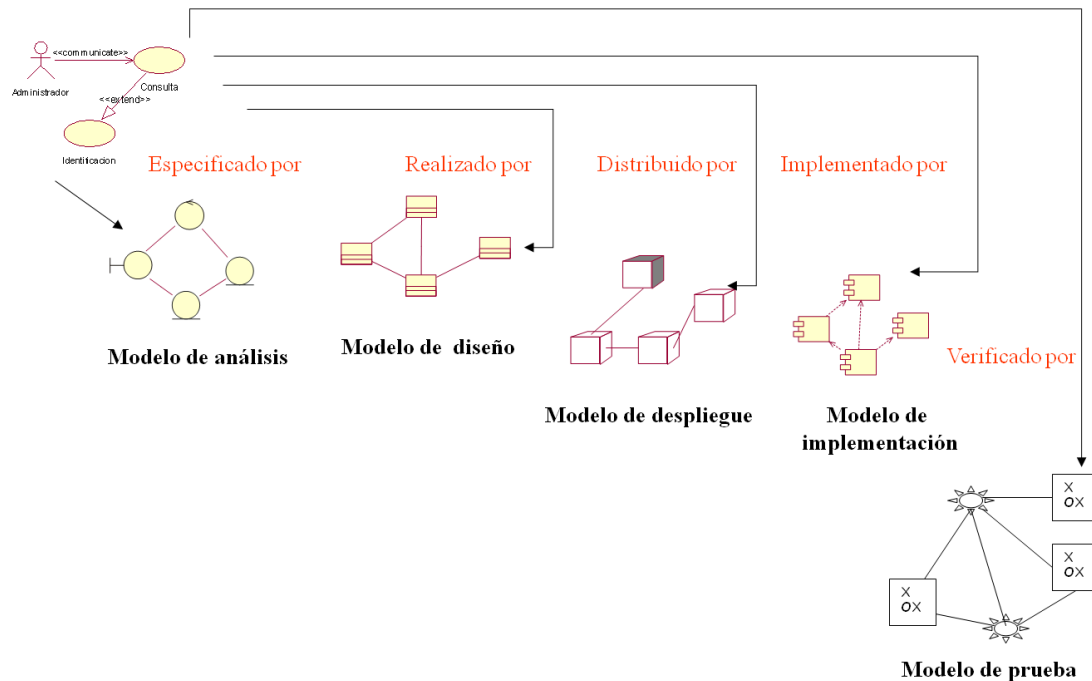


Figura 14 Dependencia de los demás modelos de los casos de uso

a) **Objetivos:** Entre los principales objetivos de los diagramas de casos de uso se pueden mencionar los siguientes:

- Definir el comportamiento del sistema.
- Definir los casos de uso y sus respectivos actores.
- Entender como documentar los casos de uso.
- Utilizar un diagrama de casos de uso para mostrar los actores, casos de uso y sus interacciones.
- Definir escenarios para los casos de uso.

b) **Artefactos:**

- **ACTOR:** Un actor es alguien o algo que interactúa con el sistema que será construido. Los casos de uso describen algo que está dentro del

alcance del sistema. Los actores son algo que es externo al alcance del sistema.

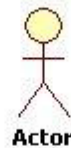


Figura 15 Diagrama de Caso de Uso: Artefacto - ACTOR

- **CASO DE USO (Use Case):** Es una secuencia de acciones realizadas por el sistema que producen un resultado observable y valioso para alguien en particular. Un caso de uso es justamente una forma de representar como alguien o algo usa nuestro sistema.



Figura 16 Diagrama de Caso de Uso: Artefacto - USE CASE

- c) **Documentación de los Casos de Uso:** Una diagrama de casos de uso describe lo que hace el sistema, pero no describe como lo hace, al construir los diagramas de casos de uso se debe tener bien en claro esta separación.

El comportamiento de un caso de uso, puede ser descrito de muchas maneras dependiendo de la conveniencia, a veces podemos usar pseudocódigo; sin embargo, comúnmente un caso de uso se documenta de manera informal mediante un lista de pasos que sigue el Actor durante su interacción con el sistema. A esta lista de pasos se le denomina FLUJO DE EVENTOS.

El Flujo de Eventos de un caso de uso contiene la información más importante derivada desde la actividad de modelado de caso de uso. Debería describir el flujo de eventos del caso de uso lo suficientemente claro para que una para persona ajena o externa lo entienda fácilmente. Recordar que el flujo de eventos debería presentar lo que el sistema hace, no cómo es que el sistema es diseñado para desempeñar el comportamiento requerido.

Un caso de uso se documenta generalmente con texto informal, por lo tanto si tenemos que especificar formalmente un algoritmo, los casos de uso no son los más adecuados, en su lugar debemos usar los diagramas de actividad, secuencia o colaboración.

Nombre			
Descripción			
Actores			
PreCondiciones		PostCondiciones	
StakeHolders	Interesado	Interés	
Evento de Inicio			
Flujo Básico o Normal de Eventos			
Nº	Actor	Sistema	Excepciones
Flujo Alternativo de Eventos (Excepciones)			
Nº Excepción	Nombre de Excepción	Acción	
Prioridad			
Frecuencia			
Fecha			
Autor			

Figura 17 Documentación de los Casos de Uso (Ejemplo)

2.1.1.3.2.1.2 Diagrama de Secuencia

Es un tipo de diagrama de interacción que muestra justamente la interacción de un conjunto de objetos, poniendo énfasis en el orden cronológico del envío de mensajes entre objetos.

Mediante este tipo de diagrama podemos dar detalle a los casos de uso, aclarándolos al nivel de mensajes de los objetos existentes, como también muestra el uso de mensajes de las clases diseñadas en el contexto de una operación.

La creación de diagramas de secuencia forma parte de la investigación para conocer el sistema, por lo que es parte del análisis del mismo. La creación de diagramas de secuencia depende de la formulación de casos de uso, porque durante la operación del sistema, los actores generan eventos solicitando alguna operación.

Podemos leer este diagrama mirando los objetos y mensajes. Los objetos que participan del flujo son mostrados en rectángulos a través de la parte superior del diagrama. Cada objeto tiene una línea de vida, dibujado como una línea vertical delineada por debajo del objeto. La línea de vida comienza cuando el objeto es instanciado y acaba cuando el objeto se destruye.

Un mensaje se traza entre las líneas de vida de dos objetos para demostrar que los objetos se comunican. Cada mensaje representa un objeto haciendo una llamada de función de otro. Más tarde durante el proceso, cuando definamos las operaciones para las clases, cada mensaje se convertirá en una operación. Los mensajes también pueden ser reflexivos, demostrando que un objeto puede llamar a una de sus propias operaciones.

a) Artefactos:

- **OBJETOS:** Nosotros vemos objetos a todo nuestro alrededor. La silla en la que usted está sentado, el libro que usted lee, y la bombilla que le ayuda a ver son todos ejemplos de objetos en la vida real. Un objeto en el mundo del software es casi lo mismo.

Un objeto es algo que encapsula información y comportamiento. Es un término que representa alguna cosa concreta, del mundo real.

Los objetos son adicionados a los diagramas de Interacción. Podemos colocar un actor o alguna otra clase (entidad, control o interfaz) encima de un diagrama de Interacción, una instancia del objeto de esa

clase automáticamente será creada. Removiendo un objeto de un diagrama en Rose no suprimirá la clase del modelo.

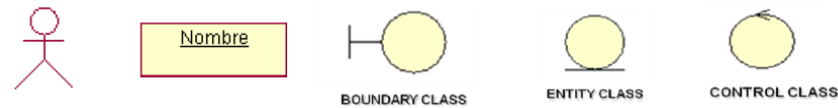


Figura 18 Diagrama de Secuencia: Artefacto – OBJETO

- **LINEA DE VIDA:** Indica la vida de un objeto durante la interacción y se representa como una línea vertical punteada debajo del rectángulo del objeto. Debe tener en cuenta que algunos objetos pueden ser creado y destruidos por la interacción.

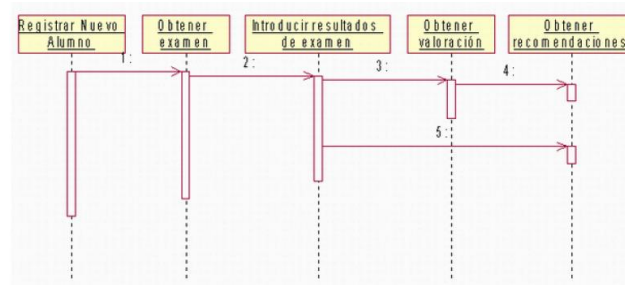


Figura 19 Diagrama de Secuencia: Artefacto - LINEA DE VIDA

- **MENSAJE:** El envío de mensajes entre objetos se representa mediante una línea sólida dirigida con cabeza de flecha abierta, desde el objeto emisor del mensaje hacia el objeto receptor. Este último ejecutará la acción indicada por el emisor.

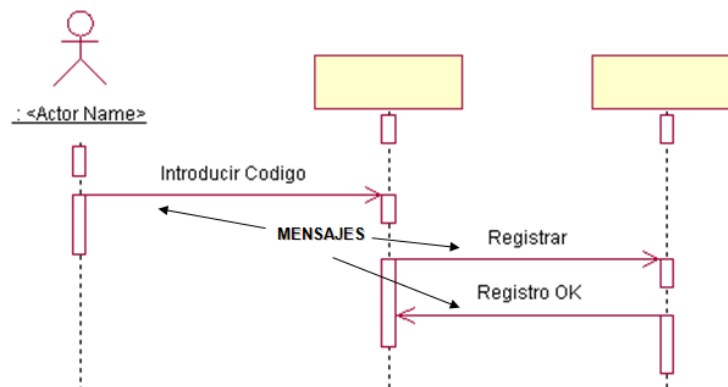


Figura 20 Diagrama de Secuencia: Artefacto - MENSAJE

2.1.1.3.2.1.3 Diagrama de Clases

Un diagrama de clases presenta las clases y objetos del sistema con sus relaciones estructurales y de herencia.

El trabajo realizado en los Diagramas de Casos de Uso y Diagramas de Secuencia aporta información para establecer las clases, objetos, atributos y métodos.

El diagrama de clases contiene varios modelos los cuales son:

- a) **Modelo Lógico:** El Modelo Lógico termina de refinar el Modelo Conceptual, aquí se reducen y/o aumentan clases y quedando aquellas que van a ser diseñadas como Tablas de la de la Base de Datos (Clases Relevantes) que almacenarán la información del Sistema.

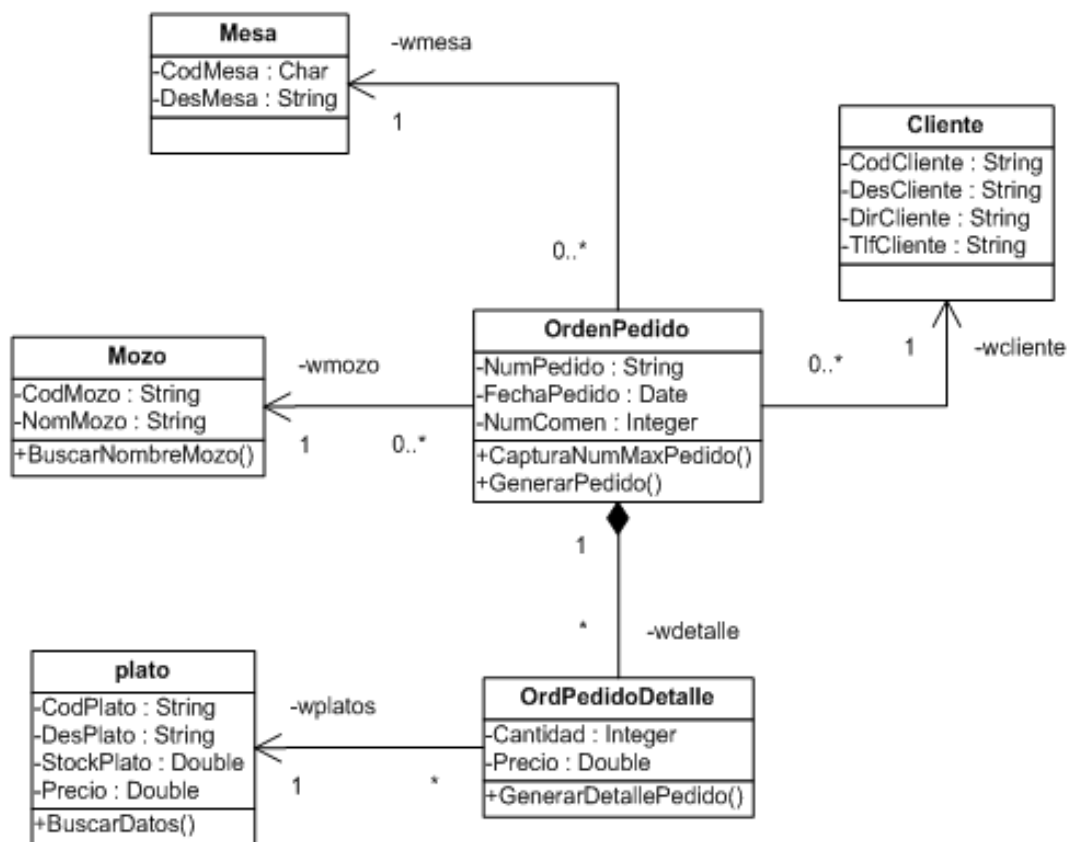


Figura 21 Diagrama de Clases: Modelo Lógico

b) **Modelo de Datos:** Representación de la vista física de los datos. Es dependiente del tipo de Base de Datos. Está compuesto por:

- Tablas.
- Columnas o campos.
- Llaves primarias y foráneas.
- Restricciones.
- Índices.
- Relaciones.
- Diagrama del Modelo de Datos.

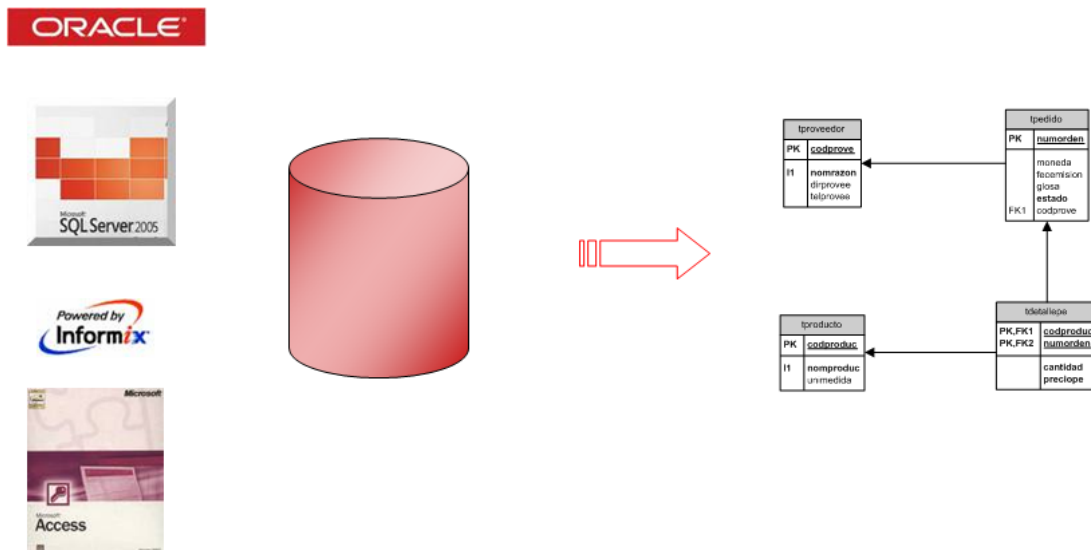


Figura 22 Diagrama de Clases: Modelo de Datos

b) **Modelo Físico:** Se definen las clases persistentes (tablas). Se crearan:

- Llave Primaria (PK)
- Llave Foránea (FK)
- Índices

c) **Modelo de Almacenamiento:** Representación de la vista física de la Arquitectura de Almacenamiento de los datos. Elementos de Almacenamiento de Base de Datos. Es dependiente de la Base de Datos. Está compuesto por:

- Componentes de Base de Datos.
- TABLESPACES.

- Script de la Base de Datos física.
- Diagrama de Componentes.

2.1.1.3.2.1.4 Diagrama de Actividad

En el mundo real todo cambia con el transcurso del tiempo; el clima, la moda, la geografía, los seres vivos, en fin, en la vida cotidiana tenemos que ver con cambios a cada instante, surgiendo pues la necesidad de representar estos cambios.

Para esto se propuso el concepto en UML de máquinas de estado, que añadido a otros conceptos como la máquina de Moor y máquinas de Mealy entre otros, forman el moderno Diagrama de Estados del UML.

Los diagramas de estado capturan los cambios de estado que sufren los objetos en respuesta a eventos. Los diagramas de clases, solo muestran los aspectos estáticos pero no muestran como son afectados los objetos cuando ocurre algo.

Sin embargo, estos comportamientos tienen que implementarse mediante software y, representarlos en algún sitio, asegura que los desarrolladores no adivinen el comportamiento y produzcan software que satisfaga los requerimientos.

Un diagrama de estados muestra el ciclo de vida de un único objeto, desde el tiempo que es creado hasta que es destruido. **Estos diagramas son una buena manera de modelar el comportamiento dinámico de una clase. En un proyecto típico, no se crea un diagrama de estados para cada clase. De hecho, muchos proyectos no los usan en absoluto.**

a) Artefactos:

- **ESTADO:** Un estado queda definido por ciertas características que un objeto mantiene en un periodo de tiempo, en el cual el objeto puede recibir cierto tipo de estímulos como alguna condición, operación o evento. Los estados no son instantáneos, sino que tienen un tiempo de duración.

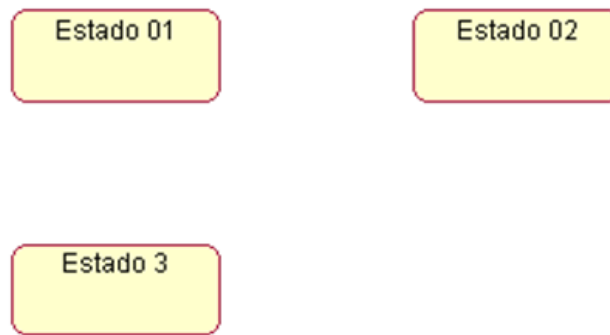


Figura 23 Diagrama de Actividad: Artefacto - ESTADO

- **ESTADO INICIAL Y FINAL:** x El estado inicial de un objeto se representa mediante un círculo relleno, mientras que su estado final se representa mediante un círculo relleno rodeado por una circunferencia. En la realidad el estado inicial y estado final son pseudoestados pues no poseen variables de estado y acciones, no constituyendo estados completos. Un objeto no puede permanecer en un estado inicial, pero si en un estado final.

Un estado final no puede tener ninguna transición y su propósito es invocar a una transición de finalización de algún estado.



Figura 24 Diagrama de Actividad : Artefacto - ESTADO INICIAL Y FINAL

- **EVENTO:** Un evento es una ocurrencia que puede causar la transición del objeto de un estado a otro. Es una ocurrencia que sucede en algún punto del tiempo. Los eventos pueden ser de muchos tipos, pero no necesariamente excluyentes:
 - Condición que toma el valor de verdadero o falso
 - Recepción de una señal de otro objeto en el modelo

- Recepción de un mensaje
- Paso de cierto período de tiempo después de entrar al estado, o llegar a cierta hora y fecha particular.

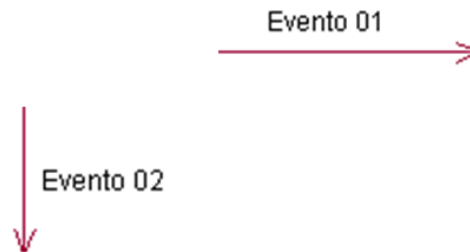


Figura 25 Diagrama de Actividad: Artefacto - EVENTO

- **TRANSICION:** Una transición es una acción que conceptualmente es instantánea, ósea atómica y no interrumpible. Una transición es una relación entre 2 estados que indica que un objeto en el primer estado puede o no entrar al segundo estado y ejecutar ciertas operaciones, cuando un evento ocurre y si ciertas condiciones son satisfechas.

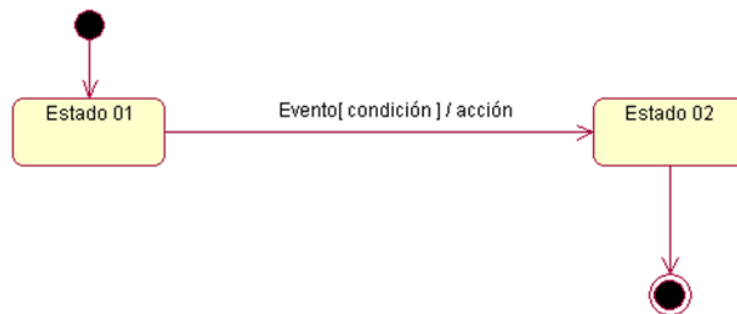


Figura 26 Diagrama de Actividad: Artefacto - TRANSICION

- **TRANSICION COMPLEJA:** Una transición compleja es una transición que relaciona dos o más estados en una transición de múltiples estados origen y/o múltiples estados de destino. Indica la subdivisión en hilos de control (threads) del objeto o una sincronización. Se representa con una corta barra vertical de sincronización o bifurcación, de la cual salen o entran varias líneas de transición de estado.

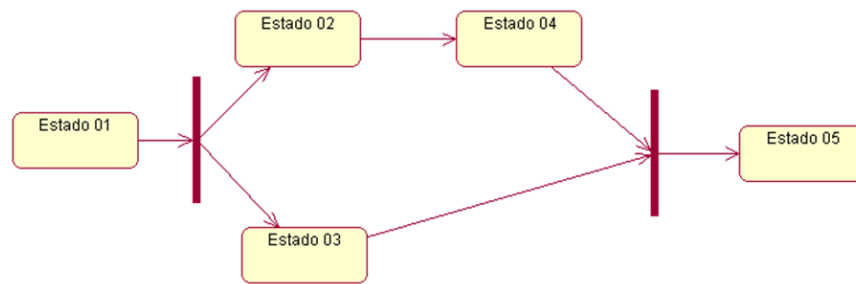
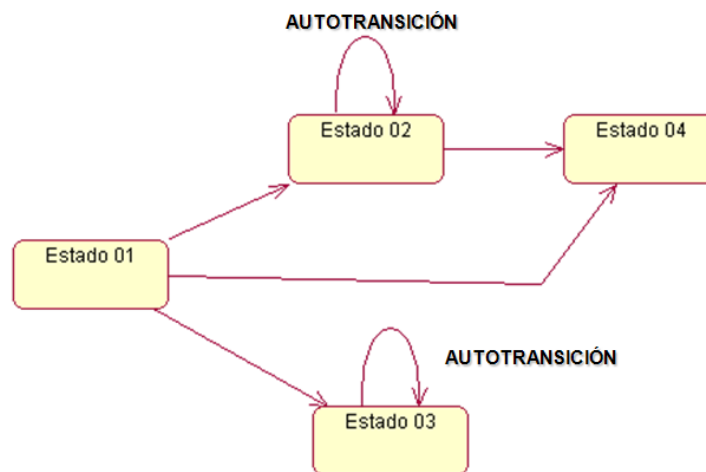


Figura 27 Diagrama de Actividad : Artefacto – TRANSICION COMPLEJA

- **AUTOTRANSICION:** Una auto transición es una transición en la que el estado de origen y el estado de destino son los mismos, pero conceptualmente se considera un cambio de estado. Cuando ocurre, se sale del estado origen y se vuelve a entrar en él, invocándose las acciones etiquetas con ENTRY y EXIT.



2.1.1.3.2.1.5 Diagrama de Componentes

En la vista Componente, se enfoca la atención en la organización física del sistema. Primero, decidiremos cómo las clases serán organizadas en librerías de código. Luego, le echaremos un vistazo a los diferentes archivos ejecutables, los archivos dinámicos de la librería del enlace (DLL), y otros archivos de ejecución en el sistema. No nos preocuparemos aún de donde los diferentes archivos serán colocados en la red. Consideraremos estos asuntos en la vista de Implementación.

Antes de generar código, traza un mapa de cada uno de sus archivos para el componente correcto (s). En C++, por ejemplo, cada clase es asociada para dos componentes; una representando el archivo .cpp para esa clase y uno representando el archivo .h. En Java, usted traza un mapa de cada clase para un componente solo, representando el archivo

.java para esa clase. Cuando genera código, Rose usará la información del componente para crear los archivos correctos de la librería de código.

Una vez que los componentes son creados, son añadidos al diagrama de Componentes y las relaciones son dibujadas entre ellos. El único tipo de relación entre componentes es una dependencia. Una dependencia sugiere que un componente debe ser compilado antes que el otro.

a) Artefactos:

- **COMPONENTES:** Un componente es un módulo físico de código. Los componentes pueden incluir tanto librerías de código fuente y archivos de ejecución. Por ejemplo, si usted usa C++, cada uno de su .cpp y sus archivos .h son un componente aparte. El archivo .exe que crea después de que el código sea compilado es también un componente. Se puede usar varios iconos diferentes para representar los tipos diferentes de componentes.

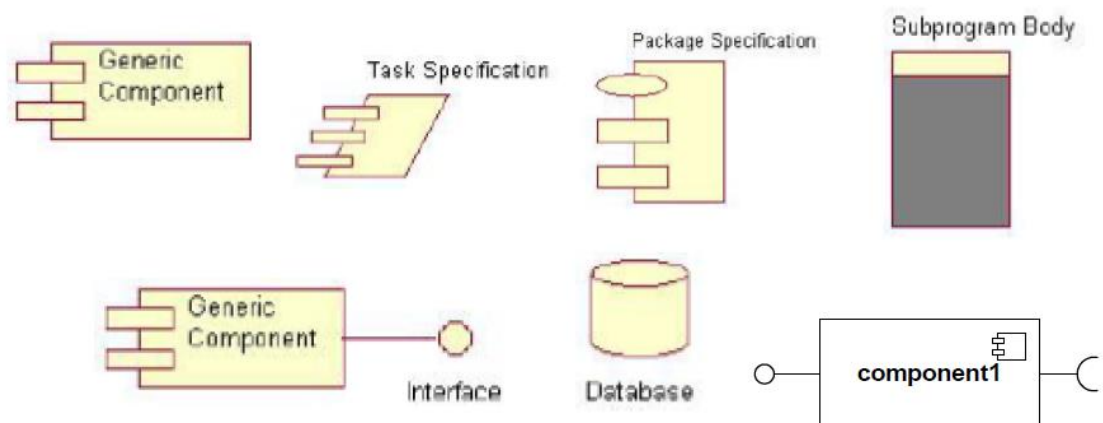


Figura 28 Diagrama de Componentes: Artefacto - COMPONENTE

2.1.1.3.2.1.6 Diagrama de Despliegue

La vista de despliegue es concernida con la implementación física de la aplicación. Esto incluye asuntos como el diseño de la red y la localización de los componentes en la red. También consideraremos los asuntos de implementación tales como cuánto de ancho de banda de la red tenemos, cuántos usuarios concurrentes podemos esperar, que hacemos si un servidor cae, y así sucesivamente.

La vista de despliegue contiene procesadores, dispositivos, procesos, y conexiones entre procesadores y dispositivos. Toda esta información es diagramada en un diagrama de despliegue. Hay sólo un diagrama de Despliegue por sistema, y por consiguiente un diagrama de Despliegue por el modelo Rose.

Un diagrama de Despliegue muestra todos los nodos en la red, las conexiones entre ellos, y los procesos que correrán en cada uno.

a) **Artefactos:**

- **NODO:** Representación de un elemento físico de la arquitectura de la aplicación. Se toman como elementos nodo los siguientes:

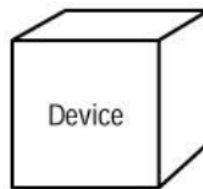


Figura 29 Diagrama de Despliegue: Artefacto – NODO

- **Procesador.** Un procesador es cualquier máquina que tiene poder de procesamiento. Los servidores, las estaciones de trabajo, y otras máquinas con procesadores son incluidos en esta categoría.
- **Dispositivo.** Un dispositivo es un hardware con un propósito único o un propósito limitado. Los dispositivos incluyen artículos como terminales tontas, impresoras, y escáneres.

Ambos procesadores y dispositivos también pueden ser llamados nodos en una red.

- **Conexiones.** Una conexión es un enlace físico entre dos procesadores, dos dispositivos, o un procesador y un dispositivo. Más comúnmente, las conexiones representan las conexiones físicas de la red entre los nodos en su red. Una conexión también puede ser un enlace de la Internet entre dos nodos.
- **Procesos.** Un proceso es un hilo único de ejecución que funciona con un procesador. Un archivo ejecutable, por ejemplo, es considerado un proceso. Cuando adicionamos procesos para el diagrama, se enfoca en sólo los procesos que guardan relación con el sistema ha ser construido. Los procesos pueden ser mostrados en un diagrama de Despliegue u oculto de la vista. Si son mostrados, se encuentran enumerados directamente debajo del procesador (s) en el cual son ejecutados. Los procesos pueden ser asignados con una prioridad. Si el procesador en el cual son ejecutados usos de la planificación

2.1.1.4 Herramientas de Construcción de Software

2.1.1.4.1 Sublime Text 2³

Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo VI llamado Vintage mode.

³ www.sublimetext.com/2/docs/

Se distribuye de forma gratuita, sin embargo no es software libre o de código abierto, se puede obtener una licencia para su uso ilimitado, pero el no disponer de ésta no genera ninguna limitación más allá de una alerta cada cierto tiempo.

Características:

- **Mini mapa:** consiste en una pre visualización de la estructura del código, es muy útil para desplazarse por el archivo cuando se conoce bien la estructura de este.
- **Multi Selección:** Hace una selección múltiple de un término por diferentes partes del archivo.
- **Multi Cursor:** Crea cursores con los que podemos escribir texto de forma arbitraria en diferentes posiciones del archivo.
- **Multi Layout:** Trae siete configuraciones de plantilla podemos elegir editar en una sola ventana o hacer una división de hasta cuatro ventanas verticales o cuatro ventanas en cuadrícula.
- **Soporte nativo para infinidad de lenguajes:** Soporta de forma nativa lenguajes de programación y texto plano.
- **Syntax Highlight configurable:** El remarcado de sintaxis es completamente configurable a través de archivos de configuración del usuario.
- **Búsqueda Dinámica:** Se puede hacer búsqueda de expresiones regulares o por archivos, proyectos, directorios, una conjunción de ellos o todo a la vez.
- **Auto completado y marcado de llaves:** Se puede ir a la llave que cierra o abre un bloque de una forma sencilla.
- **Soporte de Snippets y Plugins:** Los snippets son similares a las macros o los bundles además de la existencia de multitud de plugins.
- **Configuración total de Keybindings:** Todas las teclas pueden ser sobrescritas a nuestro gusto.
- **Acceso rápido a línea o archivo:** Se puede abrir un archivo utilizando el conjunto de teclas Cmd+P en Mac OS X o Ctrl+P en Windows y Linux y

escribiendo el nombre del mismo o navegando por una lista. También se puede ir a una línea utilizando los dos puntos ":" y el número de línea.

- **Paleta de Comandos:** Un intérprete de Python diseñado solo para el programa con el cual se puede realizar infinidad de tareas.
- **Coloreado y envoltura de sintaxis:** Si se escribe en un lenguaje de programación o marcado, resalta las expresiones propias de la sintaxis de ese lenguaje para facilitar su lectura.
- **Pestañas:** Se pueden abrir varios documentos y organizarlos en pestañas.
- **Resaltado de paréntesis e indentación:** cuando el usuario coloca el cursor en un paréntesis, corchete o llave, resalta esta y el paréntesis, corchete o llave de cierre o apertura correspondiente.

Lenguajes soportados: ActionScript, AppleScript, ASP, Batch File, C, C#. C++, Clojure, CSS, D, Diff, Erlang, Expresión regular, Go, Graphviz (DOT), Groovy, Haskell, **HTML**, Java, **JavaScript**, LaTeX, Lua, Lisp, Makefile, Markdown, MATLAB, Objective-C, Ocaml, Perl, **PHP**, Python, R, Rails, ReStructuredText, Scala, Shell Script (Bash), **SQL**, Tcl, Texto plano, Textile, XML, XSL, YAML



Figura 30 2.1.1.4 Herramientas de Construcción de Software: Sublime Text 3

2.1.1.4.2 XAMPP⁴

XAMPP es una plataforma de desarrollo de aplicaciones web de software libre. Su nombre se debe a cuatro características que comprende esta herramienta:

- X** Esta disponible para distintos Sistemas Operativos. Linux – Windows – Mac OS* - Solaris* (* Todavía son experimentales).
- A** Emplea el Servidor Wb Apache.
- M** Incluye el sistema Gestor de Base de Datos MySQL.
- P** Incluye los lenguajes de programación PHP y Perl
- P**

XAMPP para Windows utiliza las siguientes Herramientas o Tecnologías:

- Apache 2.4.3
- MySQL 5.5.27
- PHP 5.4.7
- phpMyAdmin 3.5.2.2
- Tomcat 7.0.30 (with mod_proxy_ajp connector)
- XAMPP Control Panel 3.1.0 (from hackattack 142)

Los creadores de XAMPP nos dejan muy claro que el propósito de XAMPP no es montar un sistema en producción si no un sistema de desarrollo por ello XAMPP deja tan abierto como es posible para permitir hacer a los desarrolladores lo que desearan.

La seguridad de XAMPP se puede comprobar con la opción “Chequeo de seguridad” que muestra un pagina en la que se resume el estado de seguridad de los diferentes componentes de XAMPP. Pero XAMPP no da la opción para configurar la seguridad de MySQL y XAMPP.

⁴ www.apachefriends.org/en/xampp-windows.html



Figura 31 Herramientas de Construcción de Software: XAMPP

2.1.1.4.3 Enterprise Architect 8.0⁵

Enterprise Architect (EA) es una herramienta de práctica bien definido para la realización de análisis de la empresa, el diseño, la planificación y la ejecución, utilizando un enfoque holístico en todo momento, para el éxito del desarrollo y ejecución de la estrategia. Enterprise Architect aplica principios y prácticas para orientar la arquitectura organizaciones a través de los negocios, la información, los procesos y los cambios tecnológicos necesarios para ejecutar sus estrategias. Estas prácticas utilizan los diversos aspectos de una empresa para identificar, motivar y lograr estos cambios.

Los profesionales del Enterprise Architect llaman a sí mismos arquitectos de la empresa . Un arquitecto de la empresa es una persona responsable de la realización de este complejo análisis de la estructura y los procesos de negocio y se llama a menudo para sacar conclusiones de la información recopilada. Mediante la producción de este entendimiento, los arquitectos están tratando de abordar los objetivos de la arquitectura empresarial: Eficacia, Eficiencia, agilidad y durabilidad.



Figura 32 Herramientas de Construcción de Software: Enterprise Architect

⁵ www.enterprisearchitect.org/docs/

2.1.1.4.4 MySQL⁶

MySQL es el servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB. MySQL AB es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL.

MySQL es un sistema de administración de bases de datos. MySQL usa la licencia GPL (Licencia Pública General GNU), para definir qué es lo que se puede y no se puede hacer con el software para diferentes situaciones.

El software de bases de datos MySQL consiste de un sistema cliente/servidor que se compone de un servidor SQL multihilo, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (APIs). Se puede obtener también como una biblioteca multihilo que se puede enlazar dentro de otras aplicaciones para obtener un producto más pequeño, más rápido, y más fácil de manejar.



Figura 33 MySQL

⁶ dev.mysql.com/

2.1.1.5 Técnicas y Entornos de Trabajo

2.1.1.5.1 JSON (JavaScript Object Notation)

JSON, acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON. En JavaScript, un texto JSON se puede analizar fácilmente usando la función `eval()`, lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web.

Cada vez hay más soporte de JSON mediante el uso de paquetes escritos por terceras partes. La lista de lenguajes soportados incluye ActionScript, C, C++, C#, ColdFusion, Common Lisp, Delphi, E, Eiffel, Java, JavaScript, ML, Objective-C, Objective CAML, Perl, **PHP**, Python, Rebol, Ruby, Lua y Visual FoxPro.

El término JSON está altamente difundido en los medios de programación, sin embargo, es un término mal descrito ya que en realidad es solo una parte de la definición del estándar ECMA-262 en que está basado Javascript.

2.1.1.5.2 CodeIgniter⁷

CodeIgniter es un Marco de desarrollo de aplicaciones, un conjunto de herramientas, para las personas que crean sitios web utilizando PHP. Su objetivo es permitirle

⁷ www.codeigniter.com/

desarrollar proyectos mucho más rápido de lo que podría hacerlo si estuviera escribiendo código desde cero, proporcionando un amplio conjunto de bibliotecas para las tareas más comunes, así como una interfaz simple y una estructura lógica para acceder a estas bibliotecas. CodeIgniter le permite centrarse creativamente en su proyecto al minimizar la cantidad de código necesario para una tarea determinada.



Figura 34 Framework CodeIgniter

¿Para quién es CodeIgniter?

CodeIgniter es adecuado si:

- Quiere un marco con una huella pequeña.
- Necesita un rendimiento excepcional.
- Necesita una amplia compatibilidad con cuentas de hosting estándar que ejecutan una variedad de versiones y configuraciones de PHP.
- Desea un marco que requiera una configuración casi nula.
- Desea un marco que no requiera el uso de la línea de comando.
- Desea un marco que no requiera que se adhiera a reglas de codificación restrictivas.
- No está interesado en bibliotecas monolíticas de gran escala como PEAR.
- No querrá verse forzado a aprender un lenguaje de plantillas (aunque un analizador de plantillas está opcionalmente disponible si lo desea).
- Evita la complejidad, favoreciendo soluciones simples.
- Necesita documentación clara y completa.

2.1.1.6 Base de Datos

2.1.1.6.1 Componentes principales de una Base de Datos

Datos: Los datos son la Base de Datos propiamente dicha.

Hardware: Se refiere a los dispositivos de almacenamiento en donde reside la base de datos así como los dispositivos periféricos (Unidad de Control, Canales de Comunicación, etc.) necesarios para su uso.

Software: Está constituido por un conjunto de programas que se conoce como Sistema Manejador de Base de Datos (DBMS), manejando éste todas las solicitudes formuladas por los usuarios a la base de datos.

Usuarios: Normalmente identificándose 3 tipos:

- El programador de aplicaciones.
- El usuario Final.
- El Administrador de la Base de Datos quien se encarga del control general del Sistema de Base de Datos.

2.1.1.6.2 Herramienta phpMyAdmin Administrador para el manejo de la BD⁸

phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 62 idiomas. Se encuentra disponible bajo la licencia GPL Versión 2.

⁸ www.phpmyadmin.net/

Este proyecto se encuentra vigente desde el año 1998, siendo el mejor evaluado en la comunidad de descargas de SourceForge.net como la descarga del mes de diciembre del 2002. Como esta herramienta corre en máquinas con Servidores Webs y Soporte de PHP y MySQL, la tecnología utilizada ha ido variando durante su desarrollo.

Especificaciones: Las especificaciones proveídas por el programa incluyen:

- Interface Web
- Manejador de base de datos MySQL, MariaDB y Drizzle
- Importación de datos desde CSV y SQL
- Exporta datos a varios formatos: CSV, SQL, XML, PDF (via la biblioteca TCPDF), ISO/IEC 26300 - OpenDocument Text y Spreadsheet, Word, Excel, LaTeX y otros
- Administración de múltiples servidores
- Crea gráficos PDF del diseño de la base de datos
- Crea consultas complejas usando Query-by-Example (QBE)
- Búsqueda global en una base de datos o un subconjunto de esta
- Transforma datos almacenados a cualquier formato usando un conjunto de funciones predefinidas, tal como BLOB
- Live charts para monitoriar las actividades del servidor MySQL tales como conexiones, procesos, uso de CPU/Memoria, etc.

2.1.1.7 Sistema de Información Automatizado⁹

2.1.1.7.1 La Internet

Algunos definen Internet como "La Red de Redes", y otros como "La Autopista de la Información".

⁹ blogs.rimed.cu/gestion-informacion/2010/12/20/tic-y-gestion-de-la-informacion/

Efectivamente, Internet es una Red de Redes porque está hecha a base de unir muchas redes locales de ordenadores, o sea de unos pocos ordenadores en un mismo edificio o empresa.

Por la Red Internet circulan constantemente cantidades increíbles de información. Por este motivo se le llama también La Autopista de la Información. Hay mas 30.000 millones de "Internautas", es decir, de personas que "navegan" por Internet en todo el Mundo. Se dice "navegar" porque es normal el ver información que proviene de muchas partes distintas del Mundo en una sola sesión.

Una de las ventajas de Internet es que posibilita la conexión con todo tipo de ordenadores, desde los personales, hasta los más grandes que ocupan habitaciones enteras. Incluso podemos ver conectados a la Red cámaras de vídeo, robots, y máquinas de refrescos, etc.

2.1.1.7.2 La World Wide Web

La World Wide Web consiste en ofrecer una interface 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 no interesa. Lo más interesante es que con unas pocas ordenes se puede mover por toda la Internet.

La World Wide Web permite una manera más organizada de acceder a la información disponible en Internet, presentando una interfaz amigable con el usuario mediante navegadores como Netscape, Mosaic y Microsoft Internet Explorer, Mozilla FireFox.

El surgimiento de la World Wide Web ha ayudado a un crecimiento considerable de Internet en la actualidad. Compañías pequeñas, empresas

grandes, ayuntamientos, estados, gobiernos de distintos países, universidades, bibliotecas, están presentes en Internet.

Es básicamente un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir, la web es un sistema de hipertexto que utiliza Internet como su mecanismo de transporte o desde otro punto de vista, una forma gráfica de explorar Internet.

2.1.1.8 Sistemas de Información vía Web

La evolución de Internet como red de comunicación global y el surgimiento y desarrollo del Web como servicio imprescindible para compartir información, creó un excelente espacio para la interacción del hombre con la información hipertextual, a la vez que sentó las bases para el desarrollo de una herramienta integradora de los servicios existentes en Internet. Los sitios Web, como expresión de sistemas de información, deben poseer los siguientes componentes:

- Usuarios.
- Mecanismos de entrada y salida de la información.
- Almacenes de datos, información y conocimiento.
- Mecanismos de recuperación de información.

Pudiésemos definir entonces como sistema de información al conjunto de elementos relacionados y ordenados, según ciertas reglas que aporta al sistema objeto, es decir, a la organización a la que sirve y que marca sus directrices de funcionamiento, la información necesaria para el cumplimiento de sus fines; para ello, debe recoger, procesar y almacenar datos, procedentes tanto de la organización como de fuentes externas, con el propósito de facilitar su recuperación, elaboración y presentación. Actualmente, los sistemas de información se encuentran al alcance de las grandes masas de usuarios por medio de Internet; así se crean las bases de un nuevo modelo, en el que los usuarios interactúan directamente con los sistemas de información para satisfacer sus necesidades de información.

2.1.2 Plan de Desarrollo del Software

2.1.2.1 Introducción

Este Plan de Desarrollo del Software es una versión corregida para ser incluida en la propuesta elaborada como respuesta al proyecto de la asignatura de Taller III de la Carrera de Ingeniería Informática de la Facultad de Ciencias y Tecnología de la Universidad Autónoma Juan Misael Saracho. Este documento provee una visión global del enfoque de desarrollo propuesto.

El proyecto será desarrollado por el universitario: Antonio Lujan Huanca Mamani basado en una metodología de Rational Unified Process (RUP) en la que se procederá a cumplir con las cuatro fases que marca la metodología. Es importante destacar esto puesto que utilizaremos la terminología RUP en este documento. Se incluirá el detalle para las fases de Inicio y Elaboración, adicionalmente se esbozarán las fases posteriores de Construcción para dar una visión global de todo proceso.

El enfoque del desarrollo propuesto constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los artefactos de RUP.

2.1.2.1.1 Propósito

El propósito del Plan de Desarrollo de Software es dar a conocer la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del software.

Los usuarios del Plan de Desarrollo del Software son:

- El Director del proyecto lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.

- Los miembros del equipo de desarrollo lo usan para entender lo qué deben hacer, cuándo deben hacerlo y qué otras actividades dependen de ello.
- El docente para evaluar el cumplimiento del proyecto.

2.1.2.1.2 Alcance

Con el Plan de Desarrollo del Software se pretende analizar y elaborar un proyecto de gran magnitud abarcando todas las fases requeridas en la terminología RUP. Dichas fases tendrán un cronograma de cada una de las actividades a realizar. También se especificará los detalles de construcción del proyecto para los distintos roles que cumplen los desarrolladores.

2.1.2.1.3 Resumen

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

- Vista General del Proyecto, proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.
- Organización del Proyecto, describe la estructura organizacional del equipo de desarrollo.
- Gestión del Proceso, explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.
- Planes y Guías de aplicación, proporciona una vista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.

2.1.2.2 Vista general del proyecto

2.1.2.2.1 Propósito, Alcance y Objetivos

La información que a continuación se incluye ha sido extraída de reuniones, entrevistas y charlas que se han realizado con los usuarios destino de la Brigada Parlamentaria de Tarija.

2.1.2.2.1.1 Propósito

Con el presente proyecto se pretende Mejorar la “Gestión administrativa de la Brigada Parlamentaria de Tarija”.

2.1.2.2.1.2 Alcance

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo de un Sistema Informático que está comprendido por los siguientes Módulos:

- Generar CITE
- Administrar documentos
- Comisiones
- Directivas
- Pleno de la Brigada
- Personal Administrativo
- Usuarios
- Reportes
- Buscar CITE

Analizando factores predominantes dentro de la institución, podemos llegar a la conclusión de que la implementación del sistema automatizado dará beneficios claramente perceptibles, dando soluciones a problemas que arrastra; este sistema proyectará una solución a mediano plazo que beneficiará a dicha institución.

Esta propuesta de sistema (Software) contiene una serie de alternativas de mejoramiento para las expectativas futuras de la institución, las cuales se detallan a continuación:

- Brindar seguridad al sistema mediante una clave de ingreso, permitiendo el acceso al mismo sólo al personal autorizado y dejando un registro de huellas de los movimientos que se realiza en el sistema.
- Desarrollar un manual de usuario.

2.1.2.2.1.3 Objetivo

2.1.2.2.1.3.1 Objetivo General

“Gestión administrativa de la Brigada Parlamentaria de Tarija” Mejorada

2.1.2.2.1.3.2 Objetivo Específico

- Obtener un sistema para el uso en la gestión de la información de la Institución.
- Crear una base de datos de acuerdo a las necesidades de la Institución.
- Brindar información al usuario sobre los diferentes procesos de la información.
- Implementar una interfaz para que la población pueda realizar un seguimiento a las diferentes solicitudes o propuestas y tener a mano toda la información pública que brinda la Institución.

2.1.2.2.2 Suposiciones y Restricciones

Acontecimientos que deben ocurrir para que el proyecto sea ejecutado con éxito pero que están totalmente fuera del ámbito del control del equipo de proyecto.

2.1.2.2.2.1 Suposiciones

Suponemos que la Brigada Parlamentaria, cuenta con:

- La Institución dispone equipos de computación, que incluye un computador, impresora y scanner.
- Total disponibilidad de los miembros de la Institución para proporcionar información.
- Personal con conocimientos básicos de computación.

- Presupuesto suficiente para la implementación del Sistema.
- Interés de la institución para utilizar el Sistema.
- Un servidor Web para la implementación del sistema.
- El administrador deberá contar con el conocimiento necesario para el manejo del sistema.
- Contar con un servidor Web que nos presente las características mínimas para el correcto funcionamiento del sistema.

2.1.2.2.2 Restricciones

Limitaciones generalmente fuera del ámbito de control del equipo de proyecto que pueden afectar negativamente a su alcance.

- El administrador deberá contar con el conocimiento necesario para el manejo del sistema.
- Contar con un servidor Web que presente las características mínimas para el correcto funcionamiento del sistema.
- El sistema será restringido, sólo usuarios privilegiados podrán acceder al sistema.
- Para la manipulación de la base de datos, sólo podrán acceder el personal autorizado.
- El sistema permitirá cargar documentos en digital con los formatos PNG, JPG y PDF
- El sistema no abarca:
 - Control de Personal.
 - Gestión de Inventario.
 - Control del área contable.
 - Evaluación de la situación financiera.

2.1.2.2.3 Entregables del Proyecto

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta

lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

2.1.2.2.3.1 Plan de Desarrollo del Software

Es el presente documento

2.1.2.2.3.2 Diagrama de Caso de Uso del Negocio

2.1.2.2.3.2.1 Modelo de casos de Uso de la Secretaria

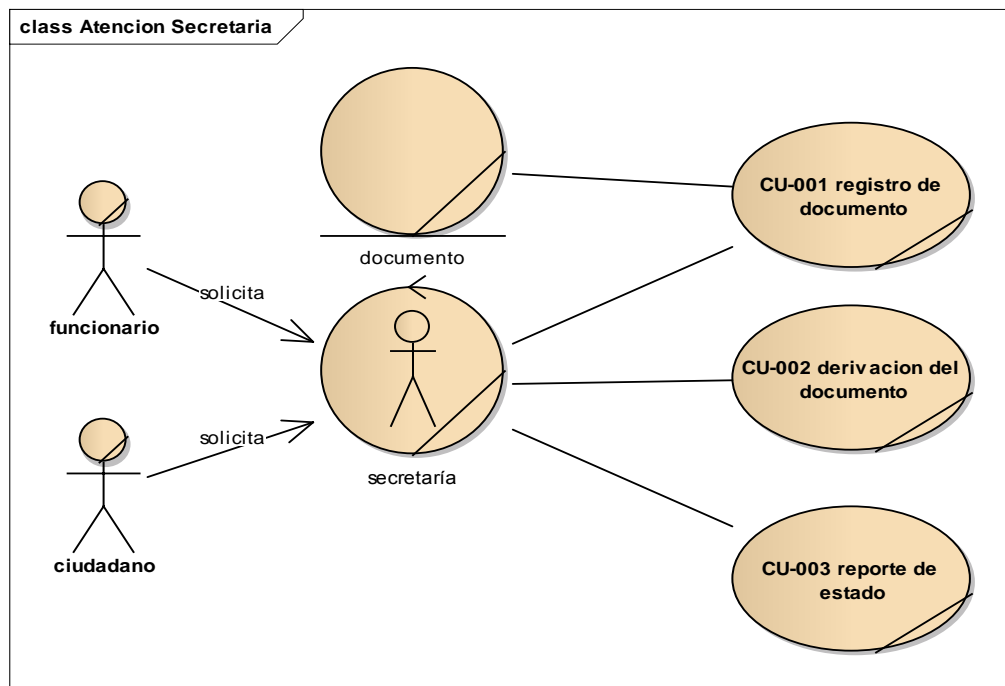


Figura 35 Modelo de Casos de Uso del Negocio de la Secretaria

2.1.2.2.3.3 Modelo de Objetos del Negocio

Es un modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo (workflows) asociados al caso de uso del negocio. Para la representación de este modelo se utilizan Diagramas de Colaboración (para mostrar actores externos, internos y las entidades (información) que manipulan, un Diagrama de Clases para mostrar gráficamente las entidades del sistema y sus relaciones, y Diagramas de Actividad para mostrar los flujos de trabajo.

2.1.2.2.3.3.1 Introducción

El Modelo de Modelo de Objetos del Negocio es un artefacto de la disciplina Requisitos en la metodología RUP la cual estamos implementando.

2.1.2.2.3.3.2 Propósito

Comprender la Estructura dinámica de los Casos de Uso de Negocio.

2.1.2.2.3.3.3 Alcance

Describe los procesos del negocio.

Identificar y definir los objetos de negocio.

2.1.2.2.3.3.4 Diagramas de Objetos del Negocio

2.1.2.2.3.3.4.1 Modelo de Objeto del Negocio:

Registrar Documento

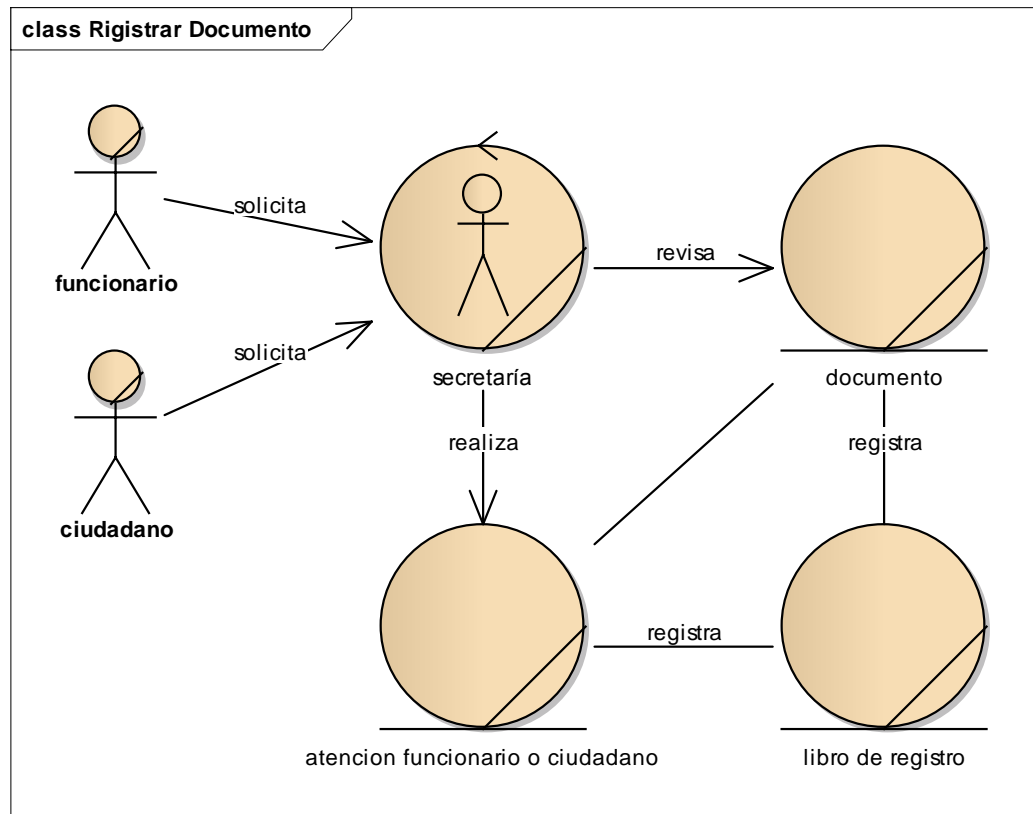


Figura 36 Modelo de Objeto del Negocio: Registrar Documento

2.1.2.2.3.3.4.2 Modelo de Objeto del Negocio: Derivacion de documento

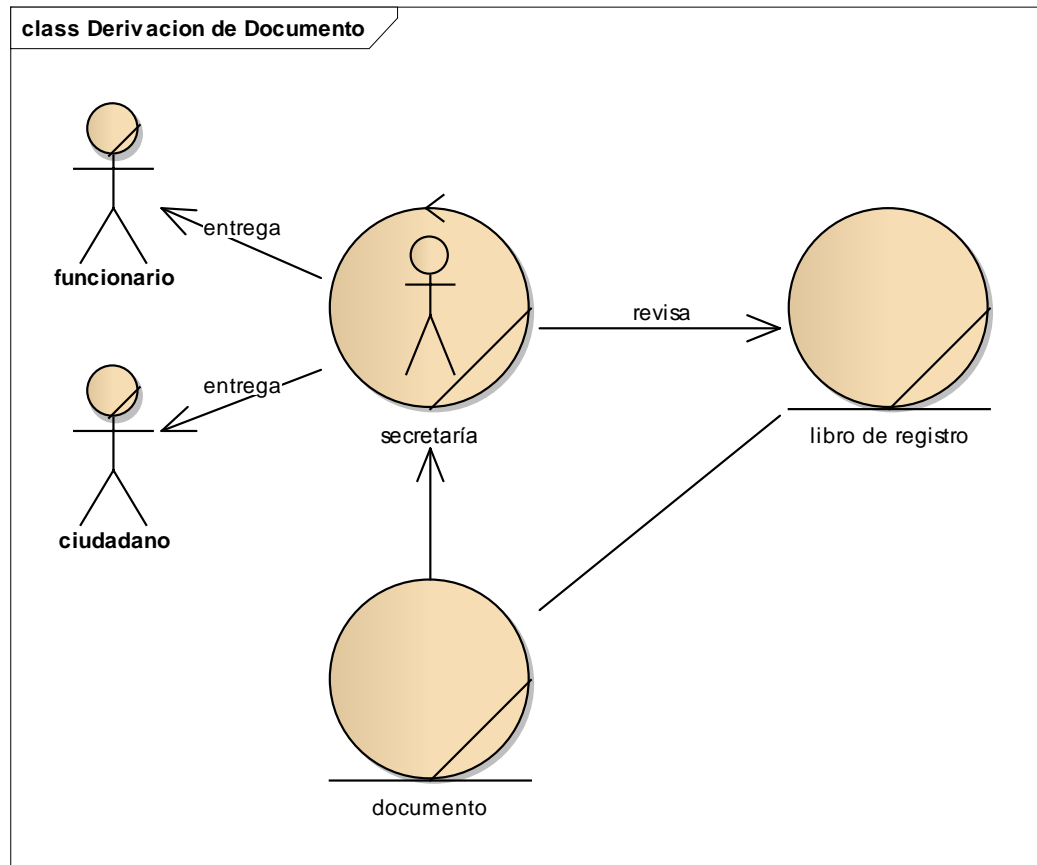


Figura 37 Modelo de Objeto del Negocio: Derivacion de documento

2.1.2.2.3.3.4.3 Modelo de Objeto del Negocio: Reporte de estado de estado

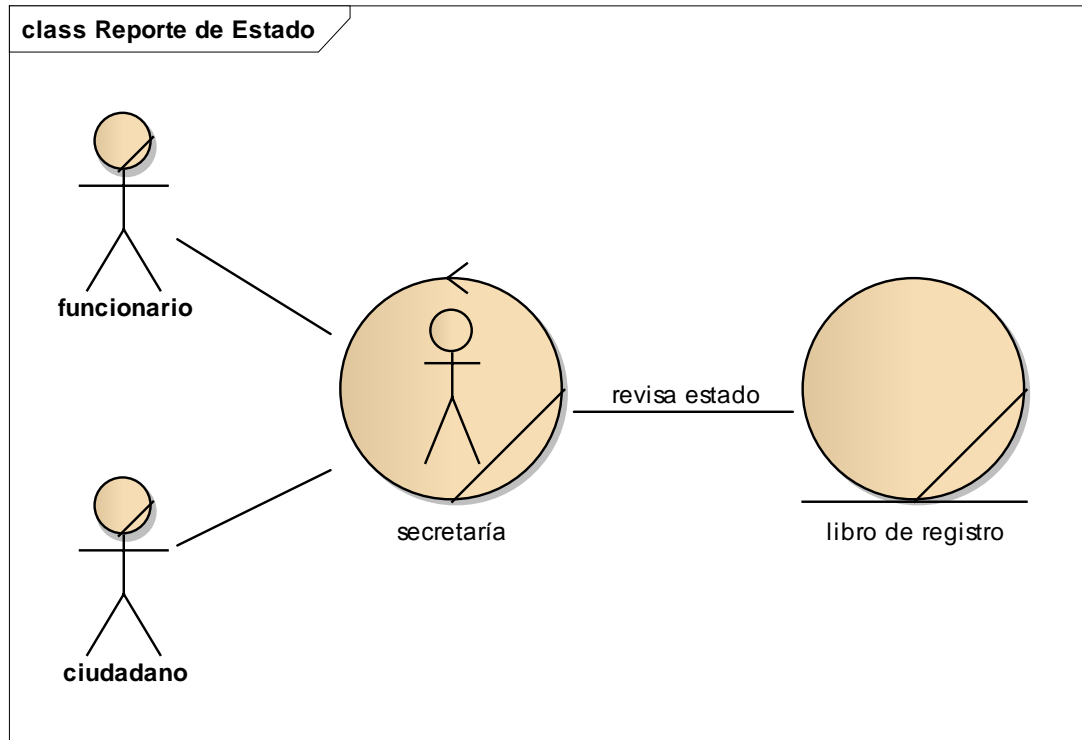


Figura 38 Modelo de Objeto del Negocio: Reporte de estado

2.1.2.2.3.4 Glosario

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.

2.1.2.2.3.4.1 Introducción

El presente documento recoge los términos manejados durante la elaboración del proyecto de desarrollo del Sistema Informático. Se trata de un diccionario informal de datos y de definiciones de la nomenclatura que se maneja en la construcción del Sistema.

2.1.2.2.3.4.2 Propósito

El propósito del presente documento es definir la terminología manejada en el proyecto a desarrollar, también sirve como guía de consulta para la clarificación de los puntos conflictivos o poco esclarecidos del proyecto.

2.1.2.2.3.4.3 Alcance

El alcance del presente documento se extiende a todo el proyecto en desarrollo.

2.1.2.2.3.4.4 Organización del Glosario

El presente documento está organizado por definiciones de términos ordenados para una comprensión clara.

2.1.2.2.3.4.5 Definiciones

2.1.2.2.3.4.5.1 Calidad

Conjunto de características que debe cumplir un servicio o producto en base a normativa preestablecida y que responda a las expectativas y necesidades de los usuarios del servicio.

2.1.2.2.3.4.5.2 Parlamentario

Un miembro del parlamento, es el representante de los votantes en un parlamento. En congresos bicamerales, esta categoría incluye específicamente a los miembros de la cámara baja.

2.1.2.2.3.4.5.3 Ciudadano

Un ciudadano es una persona considerada como miembro de un estado titular de derechos civiles y políticos, sometido a sus leyes.

2.1.2.2.3.4.5.4 Documento

Es cualquier documento formulado por cualquier medio expresivo, como

medio específico para probar, describir, comunicar y/o informar algo dentro de la estructura, la función, la actividad, y los tramites de cualquier entidad pública en cualquier parte y en cualquier época.

2.1.2.2.3.4.5.5 Documento Interno

Son aquellos que se originan y manejan al interior de la Institución.

2.1.2.2.3.4.5.6 Documento Externo

Son aquellos que provienen de diferentes organismos nacionales e internacionales, instituciones públicas, privadas y/o personas naturales.

2.1.2.2.3.4.5.7 Registro

Medio por el cual se escribe la Información de la correspondencia.

2.1.2.2.3.4.5.8 Correspondencia

Documento utilizado para transmisión escrita de información interinstitucional e interinstitucional.

2.1.2.2.3.4.5.9 Derivacion

Proceso de asignación por parte de un funcionario activo al responsable de dar respuesta a un determinado asunto.

2.1.2.2.3.4.5.10 Documento en Proceso

Conjunto de asuntos y/o expedientes ya atendidos, cuya atención se encuentra aun en curso.

2.1.2.2.3.4.5.11 Documento Finalizado

Conjunto de asuntos y/o expedientes ya atendidos, concluidos.

2.1.2.2.3.5 Modelo de Casos de Uso

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

2.1.2.2.3.5.1 Introducción

El presente documento es un artefacto de la disciplina Requisitos en la metodología RUP la cual estamos implementando.

2.1.2.2.3.5.2 Propósito

- Comprender la estructura y la dinámica del sistema desarrollado.
- Identificar el nivel de complejidad del sistema.
- Identificar posibles mejoras.

2.1.2.2.3.5.3 Alcance

Identificar y definir procesos del sistema según los objetivos de la organización.

Definir un Caso de Uso para cada proceso del sistema (el diagrama de caso de uso nos detalla el contexto y los límites de la organización).

Diagramas de Casos de Uso.

2.1.2.2.3.5.4 Diagramas de Caso de Uso

2.1.2.2.3.5.4.1 Casos de Usos del Sistema General

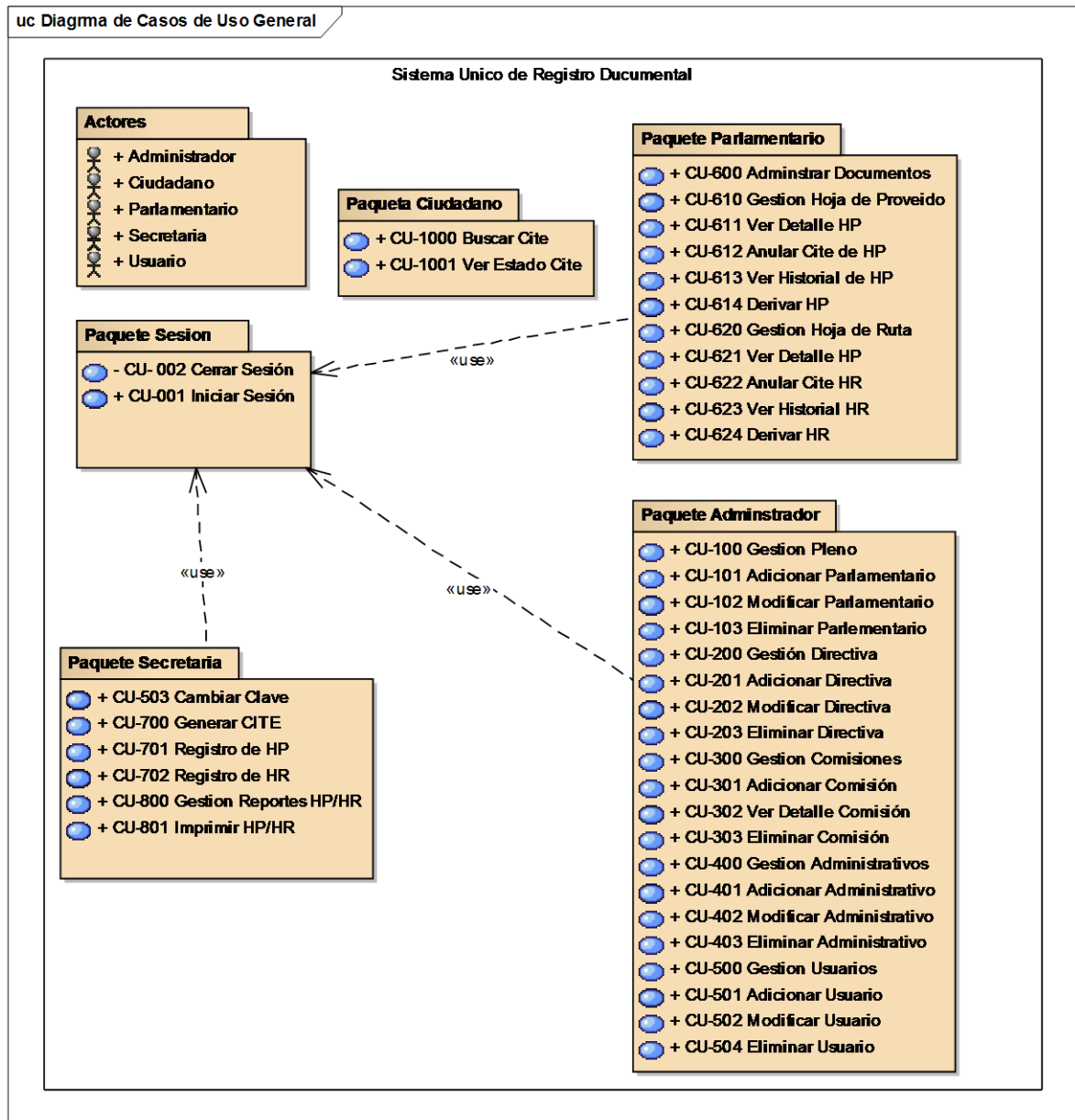


Figura 39 Casos de Usos del Sistema General

2.1.2.2.3.5.4.1.1 Modelo Casos de Uso del Módulo Administrador

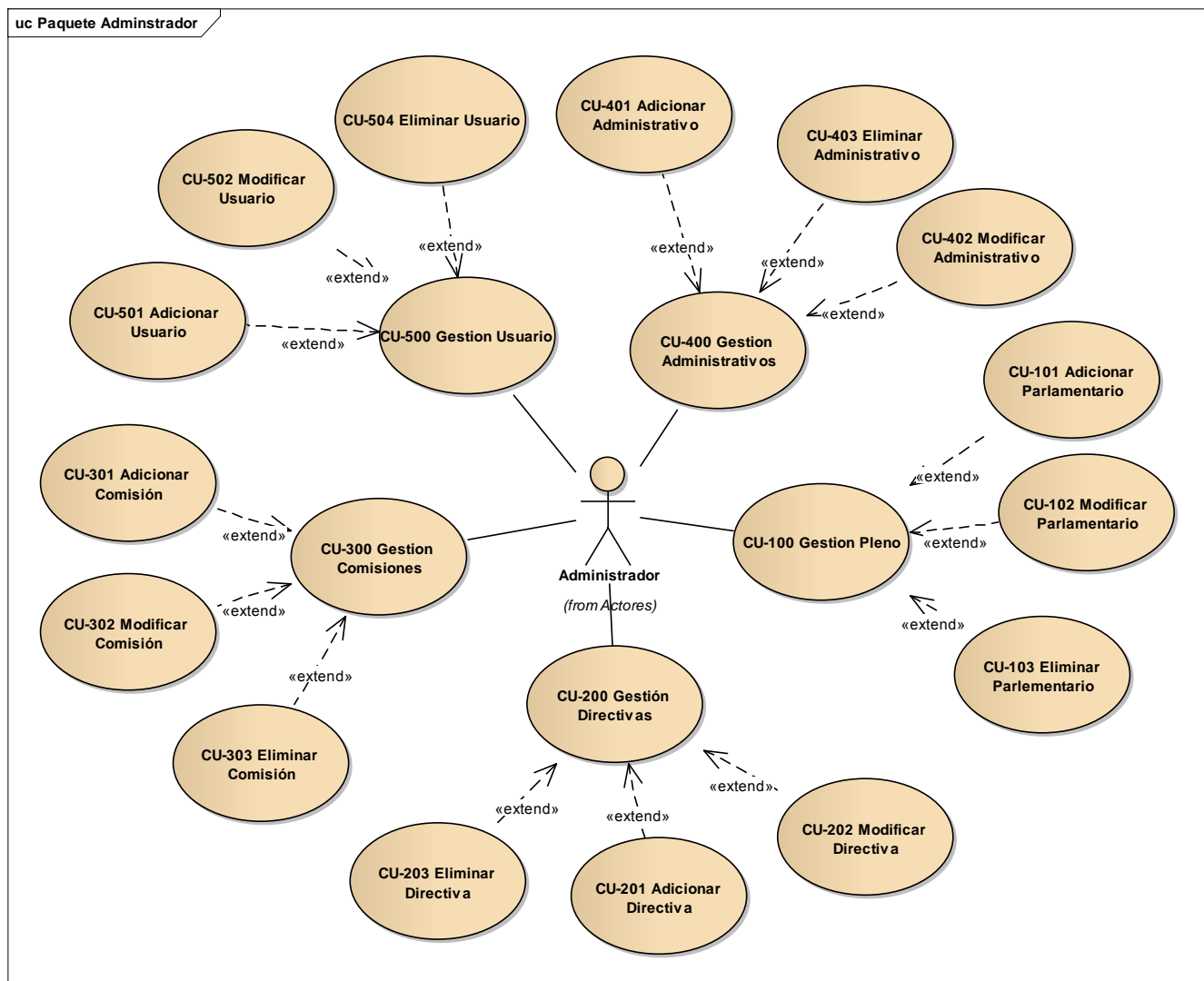


Figura 40 Modelo Casos de Uso del Módulo Administrador

2.1.2.2.3.5.4.1.2 Modelo Casos de Uso del Módulo Parlamentario

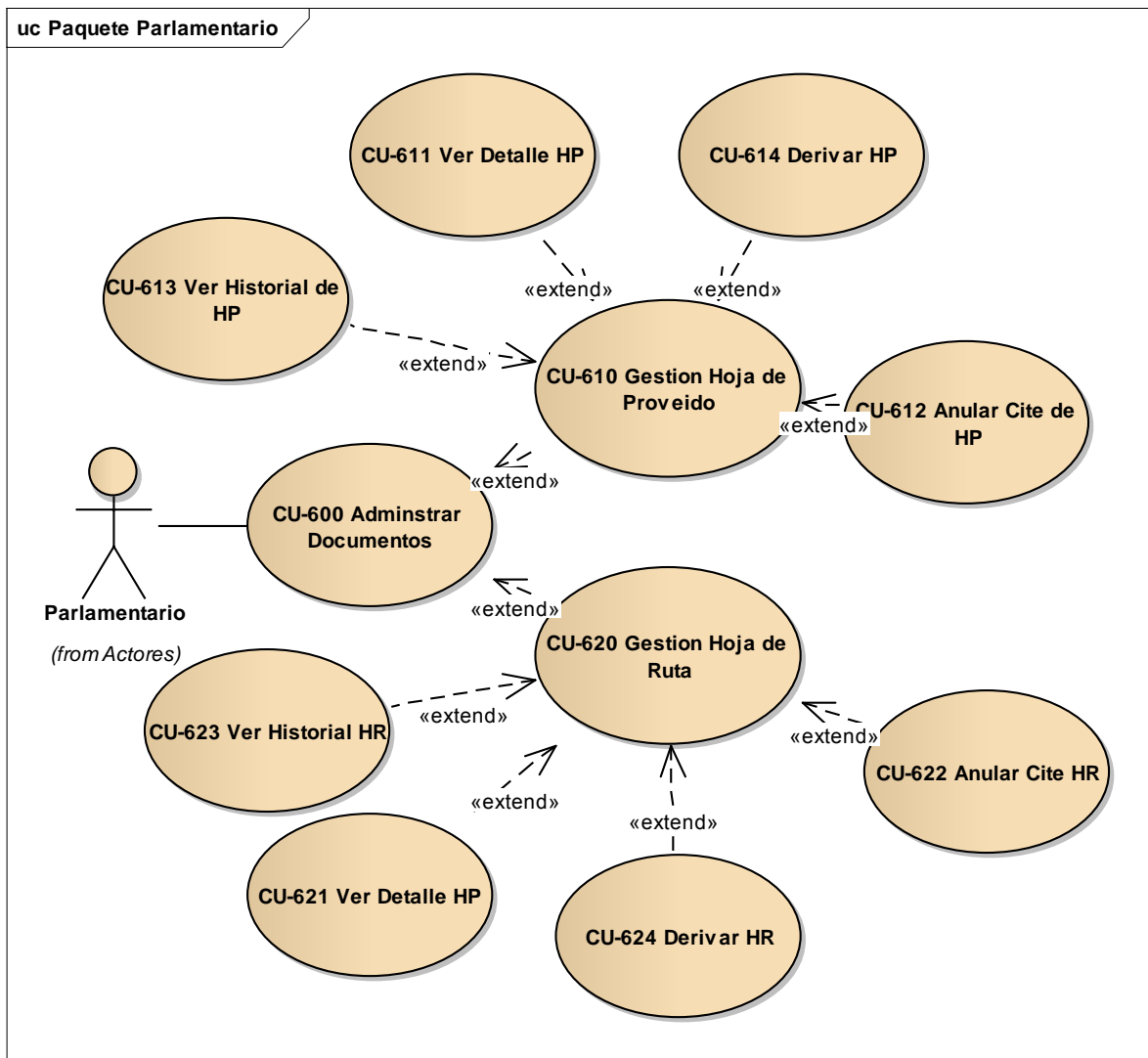


Figura 41 Modelo Casos de Uso del Módulo Parlamentario

2.1.2.2.3.5.4.1.3 Modelo Caso de Uso del Módulo Secretaria

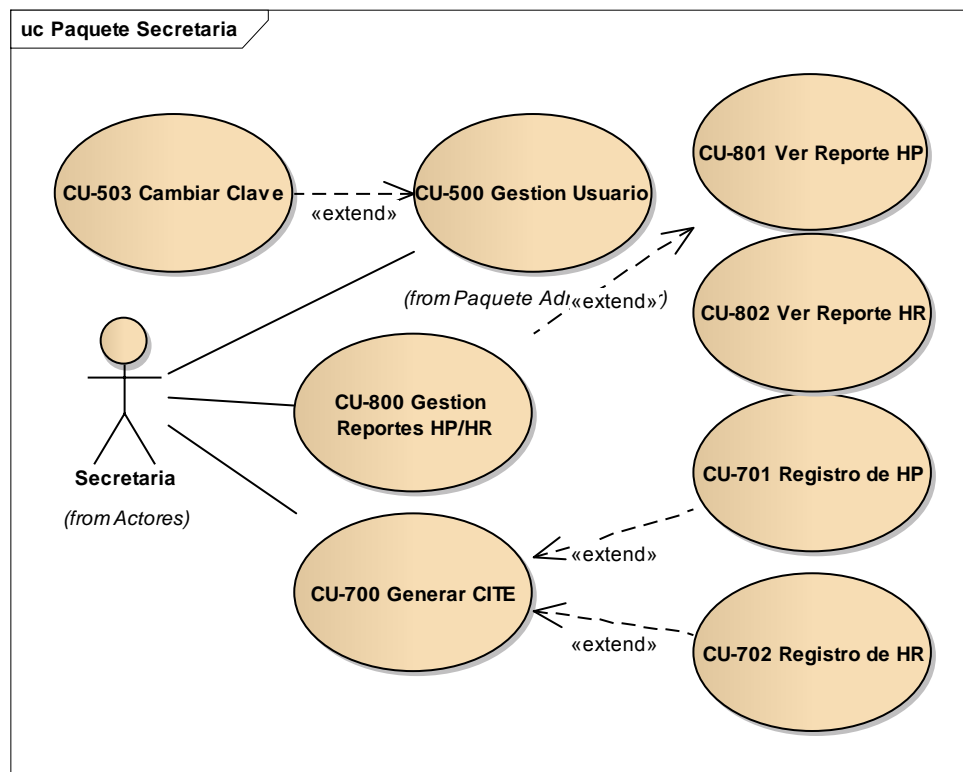


Figura 42 Modelo Caso de Uso del Módulo Secretaria

2.1.2.2.3.5.4.1.4 Modelo Caso de Uso del Módulo Ciudadano

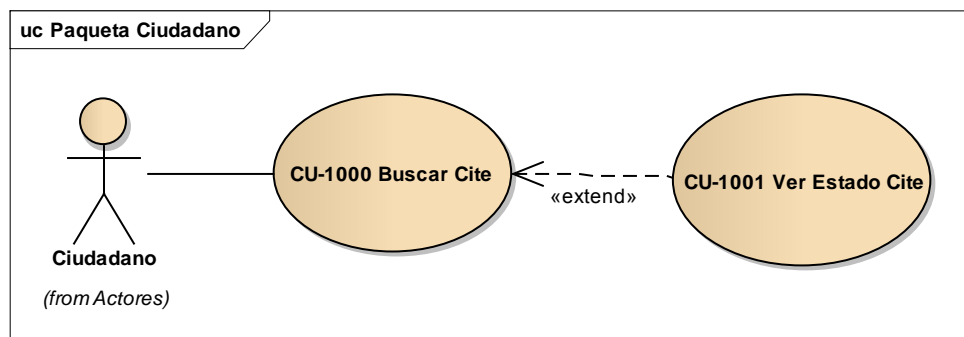


Figura 43 Modelo Caso de Uso del Módulo Ciudadano

2.1.2.2.3.5.4.1.5 Modelo Caso de Uso del Módulo Sesión

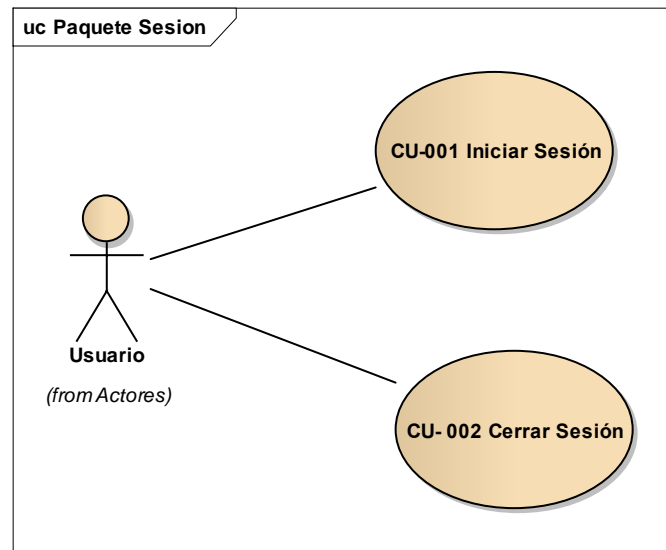


Figura 44 Modelo Caso de uso del Módulo Sesión

2.1.2.2.3.5.4.2 Casos de Usos del Sistema Específicos

2.1.2.2.3.5.4.2.1 Modelo Caso de Uso Gestión Pleno

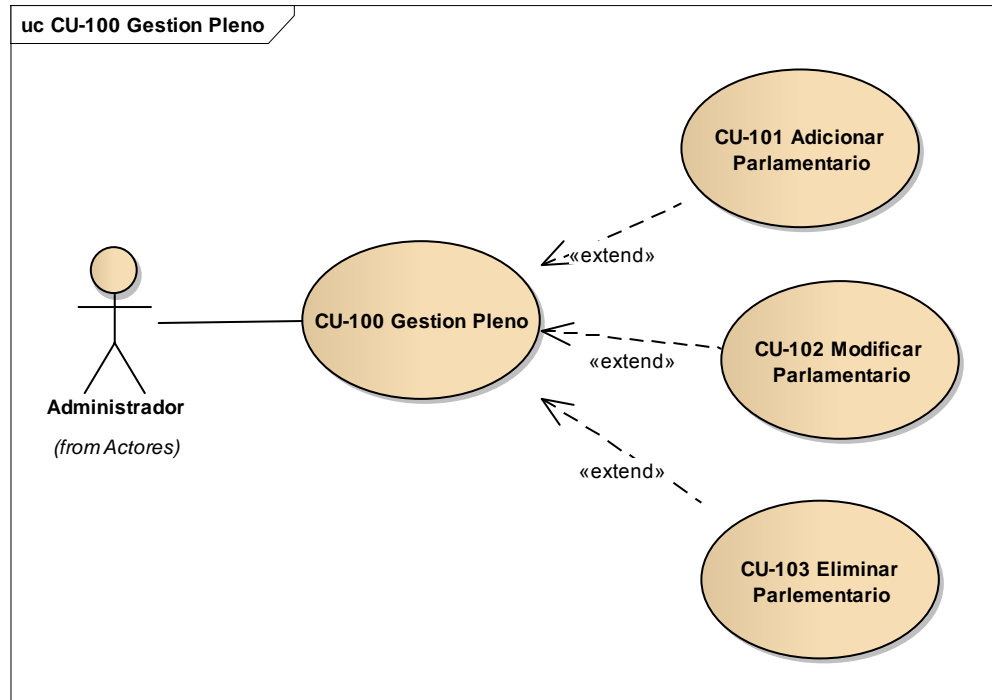


Figura 45 Modelo Caso de Uso Gestión Pleno

2.1.2.2.3.5.4.2.2 Modelo Caso de Uso Gestión Directiva

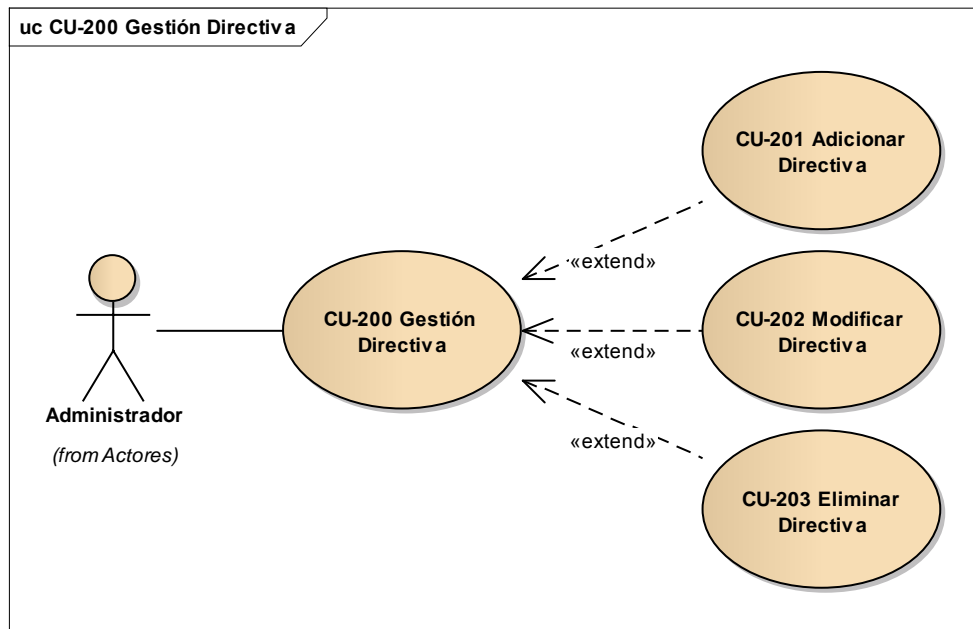


Figura 46 Modelo Caso de Uso Gestión Directiva

2.1.2.2.3.5.4.2.3 Modelo Caso de Uso Gestión Comisiones

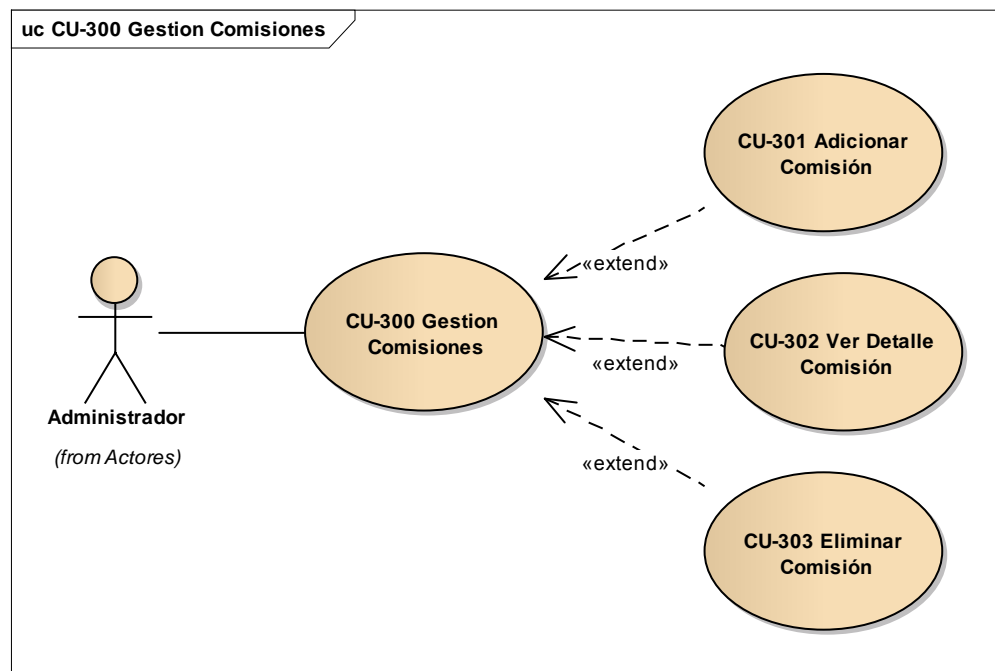


Figura 47 Modelo Caso de Uso Gestión Comisiones

2.1.2.2.3.5.4.2.4 Modelo Caso de Uso Gestión Administrativos

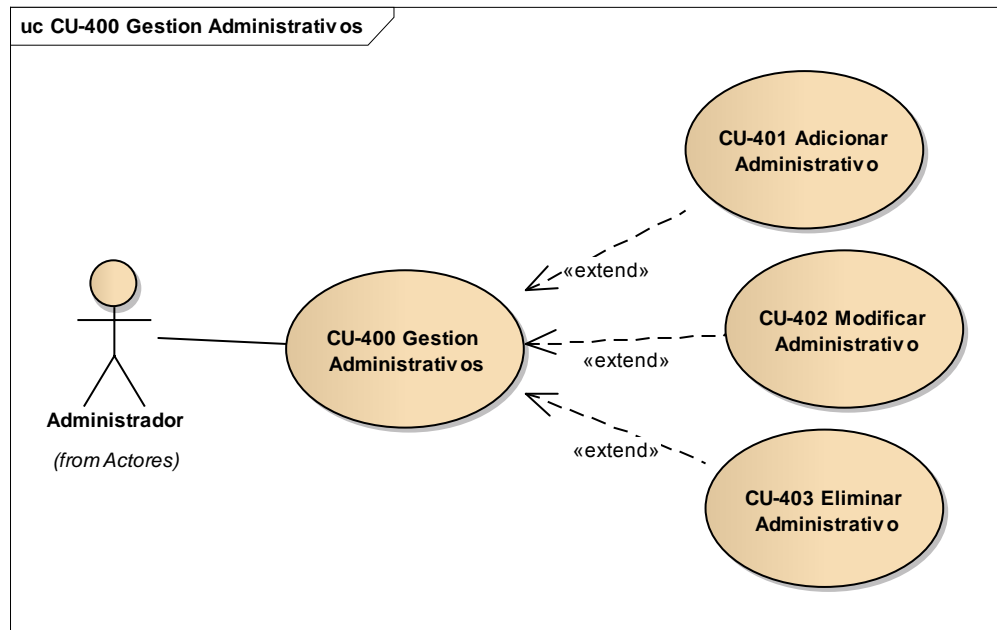


Figura 48 Modelo Caso de Uso Gestión Administrativos

2.1.2.2.3.5.4.2.5 Modelo Caso de Uso Gestión Usuario

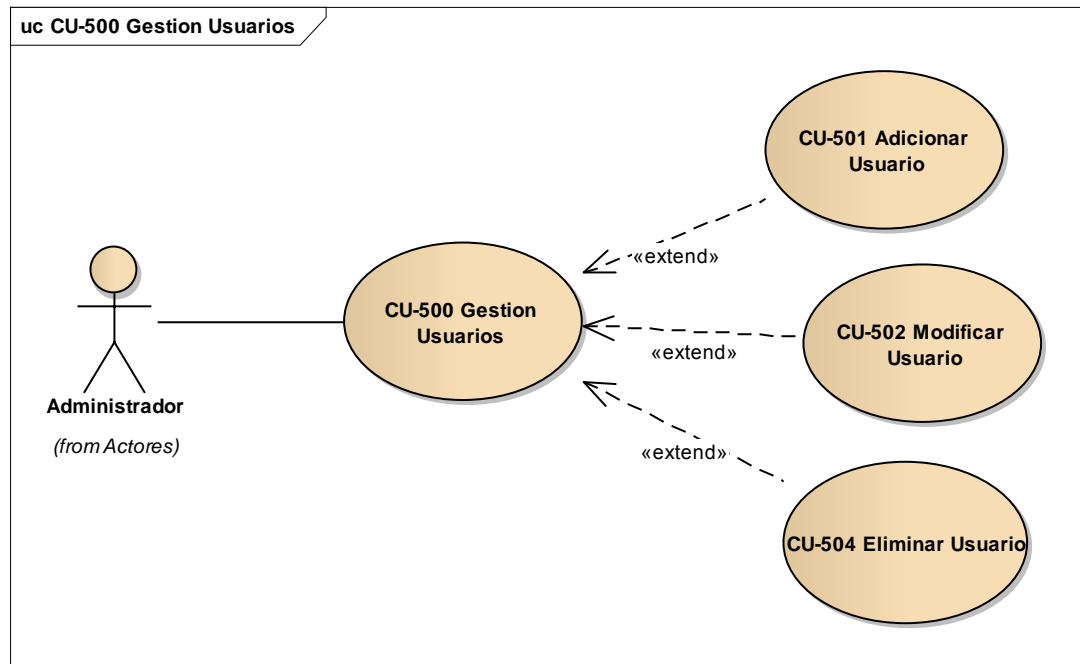


Figura 49 Modelo Caso de Uso Gestión Usuario

2.1.2.2.3.5.4.2.6 Modelo Caso de Uso Gestión Hoja de Proveído

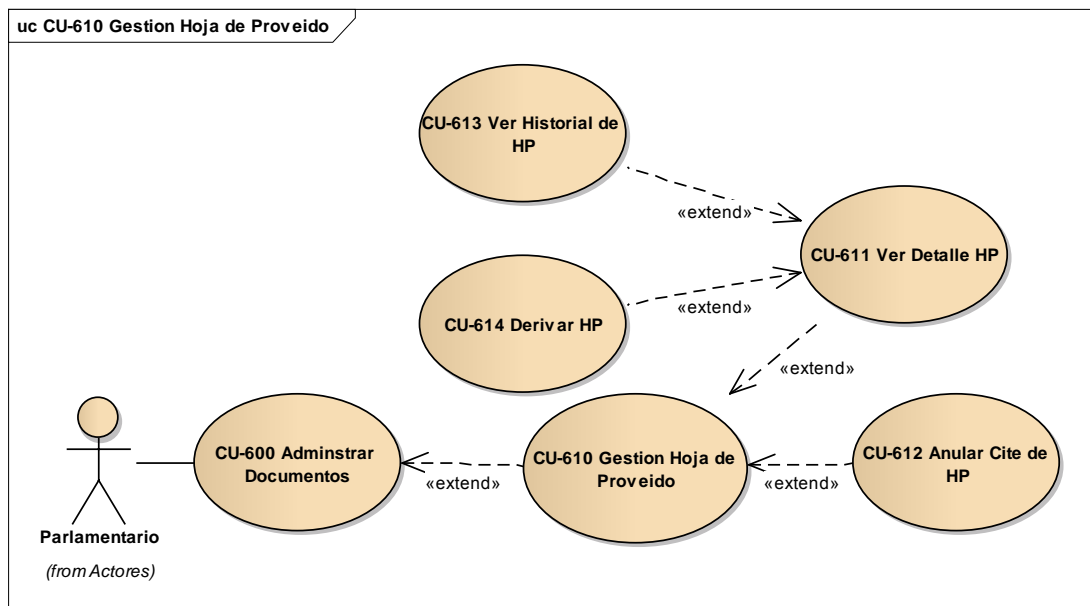


Figura 50 Modelo Caso de Uso Gestión Hoja de Proveído

2.1.2.2.3.5.4.2.7 Modelo Caso de Uso Gestión Hoja de Ruta

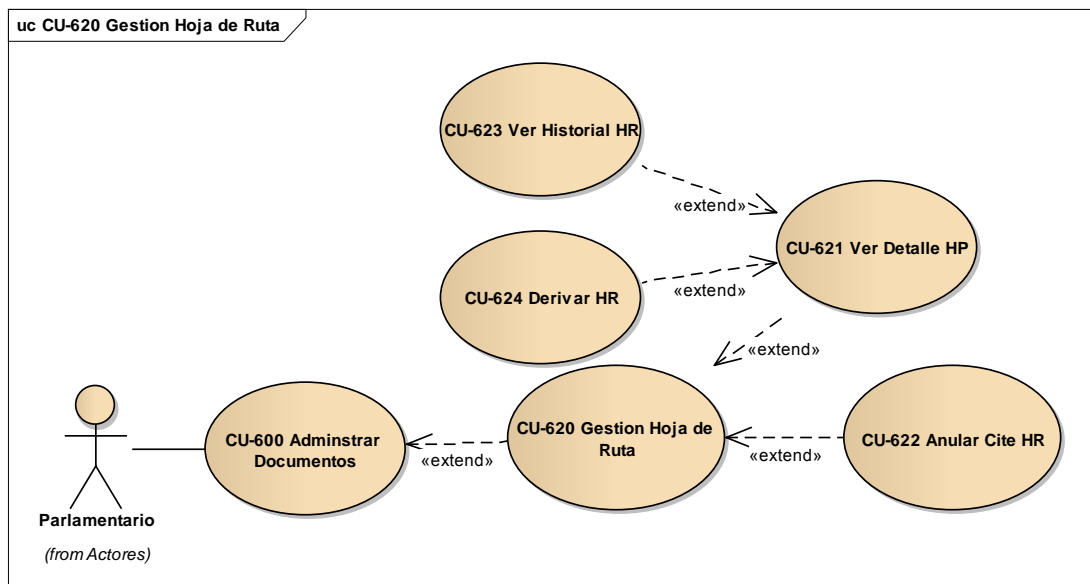


Figura 51 Modelo Caso de Uso Gestión Hoja de Ruta

2.1.2.2.3.5.4.2.8 Modelo Caso de Uso Generar Cite

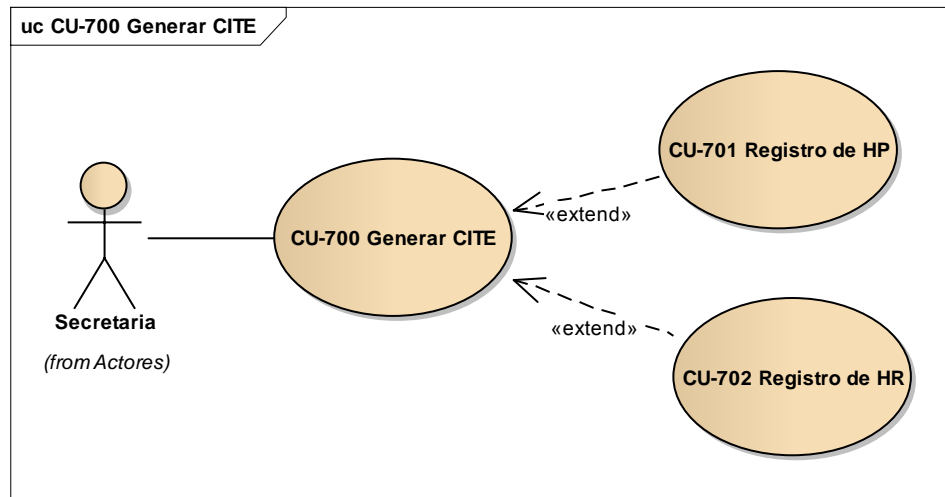


Figura 52 Modelo Caso de Uso Gestión Presupuestos

2.1.2.2.3.5.4.2.9 Modelo Caso de Uso Gestión Reportes HP/HR

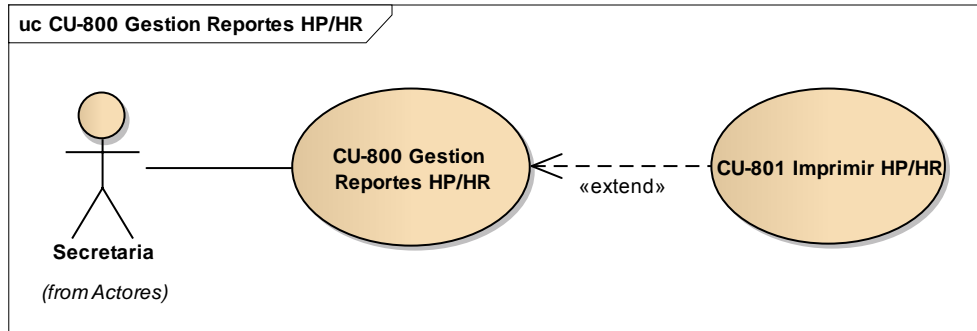


Figura 53 Modelo Caso de Uso Gestión Reportes HP/HR

2.1.2.2.3.5.4.2.10 Modelo Caso de Uso Cambiar Clave

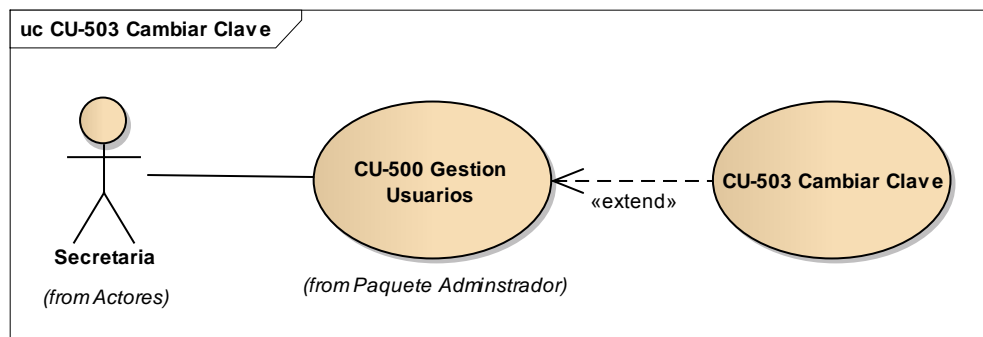


Figura 54 Modelo Caso de Uso Cambiar Clave

2.1.2.2.3.5.4.2.11 Modelo Caso de Uso Buscar Cite

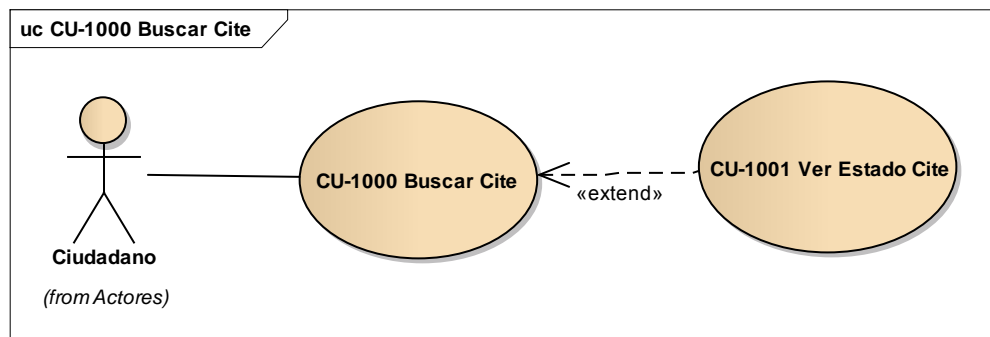


Figura 55 Modelo Caso de Uso Cambiar Clave

2.1.2.2.3.6 Visión

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema.

2.1.2.2.3.6.1 Introducción

2.1.2.2.3.6.1.1 Propósito

El propósito de éste documento es recoger, definir y analizar las necesidades más importantes y las características del desarrollo del Sistema, con el objetivo de llevar a cabo un producto que tome en cuenta las estrategias necesarias para poder brindar información valedera para el Usuario final.

2.1.2.2.3.6.1.2 Alcance

El presente documento se ocupa de reunir todas las necesidades del Usuario, para así poder diseñar un Sistema que satisfaga a las mismas como ser:

- Contar con un medio de publicación de los datos de los documentos tomando en cuenta la seguridad.
- Contar con la información centralizada en una sola organización.
- Brindar una forma virtual de manejar el historial de los documentos.

2.1.2.2.3.6.1.3 Definiciones, Acrónimos y Abreviaciones

- **RUP:** Son las siglas de Rational Unified Process. Se trata de una metodología para describir el proceso de desarrollo de software.
- **UML:** Son las siglas de Lenguaje Unificado de Modelación. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.
- **COCOMO:** Acrónimo en inglés de Modelo Constructivo de Costes. Es un modelo que permite realizar estimaciones y planificaciones de proyectos de sistemas de información.
- **LAN:** Las siglas de Red de Área Local.
- **SQA:** Acrónimo en inglés de Aseguramiento de la Calidad del Software.

2.1.2.2.3.6.1.4 Posicionamiento

2.1.2.2.3.6.1.4.1 Oportunidad de Negocio

La Institución cumple con sus objetivos de servicio a la ciudadanía, al brindar al ciudadano la oportunidad estar conectados con sus representantes (legisladores) y así sentirse representados.

Por lo tanto las ventajas de implementar un sistema informático el cual también registre y lleve un control seguro del registro documental que se

hagan vía Web nos asegurarán la integridad, veracidad y la organización de nuestra información.

2.1.2.2.3.6.1.4.2 Sentencia que define el problema

El problema de	Inadecuado proceso de manejo o administración de la información y control de Registro de documentos.
afecta a	Brigada Parlamentaria de “Tarija”.
El impacto asociado es	Genera retardos y aglomeración de folder con registros.
Una solución adecuada sería	Diseñar un Sistema informático destinado al Control de los documentos.

Tabla 12 Sentencia que define el problema

2.1.2.2.3.6.1.4.3 Sentencia que define la posición del producto

para	Brigada Parlamentaria de “Tarija”.
quienes	Directiva, Pleno y población en general de Tarija
El nombre del producto	“Brigada Parlamentaria Virtual”
que	Gestiona oportunamente y efectivamente la información de los registros de los documentos.
Nuestro producto	Mejorar y automatizar el registro de los documentos mediante una interfaz sencilla y amigable.

Tabla 13 Sentencia que define la posición del producto

2.1.2.2.3.6.1.5 Descripción de los Stakeholders (Participantes del proyecto) y Usuarios

Para proveer de una forma efectiva productos y servicios que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del proceso de modelado de requerimientos. También es necesario identificar a los usuarios del sistema y asegurarse de que el conjunto de participantes en el proyecto los representa adecuadamente. Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los problemas más importantes que éstos perciben para enfocar la solución propuesta hacia ellos. No describe sus requisitos específicos ya que éstos se capturan mediante otro artefacto. En lugar de esto proporciona la justificación de por qué estos requisitos son necesarios.

2.1.2.2.3.6.1.5.1 Perfil de los Stakeholders

2.1.2.2.3.6.1.5.2 Universidad Autónoma Juan Misael Saracho

Representante	Universidad Autónoma Juan Misael Saracho.
Descripción	Entidad que asesora el proyecto.
Tipo	Proveedor.
Responsabilidades	Seguimiento del desarrollo del proyecto. Aprueba requisitos y funcionalidades.
Grado de participación	Velar que el proyecto sea ejecutado.

Tabla 14 Perfil Stakeholder: Universidad Autónoma Juan Misael Saracho

2.1.2.2.3.6.1.5.3 Brigada Parlamentaria de TARIJA

Representante	Jacqueline Mendieta
Descripción	Presidenta de la Brigada Parlamentaria de Tarija
Tipo	Parlamentaria
Responsabilidades	Administrador del Sistema SURD
Grado de participación	Velar por que se cumplan requisitos y funcionalidades, y que el proyecto sea ejecutado.

Tabla 15 Perfil Stakeholder: Brigada Parlamentaria de Tarija

2.1.2.2.3.6.1.5.4 Docente

Representante	Ing. Silvana Paz
Descripción	Ente regulador universitario.
Tipo	Guía de desarrollo.
Responsabilidades	<p>Realiza un control paso a paso del desarrollo del proyecto.</p> <p>Aprueba los distintos puntos tratados en el proyecto.</p> <p>Analiza los distintos documentos presentados acerca del producto.</p>
Grado de participación	Guiar y evaluar el análisis, diseño y desarrollo del sistema.

Tabla 16 Perfil Stakeholder: Docente

2.1.2.2.3.6.1.6 Perfil del Usuario

2.1.2.2.3.6.1.6.1 Administrador

Representante	Administrador
Descripción	Administrador de los recursos del sistema.
Tipo	Usuario master.
Responsabilidades	Responsable del control de permisos y acceso, usuarios del sistema; además es el responsable de administrar todos los componentes de la clínica
Grado de participación	Usuario principal del Sistema

Tabla 17 Perfil Stakeholder: Administrador

2.1.2.2.3.6.1.6.2 Parlamentario

Representante	Parlamentario
Descripción	Usuario con ciertos atributos- permisos de ejecución de la aplicación.
Tipo	Usuario principal.
Responsabilidades	Es el encargado del registro y actor principal en el control de los documentos
Grado de participación	Usar el Sistema, buscar debilidades del mismo.

Tabla 18 Perfil Stakeholder: Parlamentario

2.1.2.2.3.6.1.6.3 Ciudadano

Representante	Ciudadano
Descripción	Usuario que podrá ver el reporte actual de documentos registrados
Tipo	Usuario regular
Responsabilidades	Es el usuario que interactúa directamente con la página Web
Grado de participación	Ver reportes, proponer mejoras de la página web.

Tabla 19 Perfil Stakeholder: Ciudadano

2.1.2.2.3.6.2 Descripción Global del Proyecto

Debido a la creciente demanda de las nuevas tecnologías de información y comunicación y el impacto sociocultural que causan en nuestro entorno, elaboramos el proyecto “SURD”.

Los beneficios de la utilización de nuevas tecnologías de comunicación como ser el Internet nos aseguran la integridad, veracidad y la organización de nuestra información.

2.1.2.2.3.7 Especificaciones de Casos de Usos

Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Máquina de Estados.

2.1.2.2.3.8 Introducción

Las Especificaciones de los casos de Uso es una descripción detallada de los casos de uso del sistema.

2.1.2.2.3.9 Propósito

Interpretar y describir los casos de uso.

2.1.2.2.3.10 Alcance

Describe los procesos internos de los casos de uso.

Detalla los flujos de los casos de uso según lo establecido por la organización.

2.1.2.2.3.11 Especificación de los Casos de Uso

2.1.2.2.3.11.1 Especificación de CU-001 Iniciar Sesión

Caso de uso:	Ingresar al sistema.
Descripción: Permite ingresar al sistema, este caso tiene como función controlar el acceso y al mismo tiempo recuperar los permisos correspondientes al momento que el usuario introduzca su username y password en el sistema.	
Actores: Usuario.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El actor introduce datos de username y password en el sistema. 2. El actor presiona botón iniciar Sesión. 3. El sistema valida sus datos. 4. El actor espera su validación. 5. Si los datos son correctos muestra	Flujo Alternativo: 1. Si el actor no llena el formulario de acceso, se muestra un mensaje donde se informa que los campos son requeridos. 2. Si el usuario no existe se muestra mensaje de error y vuelve a mostrar la pantalla principal.

la Pantalla Principal de Administración.	
Pos condiciones: Ninguno	

Tabla 20 Especificación de CU-001 Iniciar Sesión

2.1.2.2.3.11.2 Especificación de Caso de Uso del Administrador

2.1.2.2.3.11.2.1 Especificación CU-100 Gestión Pleno

Caso de uso:	Gestión Pleno
Descripción: Permite Listar, Adicionar, Modificar, Eliminar al Parlamentario.	
Actores: Administrador.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El sistema muestra Listado de los Parlamentarios registrados en el sistema. 2. El actor realiza la Búsqueda de los Parlamentarios en el Listado. 3. El sistema muestra los Parlamentarios que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona la opción que requiere.	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno	

Tabla 21 Especificación CU-100 Gestión Parlamentario

2.1.2.2.3.11.2.2 Especificación CU-101 Adicionar Parlamentario

Caso de uso:	Adicionar Parlamentario		
Descripción: Permite registrar un nuevo Parlamentario en el sistema.			
Actores: Administrador.			
Precondiciones: Actor debe haber iniciado sesión			
Flujo Normal: 1. El actor presiona botón Adicionar nuevo Parlamentario. 2. El sistema le muestra formulario. 3. El actor introduce datos del Parlamentario en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario y muestra mensaje: “Registro Correcto” 7. El actor presiona botón Cancelar y vuelve a la pantalla Gestión Pleno		Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Parlamentario. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.	
Pos condiciones: Ninguno			

Tabla 22 Especificación CU-101 Adicionar Parlamentario

2.1.2.2.3.11.2.3 Especificación CU-102 Modificar Parlamentario

Caso de uso:	Modificar Parlamentario
Descripción: Permite modificar datos de un Parlamentario en el sistema.	

Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión, Estar registrado en el sistema el Parlamentario	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Editar. 2. El sistema le muestra formulario con datos del Parlamentario. 3. El actor realiza los cambios de datos del parlamentario en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema actualiza los datos del formulario. 7. El sistema regresa a la pantalla de Gestión Pleno. 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Pleno. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.
Pos condiciones: Se Modificó Satisfactoriamente	

Tabla 23 Especificación CU-102 Modificar Parlamentario

2.1.2.2.3.11.2.4 Especificación CU-103 Eliminar Parlamentario

Caso de uso:	Eliminar Parlamentario
Descripción: Permite eliminar un Parlamentario registrado en el sistema.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión.	
Flujo Normal:	Flujo Alternativo:

<ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación lógica (Baja) 4. El sistema regresa a la pantalla de Gestión Pleno. 	<ol style="list-style-type: none"> 1. Por no el sistema no realizará ninguna acción.
Pos condiciones: Se eliminó Satisfactoriamente	

Tabla 24 Especificación CU-103 Eliminar Parlamentario

2.1.2.2.3.11.2.5 Especificación CU-200 Gestión Directiva

Caso de uso:	Gestión Directiva		
Descripción: Permite Listar, Adicionar, Modificar y Eliminar.			
Actores: Administrador.			
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.			
Flujo Normal: 1. El sistema muestra Listado de los Miembros de la Directiva registrados en el sistema. 2. El actor realiza la Búsqueda de los Miembros de la Directiva en el Listado. 3. El sistema muestra los miembros de la directiva que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona la opción que requiere.		Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.	
Pues condiciones: Ninguno			

Tabla 25 Especificación CU-200 Gestión Directiva

2.1.2.2.3.11.2.6 Especificación CU-201 Adicionar Directiva

Caso de uso:	Adicionar Directiva		
Descripción: Permite registrar un nuevo miembro de la Directiva en el sistema.			
Actores: Administrador.			
Precondiciones: Actor debe haber iniciado sesión			
Flujo Normal: 1. El actor presiona botón Adicionar Nuevo miembro de la Directiva. 2. El sistema le muestra formulario. 3. El actor introduce datos del permiso en el formulario. 4. Selecciona botón Aceptar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario y muestra mensaje: “Registro Correcto” 7. El actor presiona botón Cancelar y vuelve a la pantalla Gestión Permisos		Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Directiva. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.	
Pos condiciones: Ninguno			

Tabla 26 Especificación CU-201 Adicionar Directiva

2.1.2.2.3.11.2.7 Especificación CU-202 Modificar Directiva

Caso de uso:	Modificar Permiso
Descripción:	

Permite modificar datos de un miembro de la Directiva.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión, Estar registrado en el sistema la Directiva	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Editar. 2. El sistema le muestra formulario con datos de miembro de la Directiva. 3. El actor realiza los cambios de datos en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema actualiza los datos del formulario. 7. El sistema regresa a la pantalla de Gestión Directiva. 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Directiva.
Pos condiciones: Se Modificó Satisfactoriamente	

Tabla 27 Especificación CU-202 Modificar Directiva

2.1.2.2.3.11.2.8 Especificación CU-203 Eliminar Directiva

Caso de uso:	Eliminar Directiva
Descripción: Permite eliminar un miembro de la Directiva registrado en el sistema.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión.	

Flujo Normal: 1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación lógica (Baja). 4. El sistema regresa a la pantalla de Gestión Directiva.	Flujo Alternativo: 1. Por no el sistema no realizará ninguna acción.
Pos condiciones: Se eliminó Satisfactoriamente	

Tabla 28 Especificación CU-203 Eliminar Directiva

2.1.2.2.3.11.2.9 Especificación CU-300 Gestión Comisiones

Caso de uso:	Gestión Comisiones
Descripción: Permite Listar, Adicionar, Ver Detalle y Eliminar Comisión.	
Actores: Administrador.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El sistema muestra Listado de Comisiones registrados en el sistema. 2. El actor realiza la Búsqueda de las Comisiones en el Listado. 3. El sistema muestra los Tratamientos que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona la opción que requiere.	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno	

Tabla 29 Especificación CU-300 Gestión Comisiones

2.1.2.2.3.11.2.10 Especificación CU-301 Adicionar Comisión

Caso de uso:	Adicionar Comisión		
Descripción: Permite registrar una nueva comisión en el sistema.			
Actores: Administrador.			
Precondiciones: Actor debe haber iniciado sesión			
Flujo Normal: 1. El actor presiona botón Adicionar Nuevo. 2. El sistema le muestra formulario. 3. El actor introduce datos de la Comisión. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario y muestra mensaje: “Se adiciono exitosamente”. 7. El actor presiona botón Cancelar y vuelve a la pantalla Gestión Comisiones		Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Comisiones. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.	
Pos condiciones: Ninguno			

Tabla 30 Especificación CU-301 Adicionar Comisión

2.1.2.2.3.11.2.11 Especificación CU-302 Ver Detalle Comisión

Caso de uso:	Ver Detalle Comisión
Descripción:	

Permite Ver los datos de una comisión en el sistema.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión, Estar registrado en el sistema el tratamiento	
Flujo Normal: 1. El actor presiona en el menú la opción Ver Detalle. 2. El sistema le muestra formulario con datos de la comisión. 3. Selecciona botón Cerrar. 4. El sistema regresa a la pantalla de Gestión Comisiones.	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Se Modificó Satisfactoriamente	

Tabla 31 Especificación CU-302 Ver Detalle Comisión

2.1.2.2.3.11.2.12 Especificación CU-303 Eliminar Comisión

Caso de uso:	Eliminar Comisión
Descripción: Permite eliminar una comisión registrada en el sistema.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión.	
Flujo Normal: 1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la	Flujo Alternativo: 1. Por no el sistema no realizará ninguna acción.

eliminación lógica (Baja).	
4. El sistema regresa a la pantalla de Gestión Comisiones.	
Pos condiciones: Se eliminó Satisfactoriamente	

Tabla 32 Especificación CU-303 Eliminar Comisión

2.1.2.2.3.11.2.13 Especificación CU-400 Gestión Administrativos

Caso de uso:	Gestión administrativos
Descripción: Permite Listar, Adicionar, Modificar, Eliminar.	
Actores: Administrador.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El sistema muestra Listado de los Administrativos registrados en el sistema. 2. El actor realiza la Búsqueda de los Administrativos en el Listado. 3. El sistema muestra los administrativos que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona la opción que requiere.	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno	

Tabla 33 Especificación CU-400 Gestión Administrativos

2.1.2.2.3.11.2.14 Especificación CU-401 Adicionar Administrativo

Caso de uso:	Adicionar administrativo
---------------------	--------------------------

Descripción: Permite registrar un nuevo administrativo en el sistema.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona botón Adicionar nuevo. 2. El sistema le muestra formulario. 3. El actor introduce datos del Administrativo en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario y muestra mensaje: “Se adiciono exitosamente”. 7. El actor presiona botón Cancelar y vuelve a la pantalla Gestión Administrativos 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Administrativos. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.
Pos condiciones: Ninguno	

Tabla 34 Especificación CU-401 Adicionar Área

2.1.2.2.3.11.2.15 Especificación CU-402 Modificar Administrativo

Caso de uso:	Modificar Administrativo
Descripción: Permite modificar datos de un Administrativo en el sistema.	
Actores: Administrador.	

Precondiciones: Actor debe haber iniciado sesión. Estar registrado en el sistema el Administrativo	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Editar. 2. El sistema le muestra formulario con datos del Administrativo. 3. El actor realiza los cambios de datos del Administrativo en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema actualiza los datos del formulario. 7. El sistema regresa a la pantalla de Gestión Administrativo. 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Administrativo. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.
Pos condiciones: Se Modificó Satisfactoriamente	

Tabla 35 Especificación CU-402 Modificar Administrativo

2.1.2.2.3.11.2.16 Especificación CU-403 Eliminar Administrativo

Caso de uso:	Eliminar Administrativo
Descripción: Permite eliminar un Administrativo registrado en el sistema.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Por no el sistema no realizará ninguna acción.

confirmación. 3. Por Si, El sistema realiza la eliminación lógica (Baja). 4. El sistema regresa a la pantalla de Gestión Administrativo.	
Pos condiciones: Se eliminó Satisfactoriamente	

Tabla 36 Especificación CU-403 Eliminar Administrativo

2.1.2.2.3.11.2.17 Especificación CU-500 Gestión Usuarios

Caso de uso:	Gestión Usuario
Descripción: Permite Listar, Adicionar, Modificar, Eliminar.	
Actores: Administrador.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El sistema muestra Listado de los Usuarios registrados en el sistema. 2. El actor realiza la Búsqueda de los Usuarios en el Listado. 3. El sistema muestra los Usuarios que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona la opción que requiere.	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno	

Tabla 37 Especificación CU-500 Gestión Usuarios

2.1.2.2.3.11.2.18 Especificación CU-501 Adicionar Usuario

Caso de uso:	Adicionar Usuario		
Descripción: Permite registrar un nuevo Usuario en el sistema.			
Actores: Administrador.			
Precondiciones: Actor debe haber iniciado sesión			
Flujo Normal: 1. El actor presiona botón Adicionar Nuevo. 2. El sistema le muestra formulario. 3. El actor introduce datos del Usuario en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario y muestra mensaje: “Se adiciono exitosamente”. 7. El actor presiona botón Cancelar y vuelve a la pantalla Gestión Usuarios		Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Usuarios. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.	
Pos condiciones: Ninguno			

Tabla 38 Especificación CU-501 Adicionar Usuario

2.1.2.2.3.11.2.19 Especificación CU-402 Modificar Usuario

Caso de uso:	Modificar Usuario
Descripción:	

Permite modificar datos de un Usuario en el sistema.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión. Estar registrado en el sistema el Usuario	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Editar. 2. El sistema le muestra formulario con datos del Usuario. 3. El actor realiza los cambios de datos del Usuario en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema actualiza los datos del formulario. 7. El sistema regresa a la pantalla de Gestión Usuarios. 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Usuarios. 2. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.
Pos condiciones: Se Modificó Satisfactoriamente	

Tabla 39 Especificación CU-502 Modificar Usuario

2.1.2.2.3.11.2.20 Especificación CU-504 Eliminar Usuario

Caso de uso:	Eliminar Usuario
Descripción: Permite eliminar un Usuario registrado en el sistema.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión.	

Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación lógica (Baja). 4. El sistema regresa a la pantalla de Gestión Usuarios. 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Por no el sistema no realizará ninguna acción.
Pos condiciones: Se eliminó Satisfactoriamente	

Tabla 40 Especificación CU-504 Eliminar Usuario

2.1.2.2.3.11.3 Especificación de Caso de Uso del Parlamentario

2.1.2.2.3.11.3.1 Especificación CU-610 Gestión Hoja de Proveído

Caso de uso:	Gestión Hoja de Proveído
Descripción: Permite Ver Detalle HP, Anular Cite de HP, Ver Historial de HP y Derivar HP	
Actores: Parlamentario.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: <ol style="list-style-type: none"> 1. El sistema muestra Listado de las Hojas de Proveído registrados en el sistema. 2. El actor realiza la Búsqueda de las Hojas de Proveído. 3. El sistema muestra los Listados que cumple(n) con los Criterios de 	Flujo Alternativo: <ol style="list-style-type: none"> 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.

Búsqueda. 4. El actor selecciona la opción que requiere.	
Pos condiciones: Ninguno	

Tabla 41 Especificación CU-610 Gestión Hoja de Proveído

2.1.2.2.3.11.3.2 Especificación CU-611 Ver Detalle de HP

Caso de uso:	Ver Detalle de HP
Descripción: Permite ver el detalle de una Hoja de Proveído registrada en el sistema	
Actores: Parlamentario.	
Precondiciones: Actor debe haber iniciado sesión	
Flujo Normal: 1. El actor presiona botón Ver Detalle. 2. El sistema le muestra formulario con el Detalle del Registro y las Opciones Ver Historial y Derivar. 3. El actor presiona botón Cerrar y vuelve a la pantalla Gestión de Hoja de Proveído	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno	

Tabla 42 Especificación CU-611 Ver Detalle de HP

2.1.2.2.3.11.3.3 Especificación CU-613 Ver Historial de HP

Caso de uso:	Ver Historial de HP
Descripción:	

Permite ver el detalle de Derivaciones del HP registrado en el sistema	
Actores: Parlamentario.	
Precondiciones: Actor debe haber iniciado sesión.	
Flujo Normal: 1. El actor presiona botón Ver Detalle. 2. El sistema le muestra formulario con el Detalle del Registro y las Opciones Ver Historial y Derivar. 3. El actor presiona botón Ver Historial 4. El sistema muestra una tabla con todo el historial de derivaciones registradas del HP. 5. El actor presiona botón Cerrar y vuelve a la pantalla Gestión de Hoja de Proveído	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguna	

Tabla 43 Especificación CU-613 Ver Historial de HP

2.1.2.2.3.11.3.4 Especificación CU-614 Derivar HP

Caso de uso:	Derivar HP
Descripción: Permite realizar el registro de la derivación de un HP	
Actores: Parlamentario.	
Precondiciones: Actor debe haber iniciado sesión, Estar registrado en el sistema el HP	
Flujo Normal: 1. El actor presiona botón Ver	Flujo Alternativo: 1. El actor en el paso 4 después de

<p>Detalle.</p> <ol style="list-style-type: none"> 2. El sistema le muestra formulario con el Detalle del Registro y las Opciones Ver Historial y Derivar. 3. El actor presiona el botón Derivar 4. El sistema le muestra un formulario. 5. El actor introduce los datos de la derivación. 6. Selecciona botón Derivar. 7. El sistema guarda datos del formulario y muestra mensaje: “Se registró exitosamente”. 8. El actor presiona botón Cancelar y vuelve a la pantalla Gestión Hoja de Proveído 	<p>presionar el botón Derivar tiene la opción de Finalizar/Archivar el HP con esto da por finalizado el proceso del HP.</p>
<p>Pos condiciones: Se registró Satisfactoriamente</p>	

Tabla 44 Especificación CU-614 Derivar HP

2.1.2.2.3.11.3.5 Especificación CU-612 Anular Cite de HP

Caso de uso:	Anular Cite de HP		
Descripción: Permite eliminar un Cite de HP en el sistema.			
Actores: Parlamentario.			
Precondiciones: Actor debe haber iniciado sesión.			
Flujo Normal: 1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación lógica (Baja).		Flujo Alternativo: 1. Por no el sistema no realizará ninguna acción.	

4. El sistema regresa a la pantalla de Gestión Hoja de Proveído.	
Pos condiciones: Se eliminó Satisfactoriamente	

Tabla 45 Especificación CU-612 Anular Cite de HP

2.1.2.2.3.11.3.6 Especificación CU-620 Gestión Hoja de Ruta

Caso de uso:	Gestión Hoja de Ruta		
Descripción: Permite Ver Detalle HR, Anular Cite de HR, Ver Historial de HR y Derivar HR			
Actores: Parlamentario.			
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.			
Flujo Normal: 1. El sistema muestra Listado de las Hojas de Ruta registrados en el sistema. 2. El actor realiza la Búsqueda de las Hojas de Ruta. 3. El sistema muestra los Listados que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona la opción que requiere.		Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.	
Pos condiciones: Ninguno			

Tabla 46 Especificación CU-620 Gestión Hoja de Ruta

2.1.2.2.3.11.3.7 Especificación CU-621 Ver Detalle de HR

Caso de uso:	Ver Detalle de HR
---------------------	-------------------

Descripción: Permite ver el detalle de una Hoja de Ruta registrada en el sistema	
Actores: Parlamentario.	
Precondiciones: Actor debe haber iniciado sesión	
Flujo Normal: 1. El actor presiona botón Ver Detalle. 2. El sistema le muestra formulario con el Detalle del Registro y las Opciones Ver Historial y Derivar. 3. El actor presiona botón Cerrar y vuelve a la pantalla Gestión de Hoja de Ruta	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno	

Tabla 47 Especificación CU-611 Ver Detalle de HP

2.1.2.2.3.11.3.8 Especificación CU-623 Ver Historial de HR

Caso de uso:	Ver Historial de HR
Descripción: Permite ver el detalle de Derivaciones del HR registrado en el sistema	
Actores: Parlamentario.	
Precondiciones: Actor debe haber iniciado sesión.	
Flujo Normal: 1. El actor presiona botón Ver Detalle. 2. El sistema le muestra formulario	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.

con el Detalle del Registro y las Opciones Ver Historial y Derivar. 3. El actor presiona botón Ver Historial 4. El sistema muestra una tabla con todo el historial de derivaciones registradas del HR. 5. El actor presiona botón Cerrar y vuelve a la pantalla Gestión de Hoja de Ruta	
Pos condiciones: Ninguna	

Tabla 48 Especificación CU-613 Ver Historial de HP

2.1.2.2.3.11.3.9 Especificación CU-624 Derivar HR

Caso de uso:	Derivar HR
Descripción: Permite realizar el registro de la derivación de un HR	
Actores: Parlamentario.	
Precondiciones: Actor debe haber iniciado sesión, Estar registrado en el sistema el HR	
Flujo Normal: 1. El actor presiona botón Ver Detalle. 2. El sistema le muestra formulario con el Detalle del Registro y las Opciones Ver Historial y Derivar. 3. El actor presiona el botón Derivar 4. El sistema le muestra un formulario. 5. El actor introduce los datos de la derivación. 6. Selecciona botón Derivar. 7. El sistema guarda datos del formulario y muestra mensaje: “Se registró exitosamente”.	Flujo Alternativo: 1. El actor en el paso 4 después de presionar el botón Derivar tiene la opción de Finalizar/Archivar el HP con esto da por finalizado el proceso del HR.

8. El actor presiona botón Cancelar y vuelve a la pantalla Gestión Hoja de Ruta	
Pos condiciones: Se registró Satisfactoriamente	

Tabla 49 Especificación CU-624 Derivar HR

2.1.2.2.3.11.3.10 Especificación CU-622 Anular Cite de HR

Caso de uso:	Anular Cite de HR
Descripción: Permite eliminar un Cite de HR en el sistema.	
Actores: Parlamentario.	
Precondiciones: Actor debe haber iniciado sesión.	
Flujo Normal: 1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación lógica (Baja). 4. El sistema regresa a la pantalla de Gestión Hoja de Ruta.	Flujo Alternativo: 1. Por no el sistema no realizará ninguna acción.
Pos condiciones: Se eliminó Satisfactoriamente	

Tabla 50 Especificación CU-612 Anular Cite de HP

2.1.2.2.3.11.4 Especificación de Caso de Uso de la Secretaria

2.1.2.2.3.11.4.1 Especificación CU-700 Generar CITE

Caso de uso:	Generar Cite
---------------------	--------------

Descripción: Permite Registrar un Cite de Hoja de Proveído y Cite de Hoja de Ruta	
Actores: Secretaria.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El sistema muestra Un formulario Para generar un CITE con la opción de ser un HR o HP 2. El actor selecciona la opción que requiere.	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno	

Tabla 51 Especificación CU-700 Genera CITE

2.1.2.2.3.11.4.2 Especificación CU-701 Registro de HP

Caso de uso:	Registro de HP
Descripción: Permite registrar un nuevo Cite de Hoja de Provisto.	
Actores: Secretaria.	
Precondiciones: Actor debe haber iniciado sesión	
Flujo Normal: 1. El sistema muestra Un formulario Para generar un CITE con la opción de ser un HR o HP. 2. El actor selecciona el botón Provisto 3. El actor introduce datos del registro en el formulario.	Flujo Alternativo: 1. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.

4. Selecciona botón Registrar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario y muestra mensaje: “Se adiciono exitosamente”. 7. El actor presiona botón Limpiar y se muestra el Formulario de registro vacío.	
Pos condiciones: Ninguno	

Tabla 52 Especificación CU-701 Registro de HP

2.1.2.2.3.11.4.3 Especificación CU-702 Registro de HR

Caso de uso:	Registro de HR		
Descripción: Permite registrar un nuevo Cite de Hoja de Ruta.			
Actores: Secretaria.			
Precondiciones: Actor debe haber iniciado sesión			
Flujo Normal: 1. El sistema muestra Un formulario Para generar un CITE con la opción de ser un HR o HP. 2. El actor selecciona el botón Ruta 3. El actor introduce datos del registro en el formulario. 4. Selecciona botón Registrar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario y muestra mensaje: “Se adiciono exitosamente”. 7. El actor presiona botón Limpiar y se muestra el Formulario de registro vacío.		Flujo Alternativo: 1. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.	

Pos condiciones: Ninguno

Tabla 53 Especificación CU-702 Registro de HR

2.1.2.2.3.11.4.4 Especificación CU-800 Gestión Reportes HP/HR

Caso de uso:	Gestión Reportes HP/HR
Descripción: Permite Listar Todos Cites Activos Registrados	
Actores: Secretaria.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: <ol style="list-style-type: none"> 1. El sistema muestra Listado de las Hojas de Provisto y Hojas de Ruta registradas en el sistema. 2. El actor realiza la Búsqueda de las Hojas de Ruta u Hojas de Provisto. 3. El sistema muestra los Listados que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona la opción que requiere. 	Flujo Alternativo: <ol style="list-style-type: none"> 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno	

Tabla 54 Especificación CU-800 Gestión Reportes HP/HR

2.1.2.2.3.11.4.5 Especificación CU-801 Imprimir HP/HR

Caso de uso:	Imprimir HP/HR
Descripción: Permite Imprimir un reporte actual del CITE de Hoja de Provisto u Hoja de Ruta	

Actores: Secretaria.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El actor presiona botón Imprimir. 2. El sistema le muestra un formulario con todos los datos actuales del Cite de HP o HR no editables. 3. El actor presiona puede imprimir el documento y/o guardarlo	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno	

Tabla 55 Especificación CU-801 Imprimir HP/HR

2.1.2.2.3.11.4.6 Especificación CU-503 Cambiar Clave

Caso de uso:	Cambiar Clave
Descripción: Permite modificar datos de un Usuario en el sistema.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión. Estar registrado en el sistema el Usuario	
Flujo Normal: 8. El actor presiona en el menú la opción Editar. 9. El sistema le muestra formulario con datos del Usuario. 10. El actor realiza los cambios de datos del Usuario en el formulario. 11. Selecciona botón Guardar. 12. Se valida datos del formulario que son obligatorios estén introducidos.	Flujo Alternativo: 3. Selecciona botón Cancelar, se regresa a la pantalla Gestión Usuarios. 4. Se muestra un mensaje donde se informa que los campos son requeridos y se vuelve al paso 3 para que el actor pueda ingresar la información que falta.

13. El sistema actualiza los datos del formulario.	
14. El sistema regresa a la pantalla de Gestión Usuarios.	
Pos condiciones: Se Modificó Satisfactoriamente	

Tabla 56 Especificación CU-502 Modificar Usuario

2.1.2.2.3.11.5 Especificación de Caso de Uso del Ciudadano

2.1.2.2.3.11.5.1 Especificación CU-1000 Buscar Cite

Caso de uso:	Buscar Cite
Descripción: Permite buscar un Cite de Hoja de Ruta	
Actores: Ciudadano.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El sistema muestra un formulario donde se puede ingresar un Cite de Hoja de Ruta para ver su estado actual. 2. El actor selecciona la opción que requiere.	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno	

Tabla 57 Especificación CU-1000 Buscar Cite

2.1.2.2.3.11.5.2 Especificación CU-1001 Ver Estado

Caso de uso:	Ver Estado
Descripción: Permite Ver el Estado actual de un Cite de Hoja de Ruta	

Actores: Ciudadano.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El actor ingresa los datos de un Cite de HP valido. 2. El actor presiona el botón Ver Estado. 3. El sistema muestra el Detalle del Registro y también la opción de ver el Historial de derivaciones	Flujo Alternativo: 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno	

Tabla 58 Especificación CU-1000 Buscar Cite

2.1.2.2.3.12 Especificaciones Adicionales

Este documento capturará todos los requisitos que no han sido incluidos como parte de los casos de uso y se refieren requisitos no-funcionales globales. Dichos requisitos incluyen: requisitos legales o normas, aplicación de estándares, requisitos de calidad del producto, tales como: confiabilidad, desempeño, etc., u otros requisitos de ambiente, tales como: sistema operativo, requisitos de compatibilidad, etc.

2.1.2.2.3.13 Prototipos de Interfaces de Usuario

2.1.2.2.3.13.1 Introducción

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden

de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados. Asimismo, este artefacto, será desechado en la fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

2.1.2.2.3.13.2 Propósito

Presentar los prototipos de pantallas para que el usuario tenga una idea de la interfaz que se presentaran en el Sistema.

2.1.2.2.3.13.3 Alcance

Mostrar los Prototipos de Pantallas, sujeto a modificaciones a lo largo del desarrollo del Sistema.

2.1.2.2.3.13.4 Prototipo de Pantallas

2.1.2.2.3.13.4.1 Pantalla Iniciar Sesión

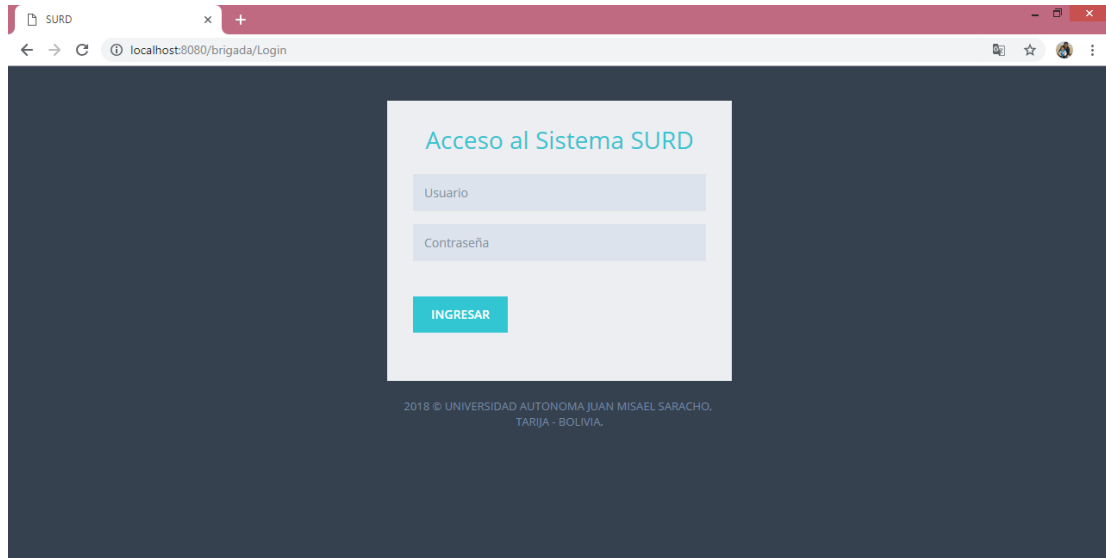


Figura 56 Pantalla Iniciar Sesión

2.1.2.2.3.13.4.1.1 Pantalla Principal Administrador

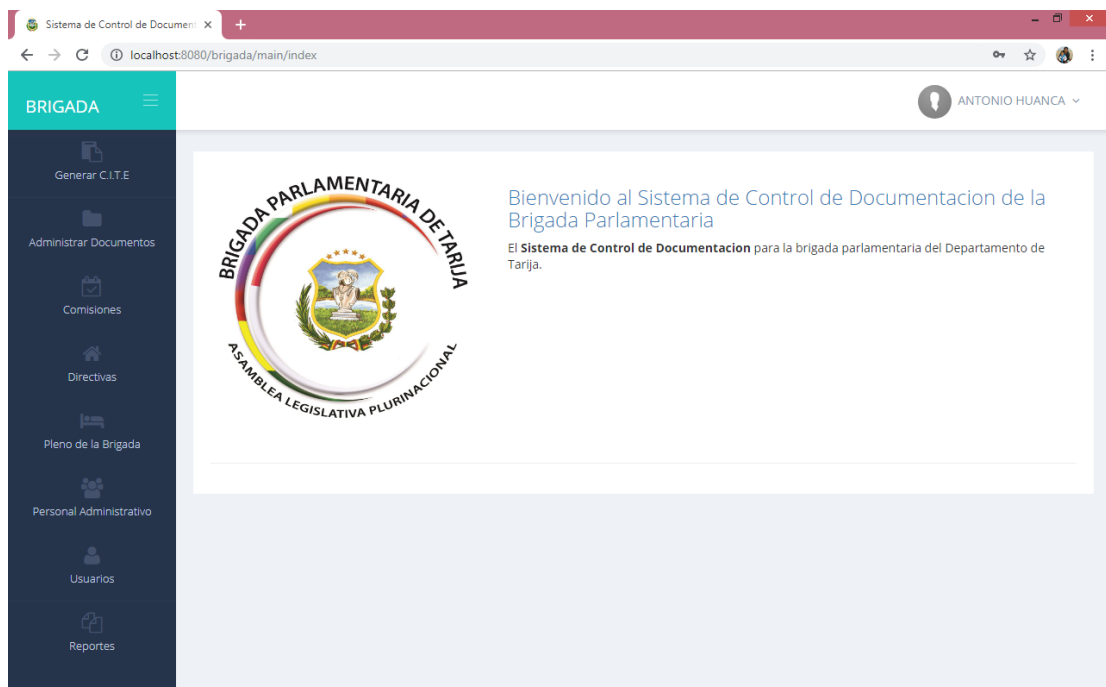


Figura 57 Pantalla Principal Administrador

2.1.2.2.3.13.4.1.2 Pantalla Gestión Pleno

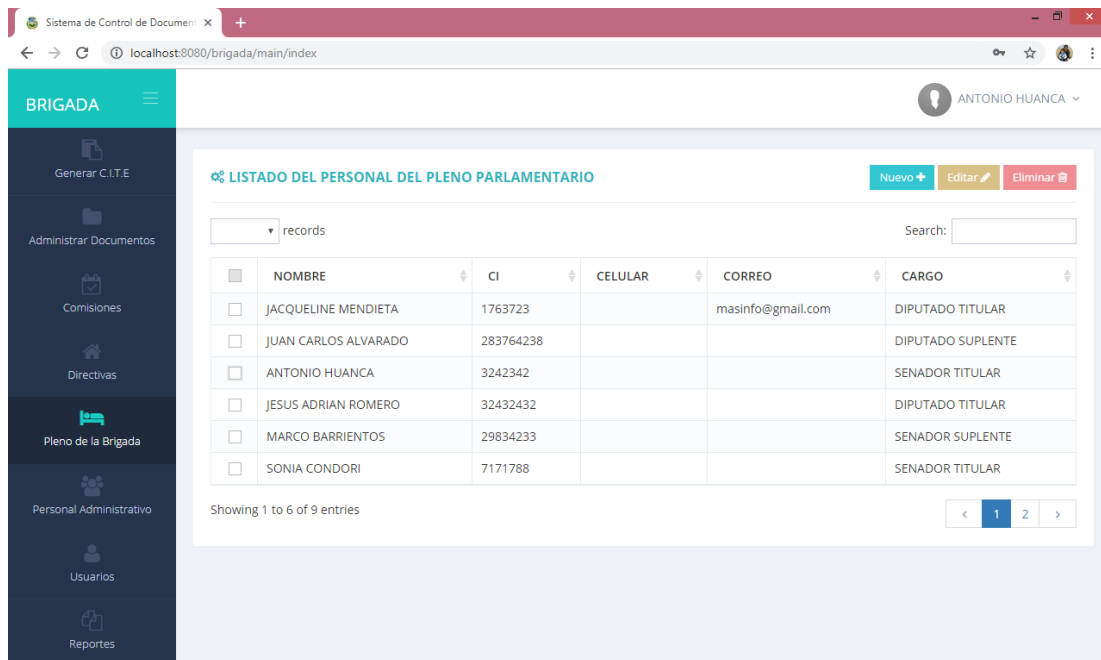


Figura 58 Pantalla Gestión Pleno

2.1.2.2.3.13.4.1.3 Pantalla Adicionar Parlamentario

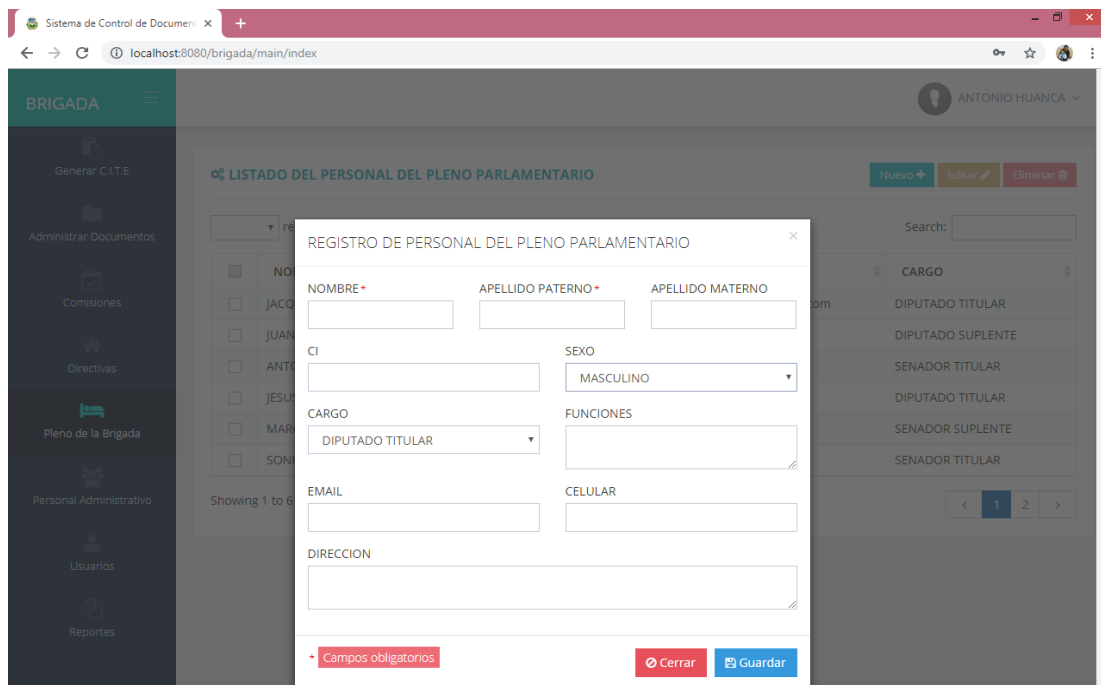


Figura 59 Pantalla Adicionar Parlamentario

2.1.2.2.3.13.4.1.4 Pantalla Modificar Parlamentario

MODIFICAR DATOS DE PERSONAL DEL PLENO PARLAMENTARIO

Nombres(s) * APELLIDO PATERNO * APELLIDO MATERNO

JACQUELINE MENDEIETA

CI SEXO

1763723 MASCULINO

CARGO FUNCIONES

DIPUTADO TITULAR

EMAIL CELULAR

masinfo@gmail.com

DIRECCION

valparaiso calle 5

Campos obligatorios

Cerrar Guardar

Figura 60 Pantalla Modificar Parlamentario

2.1.2.2.3.13.4.1.5 Pantalla Eliminar Parlamentario

BORRAR datos?

¿Está seguro de eliminar el dato seleccionado?

NO! SI!

LISTADO DEL PERSONAL DEL PLENO PARLAMENTARIO

	NOMBRE	CI	CELULAR	CORREO	CARGO
<input checked="" type="checkbox"/>	JACQUELINE MENDEIETA	1763723		masinfo@gmail.com	DIPUTADO TITULAR
<input type="checkbox"/>	JUAN CARLOS ALVARADO	283764238			DIPUTADO SUPLENTE
<input type="checkbox"/>	ANTONIO HUANCA	3242342			SENADOR TITULAR
<input type="checkbox"/>	JESU				DIPUTADO TITULAR
<input type="checkbox"/>	MAR				SENADOR SUPLENTE
<input type="checkbox"/>	SON				SENADOR TITULAR

Figura 61 Pantalla Eliminar Parlamentario

2.1.2.2.3.13.4.1.6 Pantalla Gestión Directiva

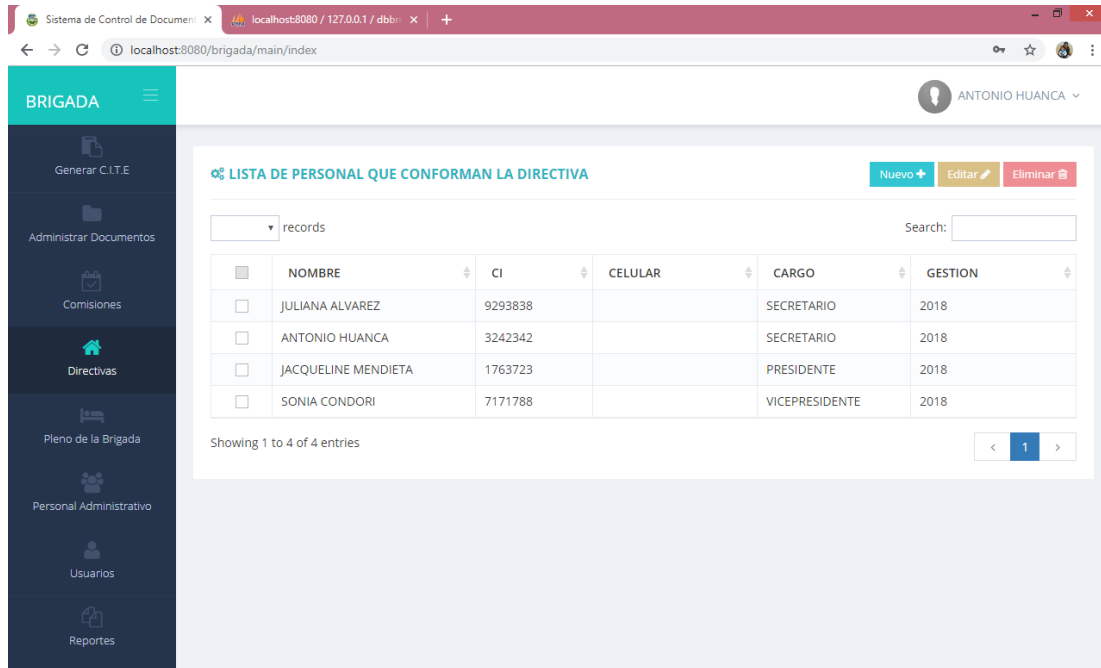


Figura 62 Pantalla Gestión Directiva

2.1.2.2.3.13.4.1.7 Pantalla Adicionar Directiva

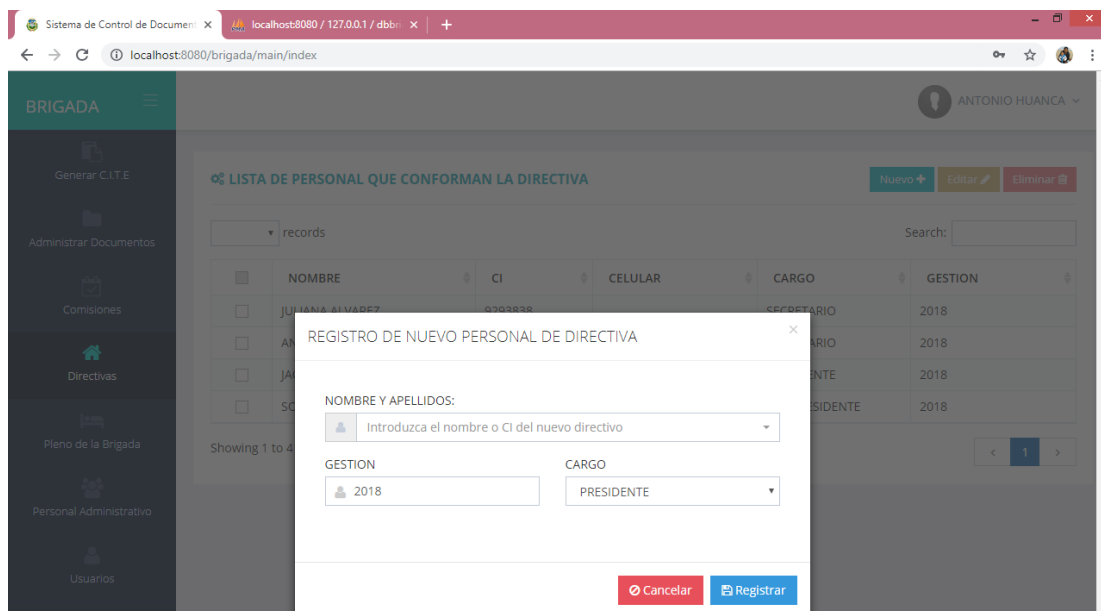


Figura 63 Adicionar Directiva

2.1.2.2.3.13.4.1.8 Pantalla Modificar Directiva

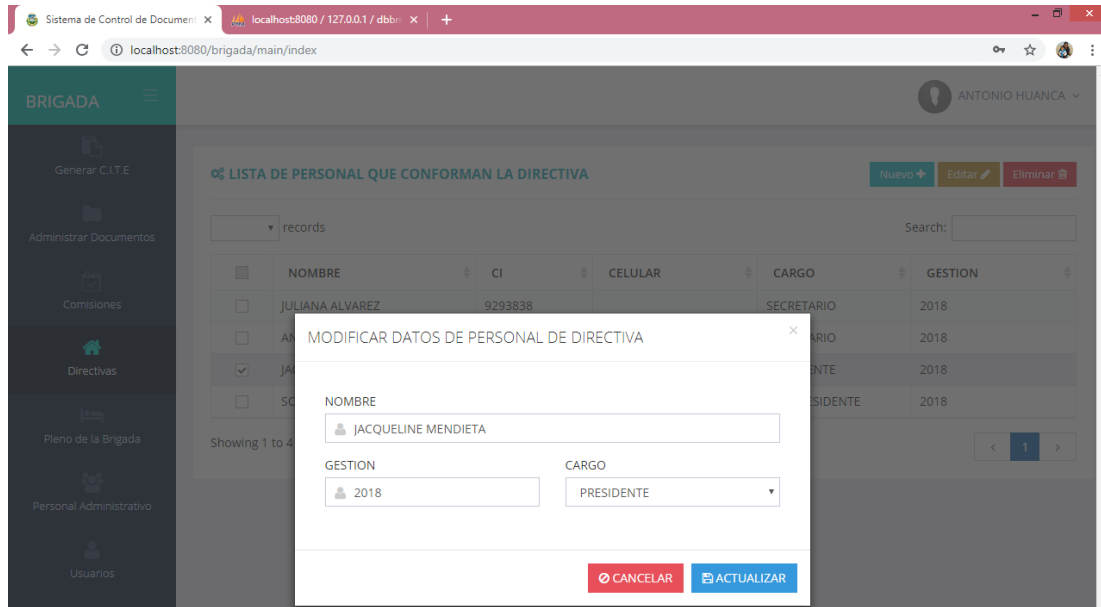


Figura 64 Pantalla Modificar Directiva

2.1.2.2.3.13.4.1.9 Pantalla Eliminar Directiva

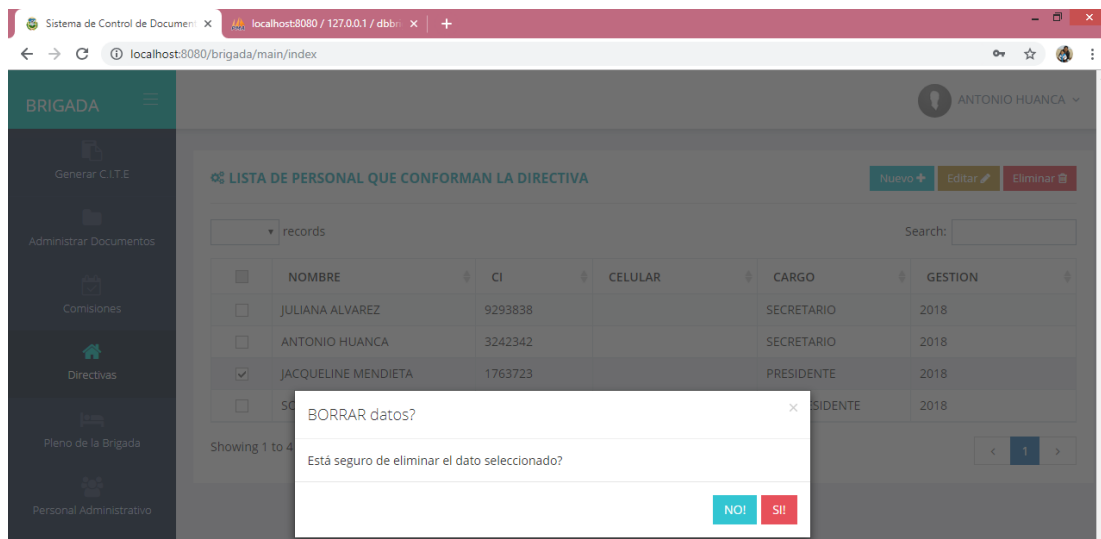


Figura 65 Pantalla Eliminar Directiva

2.1.2.2.3.13.4.1.10 Pantalla Gestión Comisiones

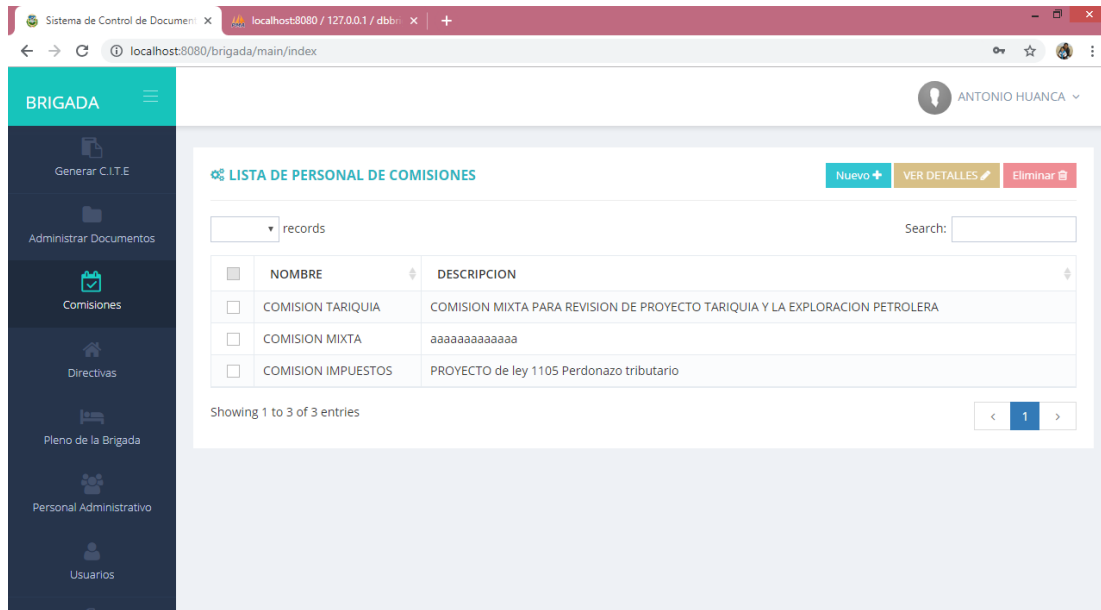


Figura 66 Pantalla Gestión Comisiones

2.1.2.2.3.13.4.1.11 Pantalla Adicionar Comisión

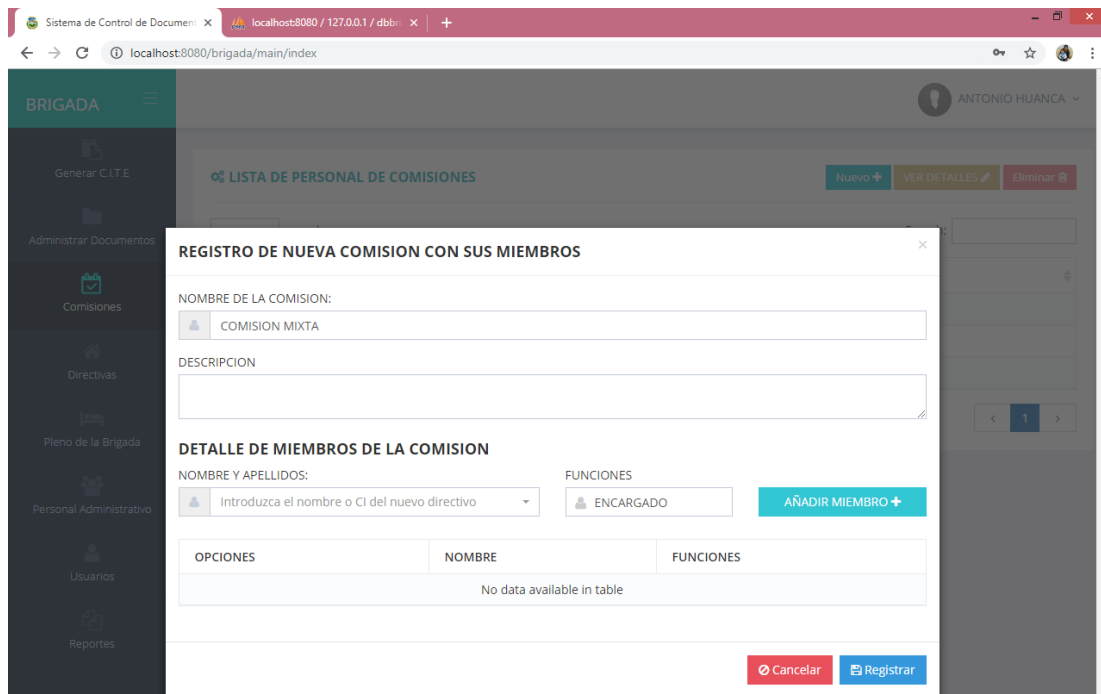


Figura 67 Pantalla Adicionar Comisión

2.1.2.2.3.13.4.1.12 Pantalla Ver Detalle Comisión

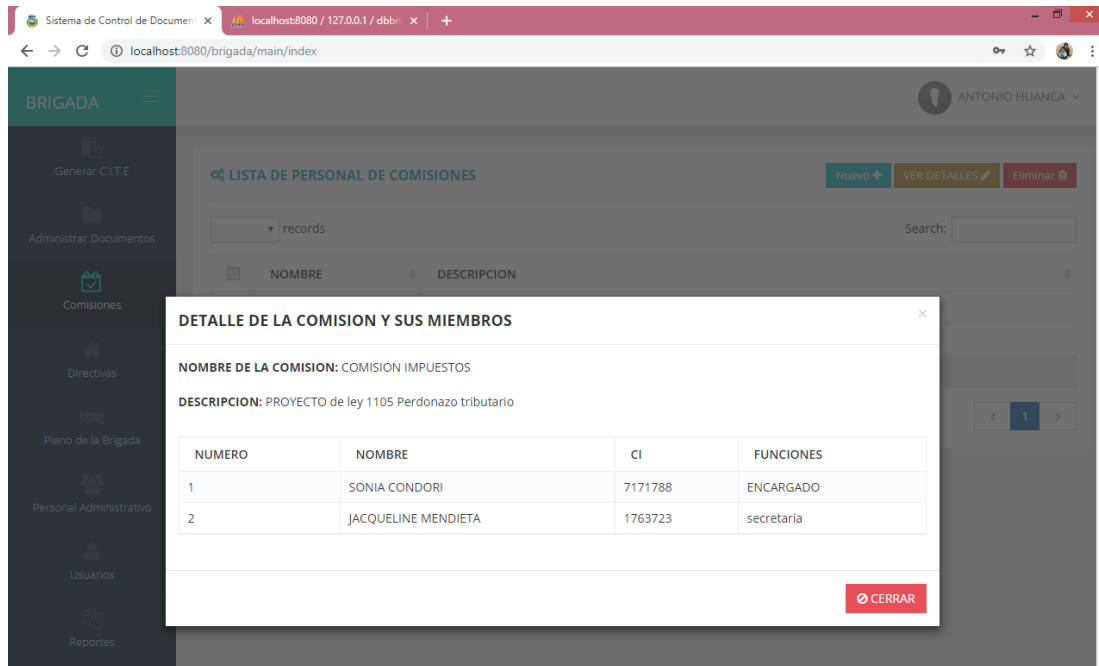


Figura 68 Pantalla Ver Detalle Comisión

2.1.2.2.3.13.4.1.13 Pantalla Eliminar Comisión

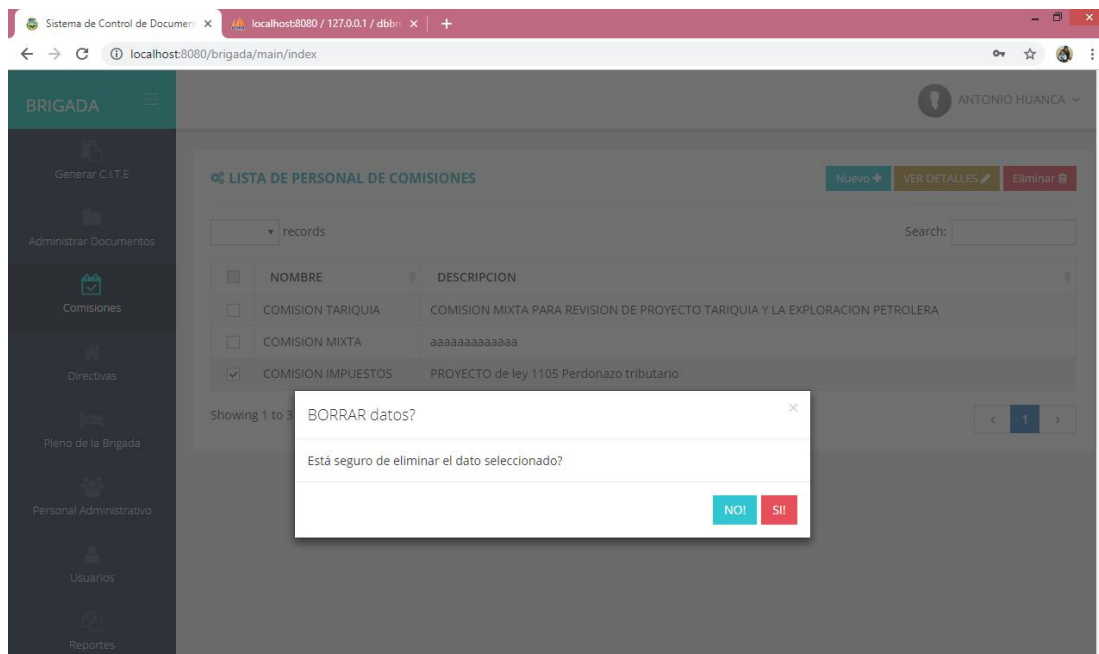


Figura 69 Pantalla Eliminar Comisión

2.1.2.2.3.13.4.1.14 Pantalla Gestión Administrativo

BRIGADA

Sistema de Control de Documentación

localhost:8080/brigada/Login/index

JUAN PORTAL

LISTADO DE PERSONAL ADMINISTRATIVO

Nuevo + Editar Eliminar

records Search:

	NOMBRE	CI	CELULAR	CORREO	CARGO
<input type="checkbox"/>	JUAN PORTAL	71298911M	34535435		SECRETARIO TECNICO
<input type="checkbox"/>	ALEX CAMPOSO	8723728	77777777		JEFE ADMINISTRATIVO 2
<input type="checkbox"/>	ANATOLIO DIAZ	82382383			ANIMADOR
<input type="checkbox"/>	JUAN IVAN EQUISE	7234238	7237443283	papi@brigada.com	JEFE DE TRANSPORTE
<input type="checkbox"/>	JANETH CONDORI	37283742			oficial mayor
<input type="checkbox"/>	JUAN ROMULO	12432432	8273648273	joelmanx2012@gmail.com	capataz

Showing 1 to 6 of 6 entries

< 1 >

Figura 70 Pantalla Gestión Administrativo

2.1.2.2.3.13.4.1.15 Pantalla Adicionar Administrativo

BRIGADA

Sistema de Control de Documentación

localhost:8080/brigada/Login/index

JUAN PORTAL

LISTADO DE PERSONAL ADMINISTRATIVO

Nuevo + Editar Eliminar

records Search:

REGISTRO DE PERSONAL ADMINISTRATIVO

NOMBRE* APELLIDO PATERNO* APELLIDO MATERNO

CI EXT. SEXO

CARGO CONTRATO FUNCIONES

EMAIL CELULAR

DIRECCION

* Campos obligatorios

Cerrar Guardar

Figura 71 Pantalla Adicionar Administrativo

2.1.2.2.3.13.4.1.16 Pantalla Modificar Administrativo

MODIFICAR DATOS DE PERSONAL ADMINISTRATIVO

Nombres(s) * APELLIDO PATERNO * APELLIDO MATERNO

JUAN PORTAL LLANOS

CI EXT. SEXO

7129891 1M MASCULINO

CARGO CONTRATO FUNCIONES

SECRETARIO TECNICO EVENTUAL APOYO DIRECTO A LA PRESIDENTA

EMAIL CELULAR

34535435

DIRECCION

Barrio fabril, calle fuerzs 322

* Campos obligatorios

Cerrar Guardar

Figura 72 Pantalla Modificar Administrativo

2.1.2.2.3.13.4.1.17 Pantalla Eliminar Administrativo

BORRAR datos?

Está seguro de eliminar el dato seleccionado?

NO! SI!

Figura 73 Pantalla Eliminar Administrativo

2.1.2.2.3.13.4.1.18 Pantalla Gestión Usuarios

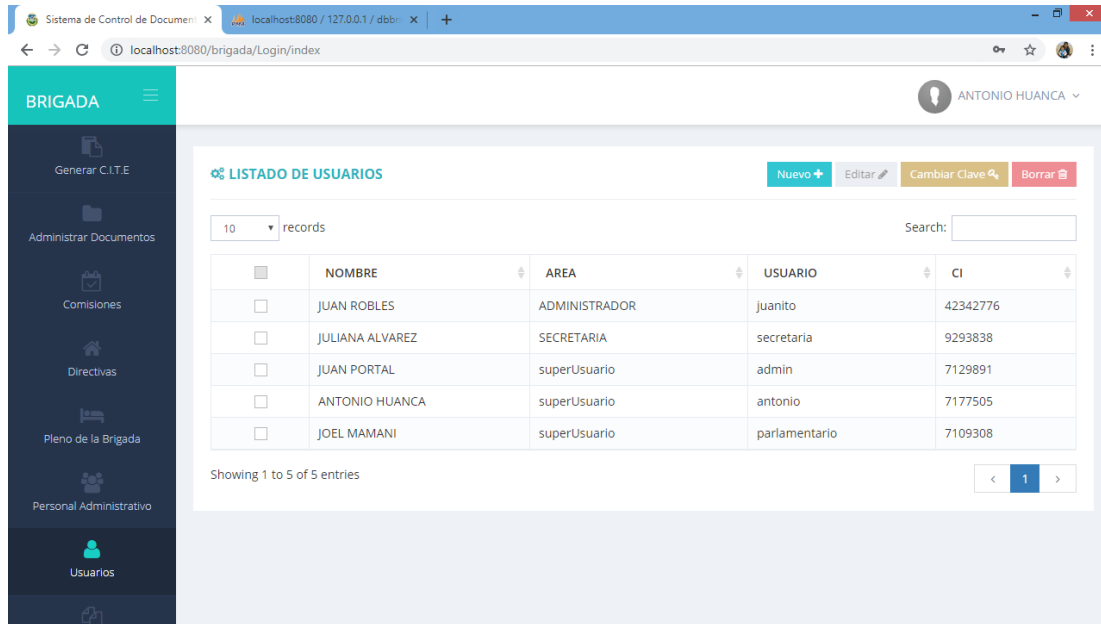


Figura 74 Pantalla Gestión Usuarios

2.1.2.2.3.13.4.1.19 Pantalla Adicionar Usuario

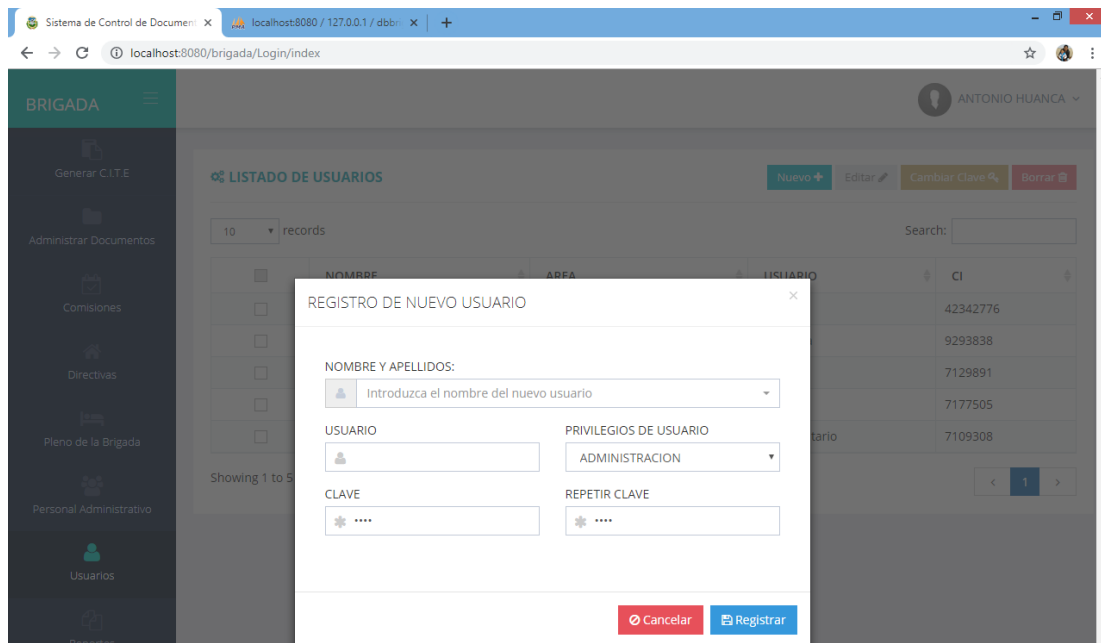


Figura 75 Pantalla Adicionar Usuario

2.1.2.2.3.13.4.1.20 Pantalla Modificar Usuario

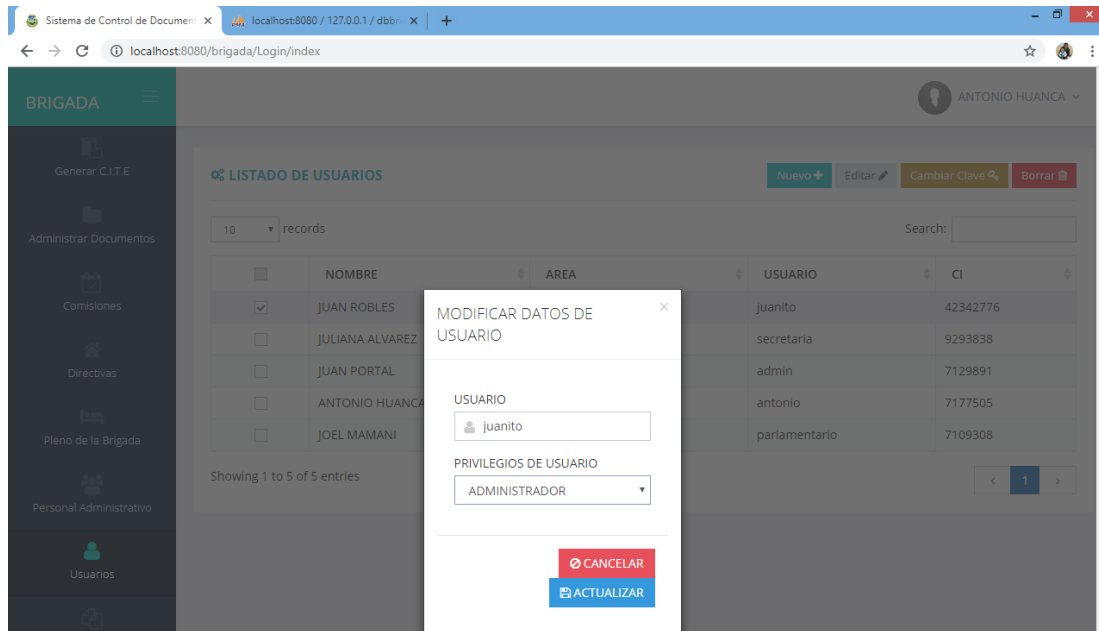


Figura 76 Pantalla Modificar Usuario

2.1.2.2.3.13.4.1.21 Pantalla Eliminar Usuario

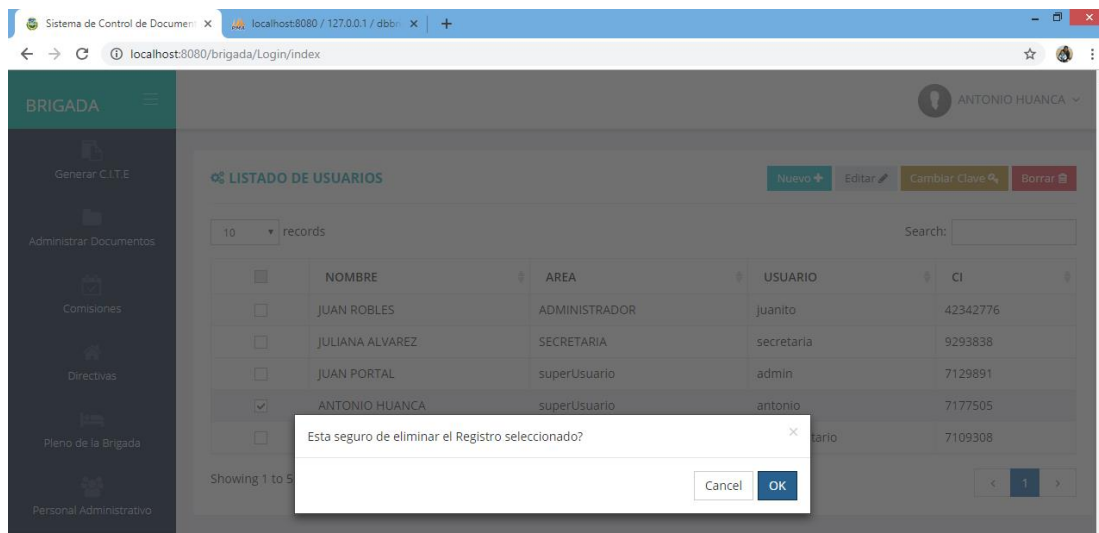


Figura 77 Pantalla Eliminar Usuario

2.1.2.2.3.13.4.1.22 Pantalla Gestión Hoja de Proveído

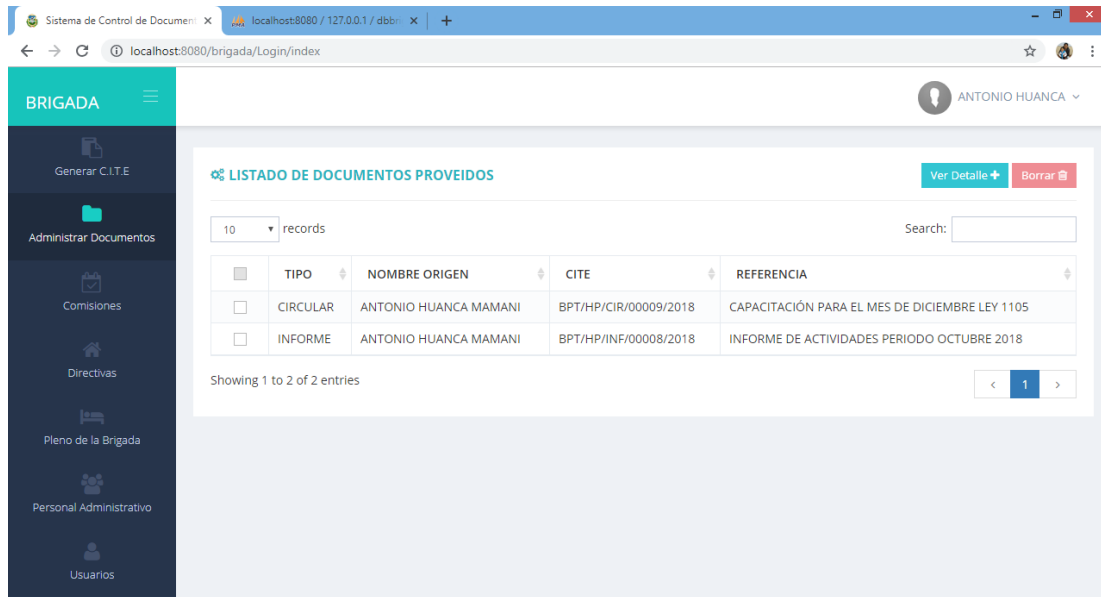


Figura 78 Pantalla Gestión Hoja de Proveído

2.1.2.2.3.13.4.1.23 Pantalla Ver Detalle del HP

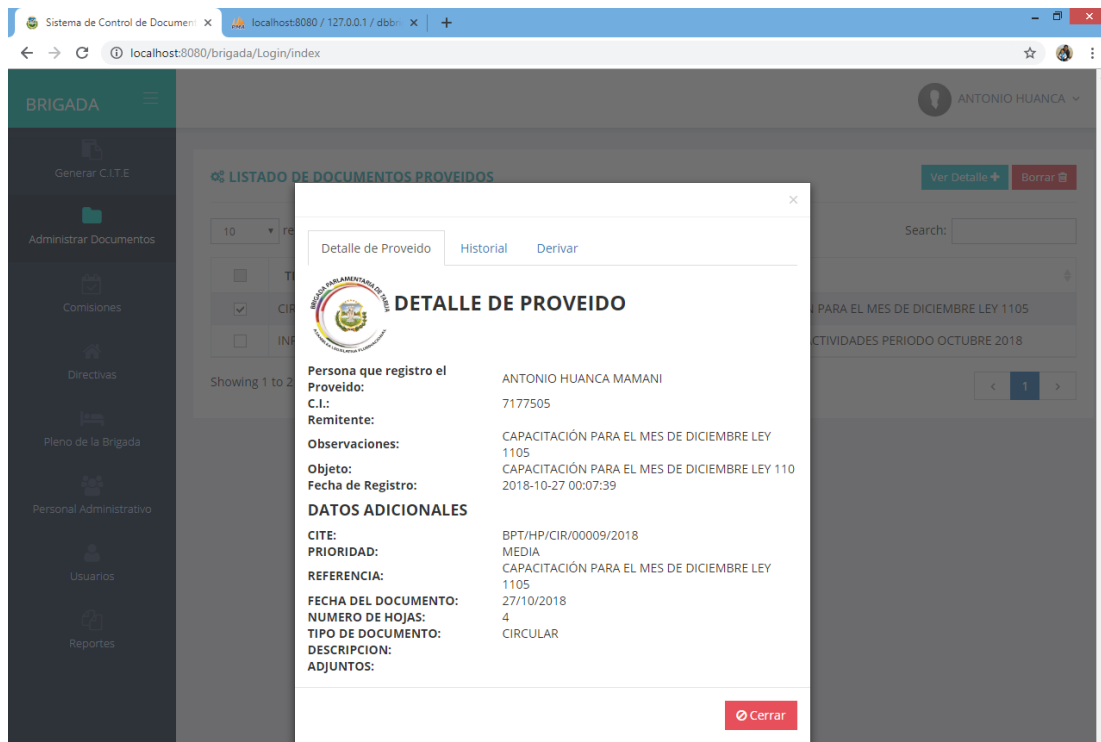


Figura 79 Pantalla Ver Detalle de HP

2.1.2.2.3.13.4.1.24 Pantalla Ver Historial del HP

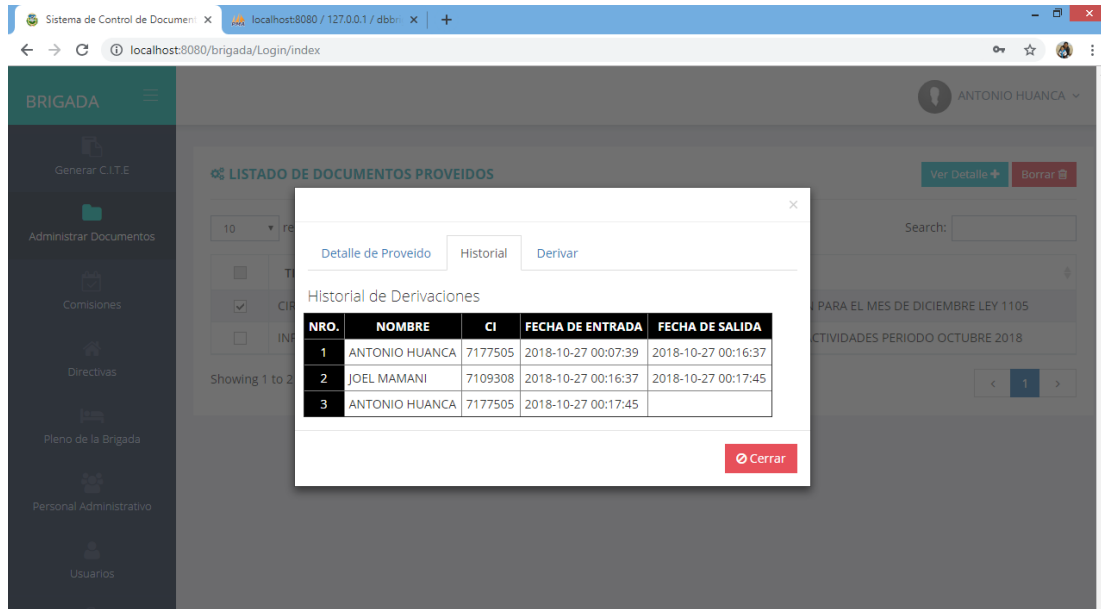


Figura 80 Pantalla Ver Historial del HP

2.1.2.2.3.13.4.1.25 Pantalla Derivar HP

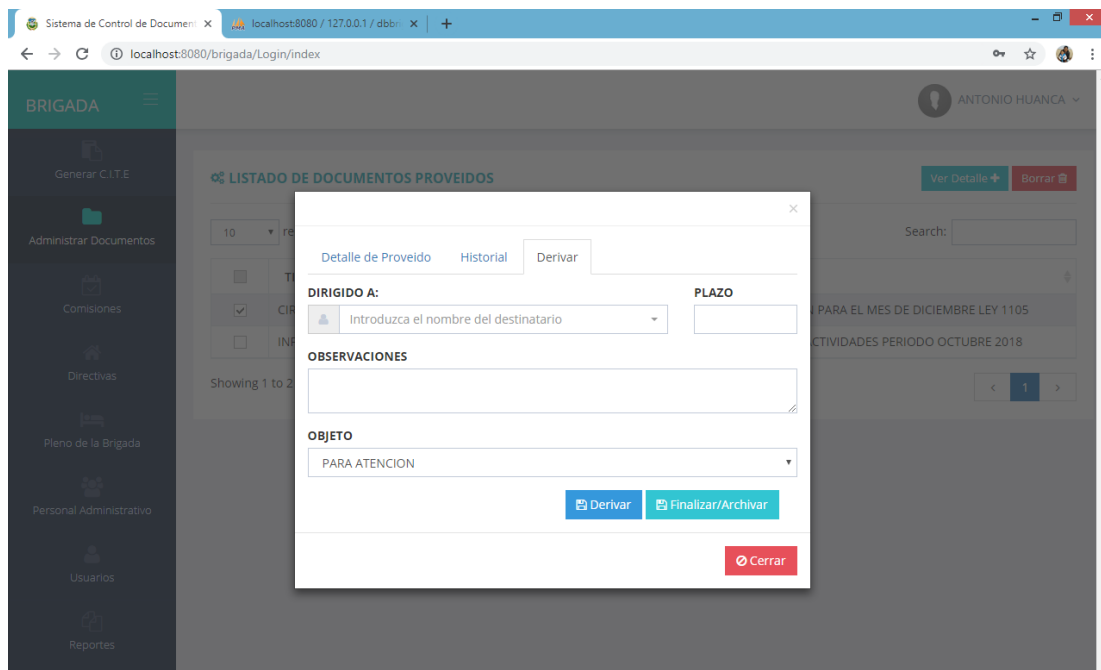


Figura 81 Pantalla Derivar HP

2.1.2.2.3.13.4.1.26 Pantalla Anular Cite de HP

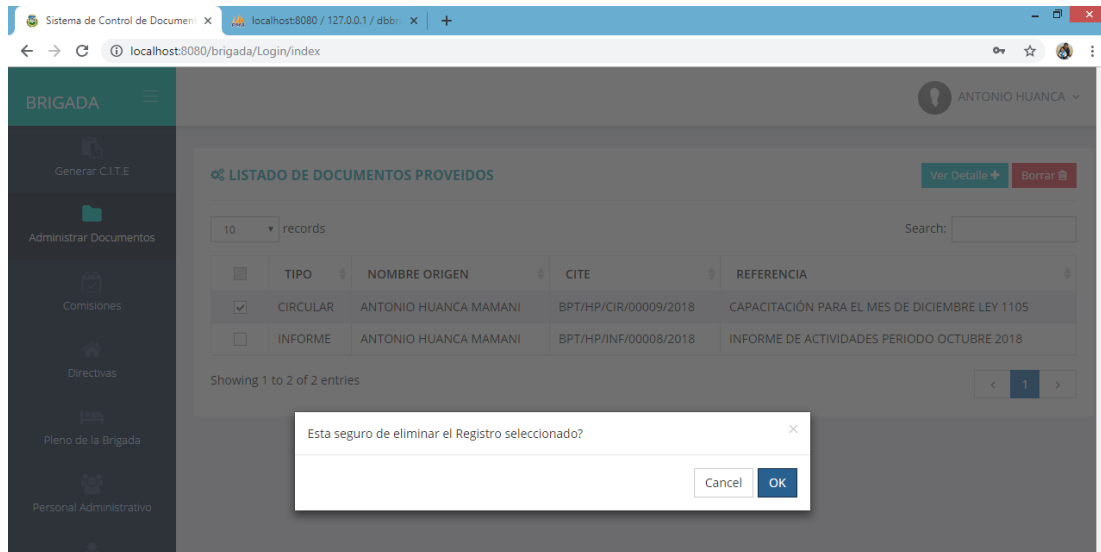


Figura 82 Pantalla Anular de Cite DE HP

2.1.2.2.3.13.4.1.27 Pantalla Gestión Hoja de Ruta

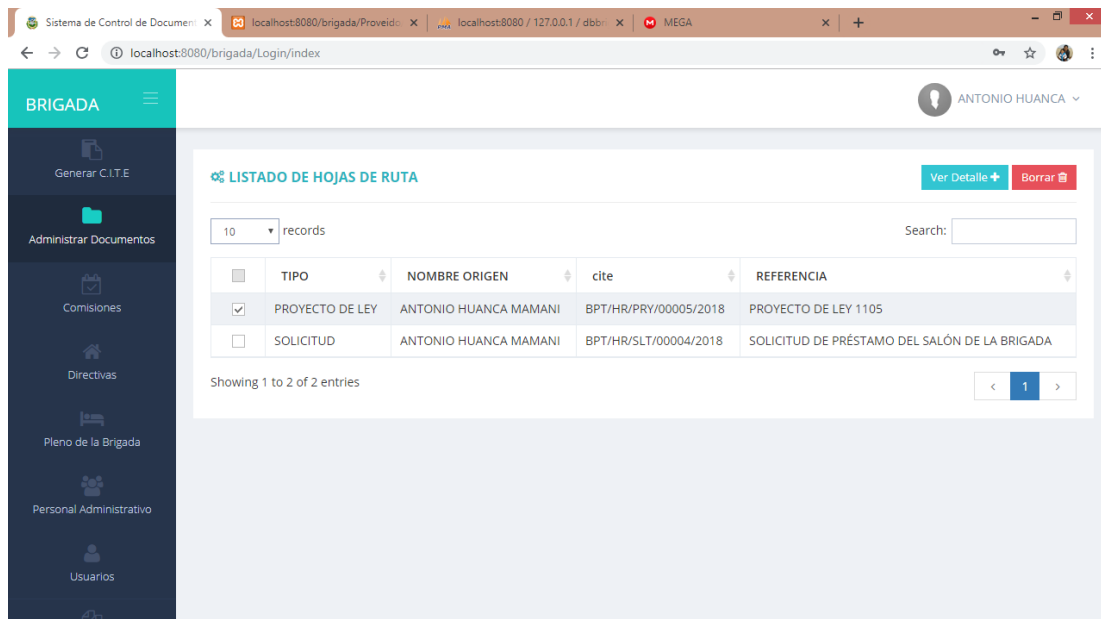


Figura 83 Pantalla Gestión Hoja De Ruta

2.1.2.2.3.13.4.1.28 Pantalla Ver Detalle de HR

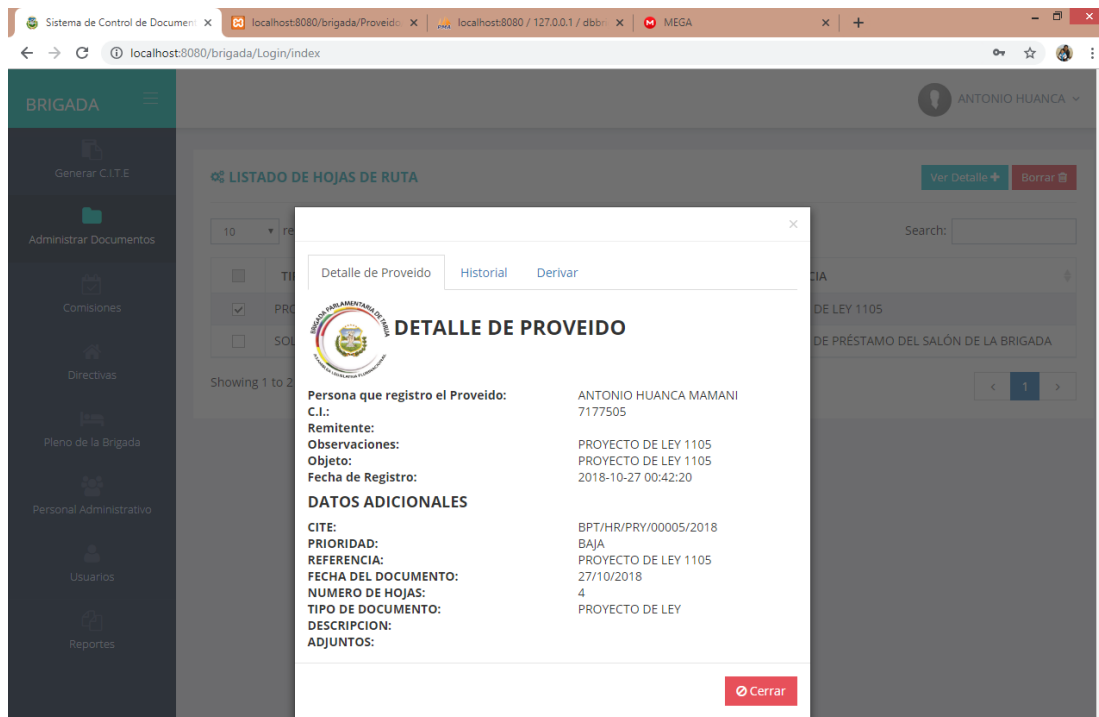


Figura 84 Pantalla Ver Detalle de HR

2.1.2.2.3.13.4.1.29 Pantalla Ver Historial de HR

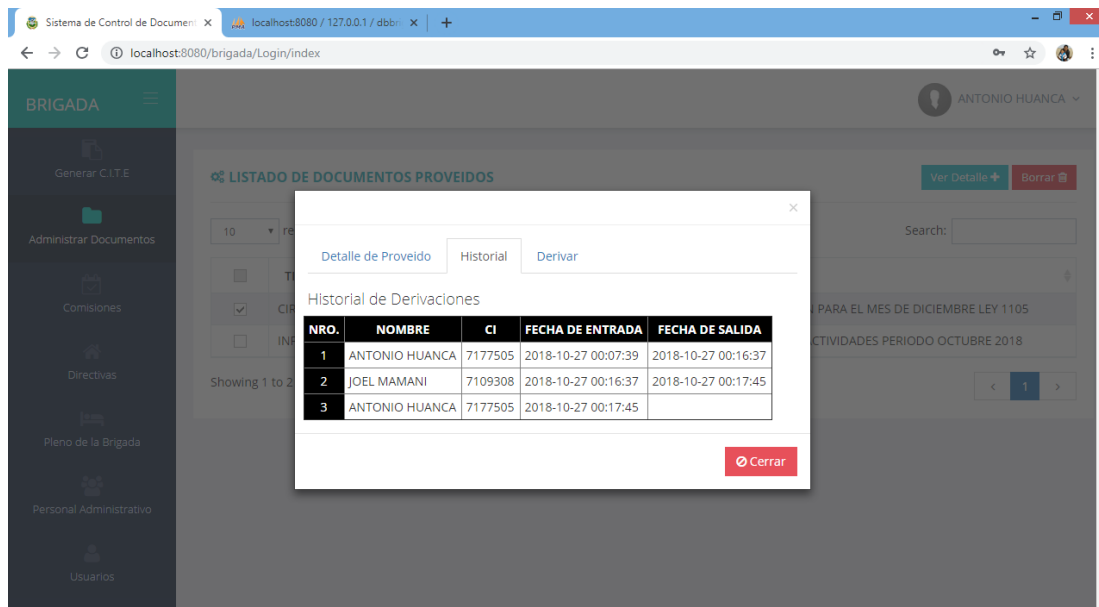


Figura 85 Pantalla Ver Historial de HR

2.1.2.2.3.13.4.1.30 Pantalla Derivar HR

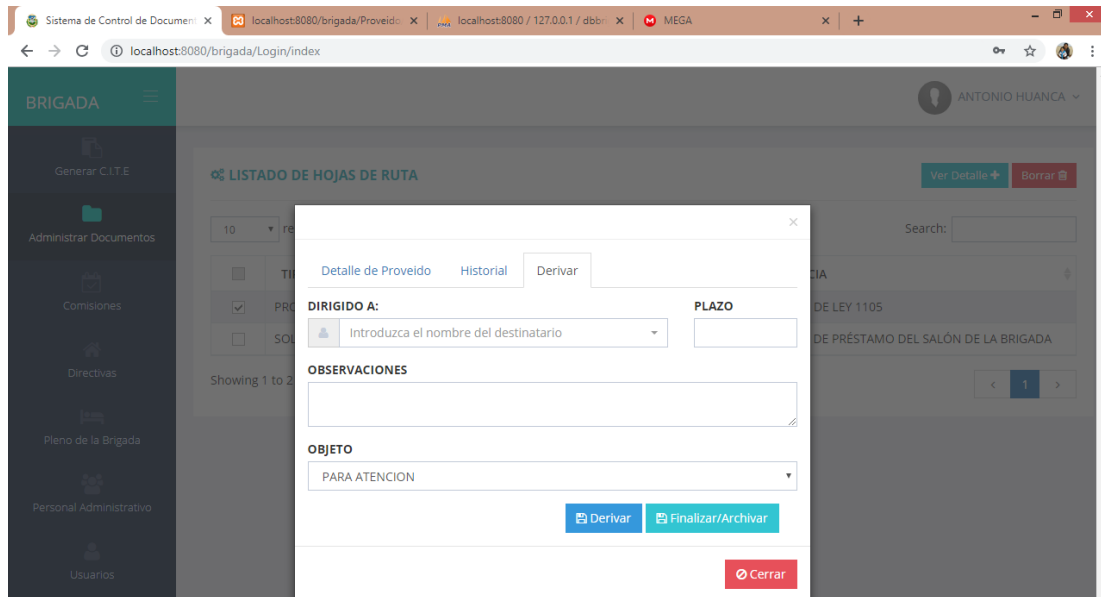


Figura 86 Pantalla Ver Historial de HR

2.1.2.2.3.13.4.1.31 Pantalla Anular Cite de HR

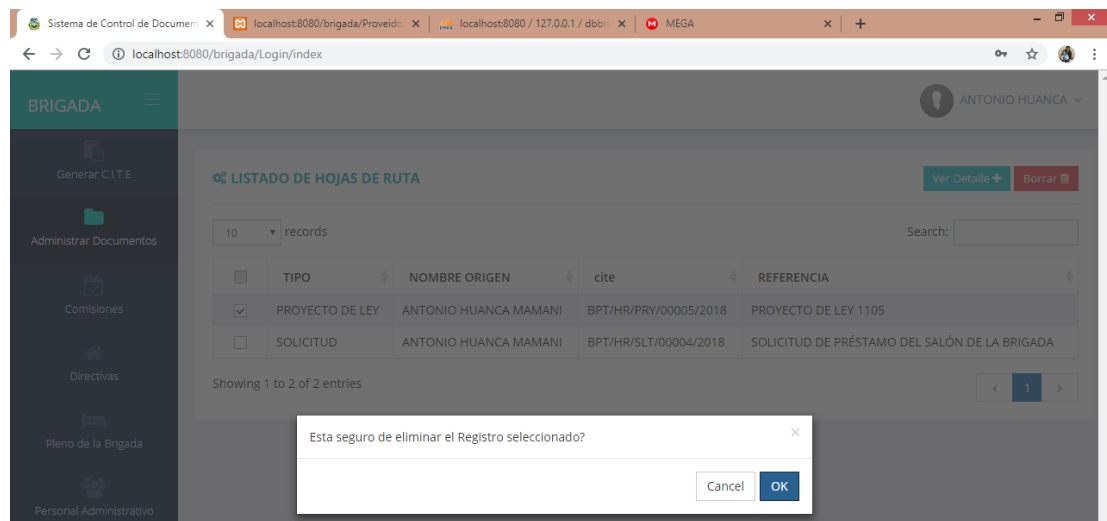


Figura 87 Pantalla Anular de Cite DE HR

2.1.2.2.3.13.4.1.32 Pantalla Generar Cite

Sistema de Control de Documentos

localhost:8080/brigada/Login/index

ANTONIO HUANCA

BRIGADA

Generar C.I.T.E.

Administrar Documentos

Comisiones

Directivas

Pleno de la Brigada

Personal Administrativo

Usuarios

Reportes

REGISTRO DE HOJA DE PROVEIDO

FECHA DEL DOCUMENTO: 29/10/2018

PRIORIDAD: ALTA

NRO. HOJAS: 4

TIPO DE DOCUMENTO: INFORME

EN RESPUESTA A: Introduzca el nombre de referencia del documento

REFERENCIA

ADJUNTOS: Seleccionar archivo Ningún archivo seleccionado

Limpiar Registrar

Figura 88 Pantalla Generar Cite

2.1.2.2.3.13.4.1.33 Pantalla Registro HP

Sistema de Control de Documentos

localhost:8080/brigada/Login/index

ANTONIO HUANCA

BRIGADA

Generar C.I.T.E.

Administrar Documentos

Comisiones

Directivas

Pleno de la Brigada

Personal Administrativo

Usuarios

Reportes

REGISTRO DE HOJA DE PROVEIDO

FECHA DEL DOCUMENTO: 29/10/2018

PRIORIDAD: ALTA

NRO. HOJAS: 4

TIPO DE DOCUMENTO: INFORME

EN RESPUESTA A: BPT/HP/INF/00008/2018 - INFORME DE ACTIVIDADES PERIODO OCTUBRE 2018

REFERENCIA: INFORME COMPLEMENTARIO AL INFORME DE ACTIVIDADES OCTUBRE 2018

ADJUNTOS: Seleccionar archivo Ningún archivo seleccionado

Limpiar Registrar

Figura 89 Pantalla Registro de HP

2.1.2.2.3.13.4.1.34 Pantalla Registro de HR

BRIGADA

Generar C.I.T.E

Administrar Documentos

Comisiones

Directivas

Pleno de la Brigada

Personal Administrativo

Usuarios

ANTONIO HUANCA

REGISTRO DE HOJA DE RUTA

FECHA DEL DOCUMENTO: 29/10/2018

PRIORIDAD: ALTA

NRO. HOJAS: 4

TIPO DE DOCUMENTO: INFORME

DIRIGIDO A: Introduzca el nombre del destinatario

CI DEL REMITENTE:

CI EXT.:

NOMBRE REMITENTE:

EN RESPUESTA A: Introduzca el nombre de referencia del documento

REFERENCIA:

ADJUNTOS: Seleccionar archivo Ningún archivo seleccionado

[Limpiar](#) [Registrar](#)

Figura 90 Pantalla Registro de HP

2.1.2.2.3.13.4.1.35 Pantalla Gestión Reportes HP/HR

BRIGADA

Generar C.I.T.E

Administrar Documentos

Comisiones

Directivas

Pleno de la Brigada

Personal Administrativo

Usuarios

Reportes

ANTONIO HUANCA

REPORTES DE DOCUMENTOS PROVEIDOS Y HOJAS DE RUTA

Imprimir Detalle

5 records Search:

	TIPO	NOMBRE ORIGEN	ORIGEN	REFERENCIA
<input type="checkbox"/>	INFORME	ANTONIO HUANCA MAMANI		INFORME COMPLEMENTARIO AL INFORME DE ACTIVIDADES OCTUBRE 2018
<input type="checkbox"/>	PROYECTO DE LEY	ANTONIO HUANCA MAMANI		PROYECTO DE LEY 1105
<input type="checkbox"/>	SOLICITUD	ANTONIO HUANCA MAMANI		SOLICITUD DE PRÉSTAMO DEL SALÓN DE LA BRIGADA
<input type="checkbox"/>	CIRCULAR	ANTONIO HUANCA MAMANI		CAPACITACIÓN PARA EL MES DE DICIEMBRE LEY 1105
<input type="checkbox"/>	INFORME	ANTONIO HUANCA MAMANI		INFORME DE ACTIVIDADES PERIODO OCTUBRE 2018

Showing 1 to 5 of 14 entries

< 1 2 3 >

Figura 91 Pantalla Gestión Reportes HP/HR

2.1.2.2.3.13.4.1.36 Pantalla Imprimir HP/HR

Figura 92 Pantalla Registro de HP

2.1.2.2.3.13.4.1.37 Pantalla Cambiar Clave

Figura 93 Pantalla Cambiar Clave

2.1.2.2.3.13.4.1.38 Pantalla Buscar Cite

VER ESTADO DE MI CITE

CODIGO CITE +

Introduce tu CITE

< Regresar ✓ VER ESTADO

Figura 94 Pantalla Buscar Cite

2.1.2.2.3.13.4.1.39 Pantallas Ver Estado

VER ESTADO DE MI CITE

Detalle de Proveido Historial

DETALLE DE PROVEIDO

Persona que registro el Proveido: ANTONIO HUANCA MAMANI

C.I.: 7177505

Remitente: INFORME DE ACTIVIDADES PERIODO OCTUBRE 2018

Observaciones: INFORME DE ACTIVIDADES PERIODO OCTUBRE 2018

Objeto: INFORME DE ACTIVIDADES PERIODO OCTUBRE 2018

Fecha de Registro: 2018-10-27 00:06:03

DATOS ADICIONALES

CITE: BPT/HP/INF/00008/2018

PRIORIDAD: ALTA

REFERENCIA: INFORME DE ACTIVIDADES PERIODO OCTUBRE 2018

FECHA DEL DOCUMENTO: 27/10/2018

NUMERO DE HOJAS: 4

TIPO DE DOCUMENTO: INFORME

DESCRIPCION:

ADJUNTOS:

Cerrar

Figura 95 Pantalla Ver Estado

2.1.2.2.3.14 Modelo de Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

2.1.2.2.3.14.1.1 Modelo de Diagrama de Actividades

2.1.2.2.3.14.1.1.1 Diagrama de Actividad: CU-001 Iniciar Sesión

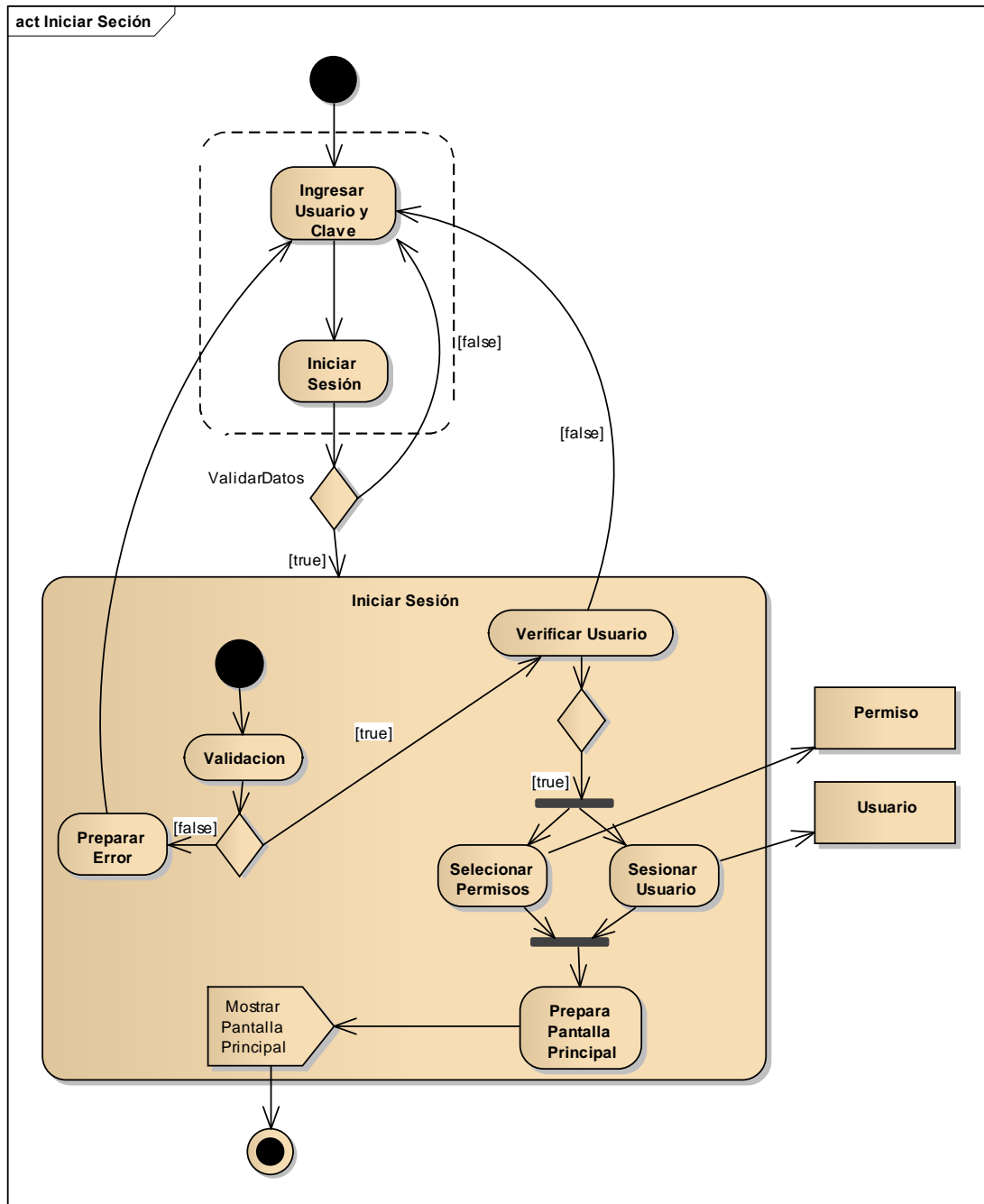


Figura 96 Diagrama de Actividad: CU-001 Iniciar Sesión

2.1.2.2.3.14.1.1.2 Diagrama de Actividad: CU-100 Gestión Pleno

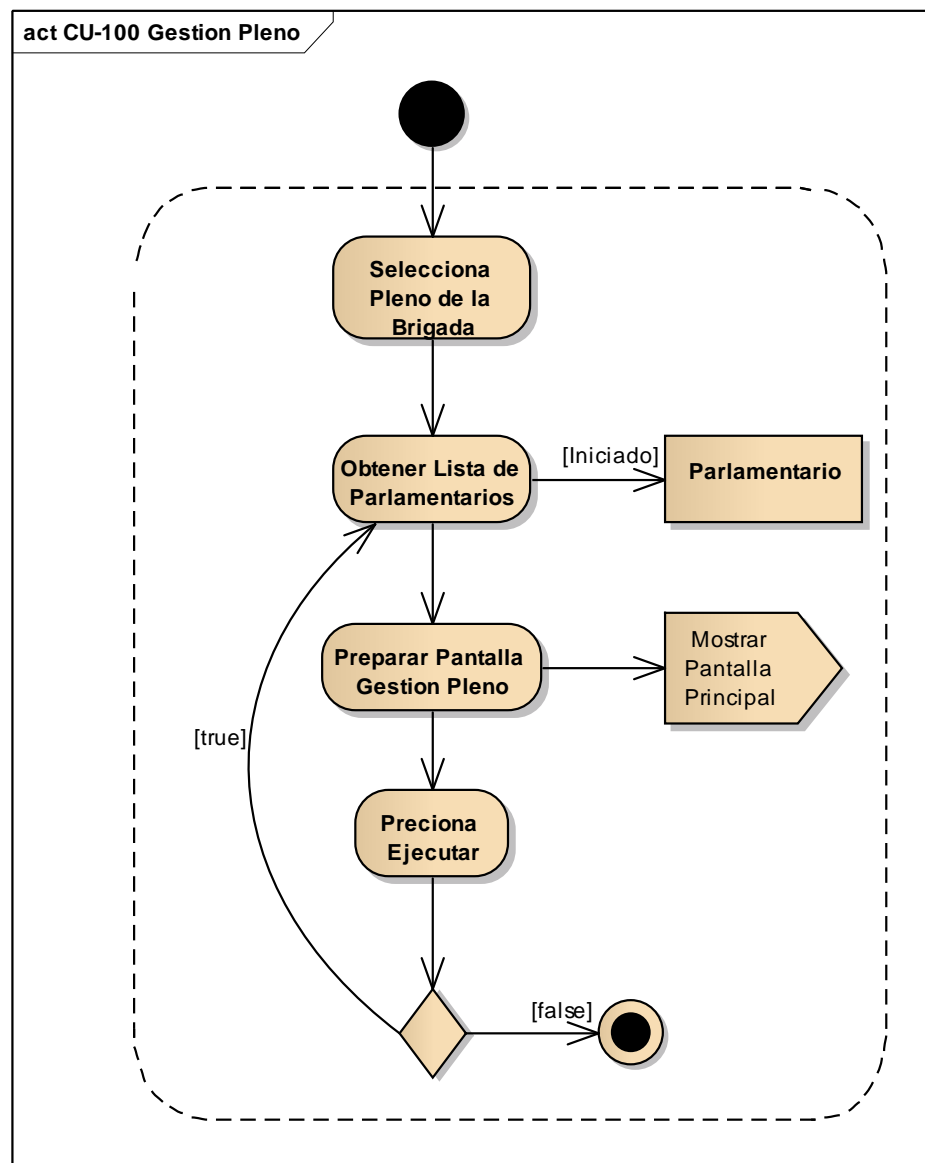


Figura 97 Diagrama de Actividad: CU-100 Gestión Pleno

2.1.2.2.3.14.1.1.3 Diagrama de Actividad: CU-101 Adicionar Parlamentario

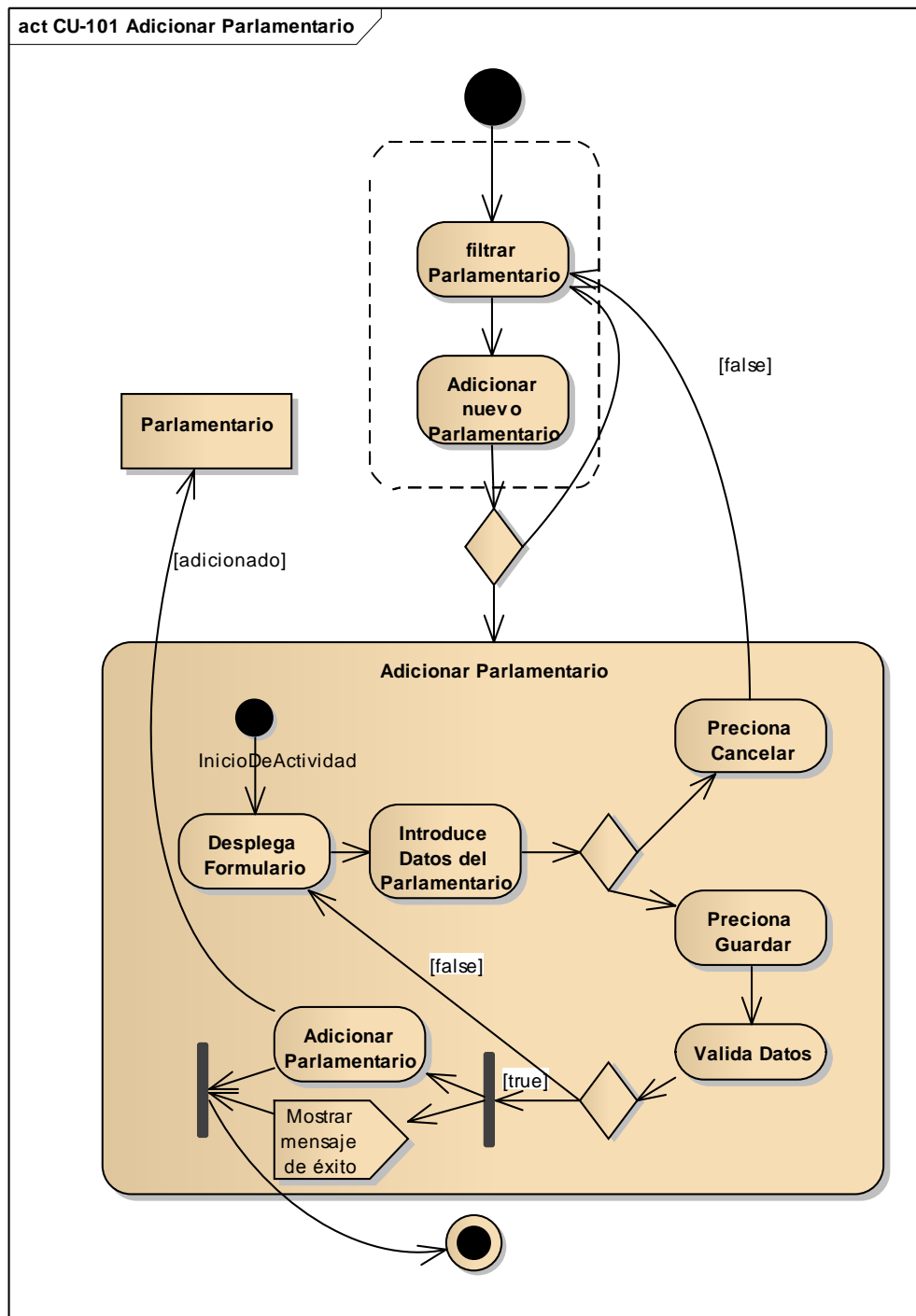


Figura 98 Diagrama de Actividad: CU-101 Adicionar Parlamentario

2.1.2.2.3.14.1.1.4 Diagrama de Actividad: CU-102 Modificar Parlamentario

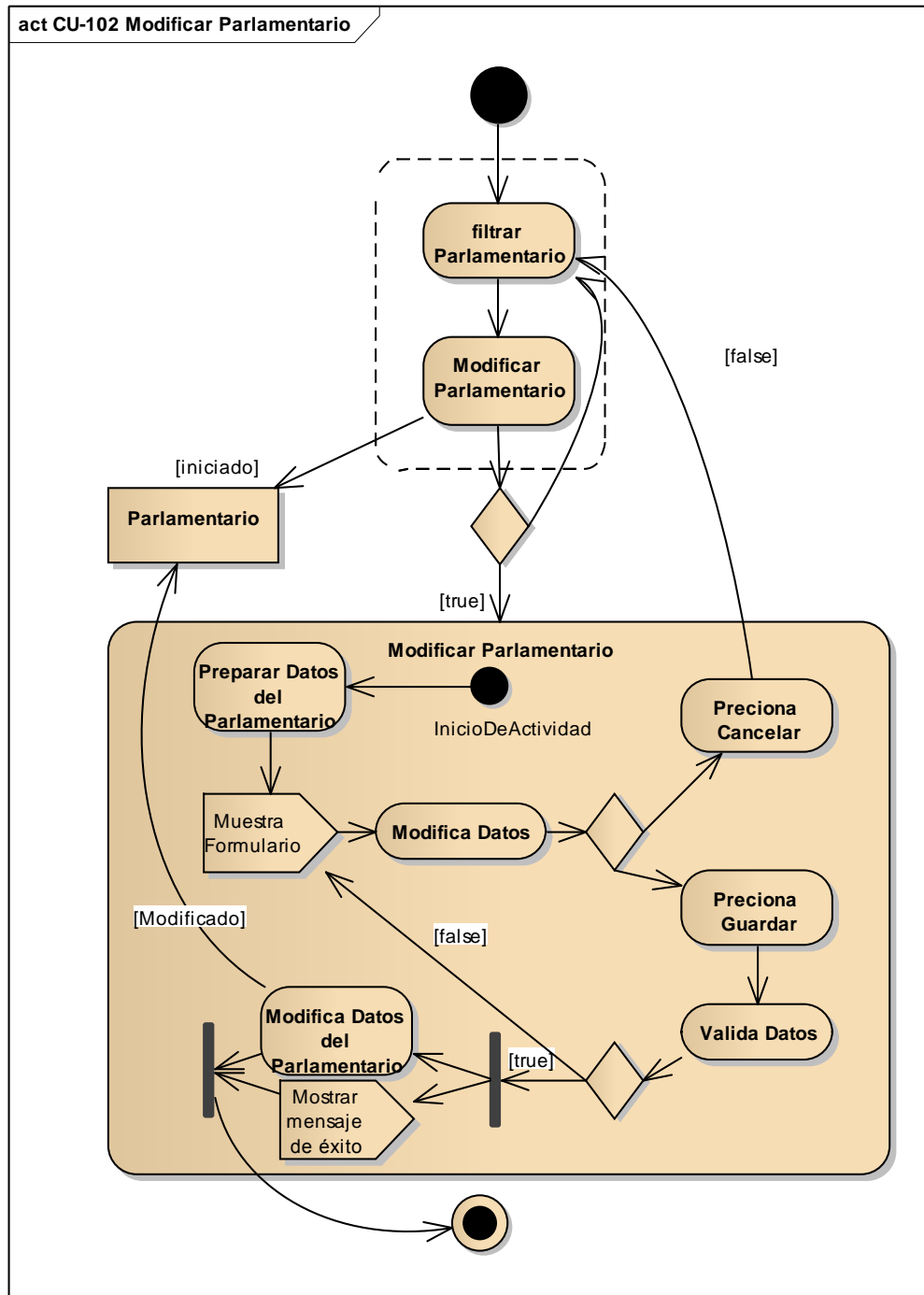


Figura 99 Diagrama de Actividad: CU-102 Modificar Parlamentario

2.1.2.2.3.14.1.1.5 Diagrama de Actividad: CU-103 Eliminar Parlamentario

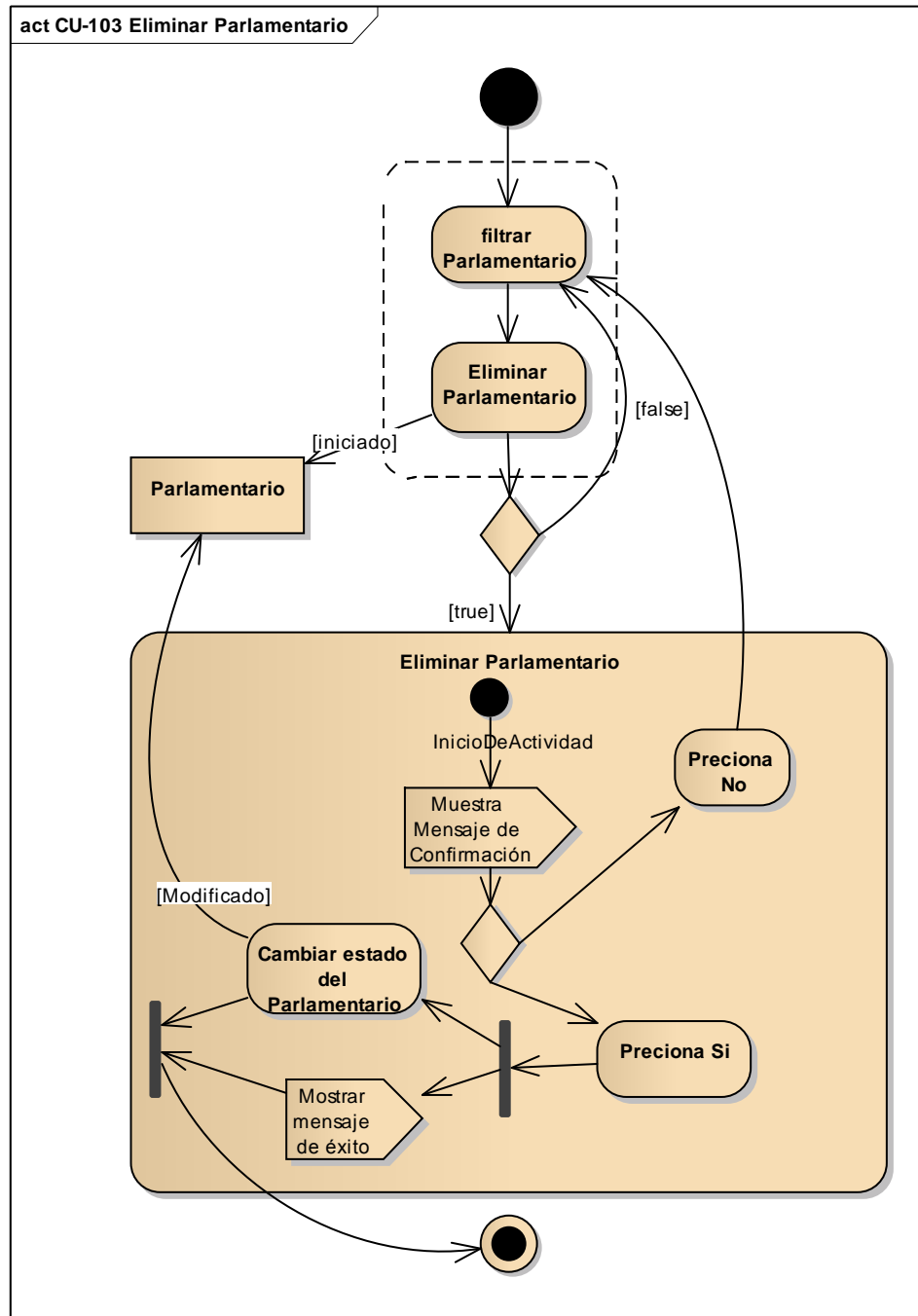


Figura 100 Diagrama de Actividad: CU-103 Eliminar Parlamentario

2.1.2.2.3.14.1.1.6 Diagrama de Actividad: CU-200 Gestión Directiva

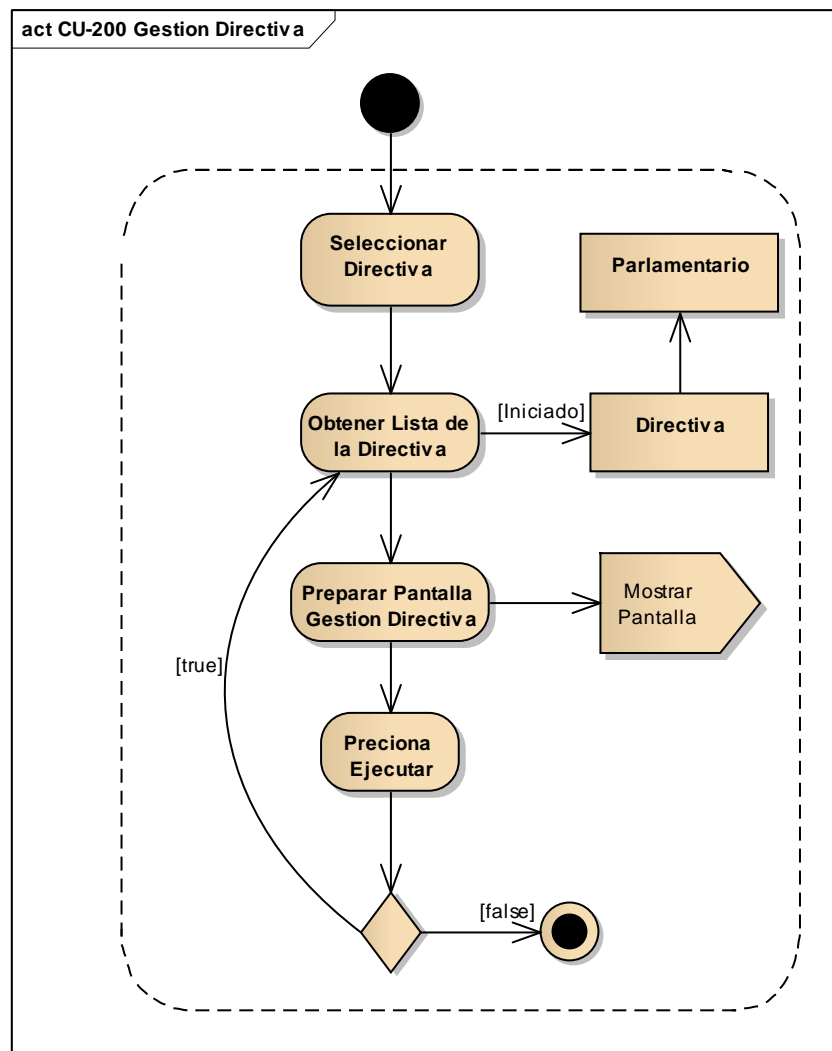


Figura 101 Diagrama de Actividad: CU-200 Gestión Directiva

2.1.2.2.3.14.1.1.7 Diagrama de Actividad: CU-300 Gestión Comisiones

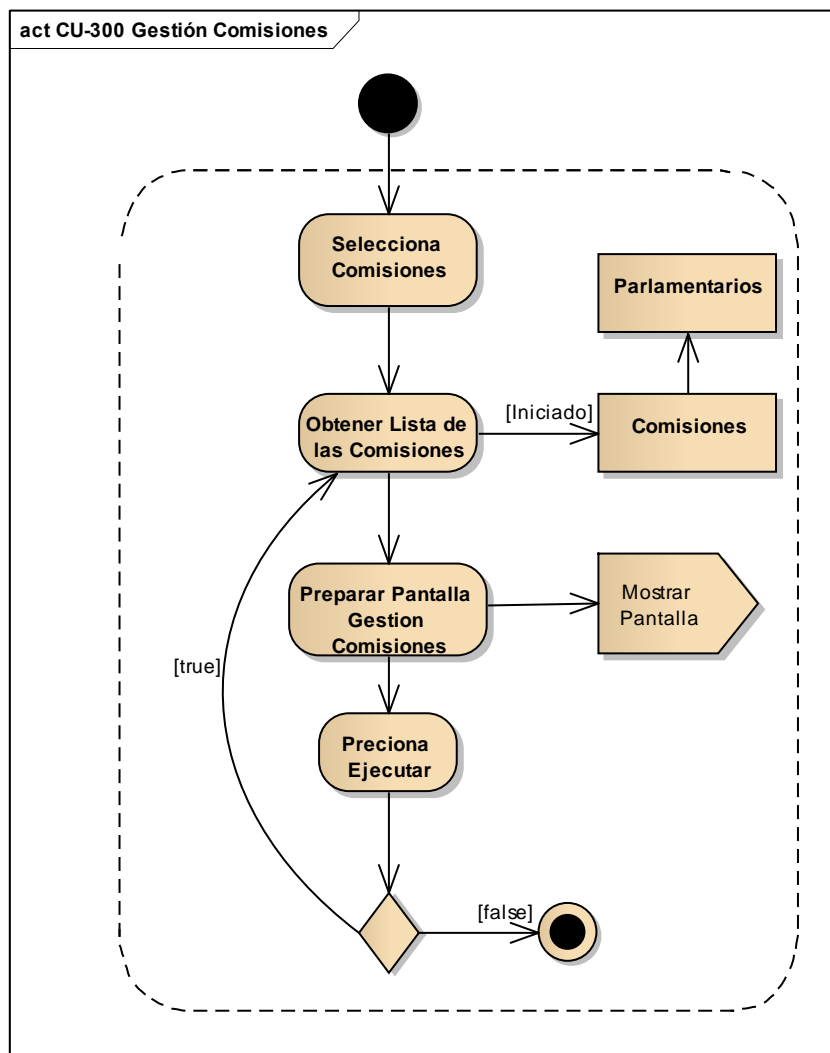


Figura 102 Diagrama de Actividad: CU-300 Gestión Comisiones

2.1.2.2.3.14.1.1.8 Diagrama de Actividad: CU-400 Gestión Administrativos

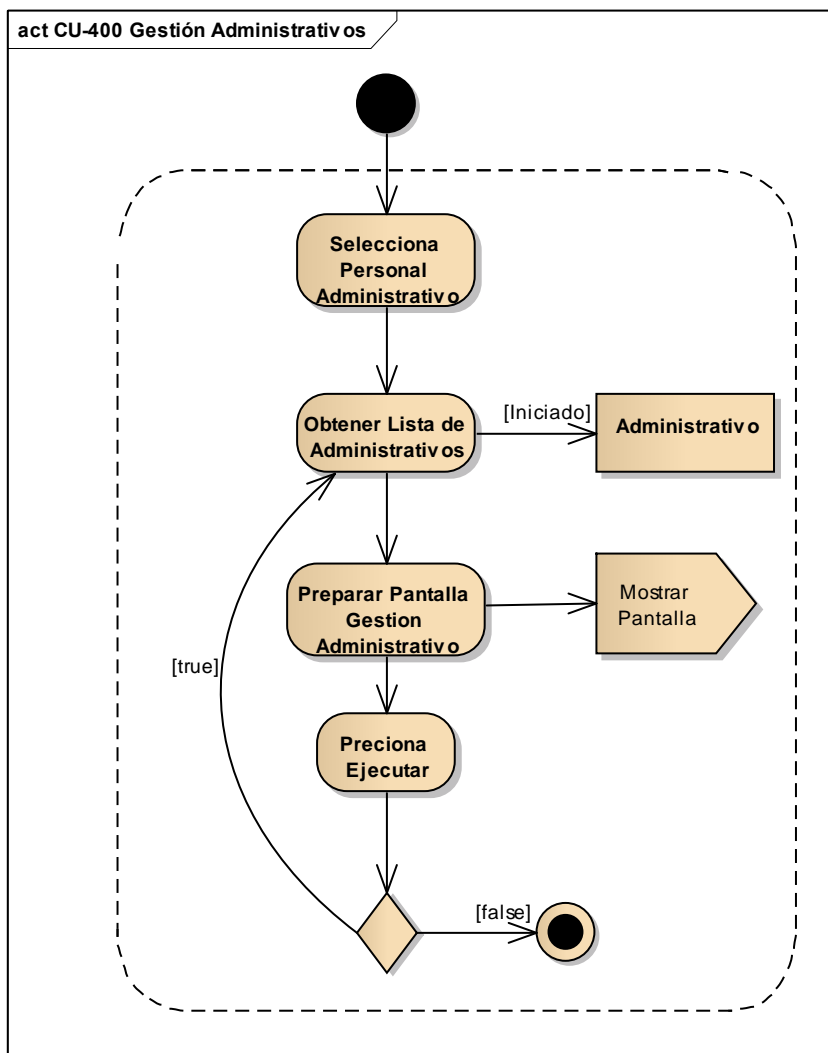


Figura 103 Diagrama de Actividad: CU-400 Gestión Administrativos

2.1.2.2.3.14.1.1.9 Diagrama de Actividad: CU-500 Gestión Usuarios

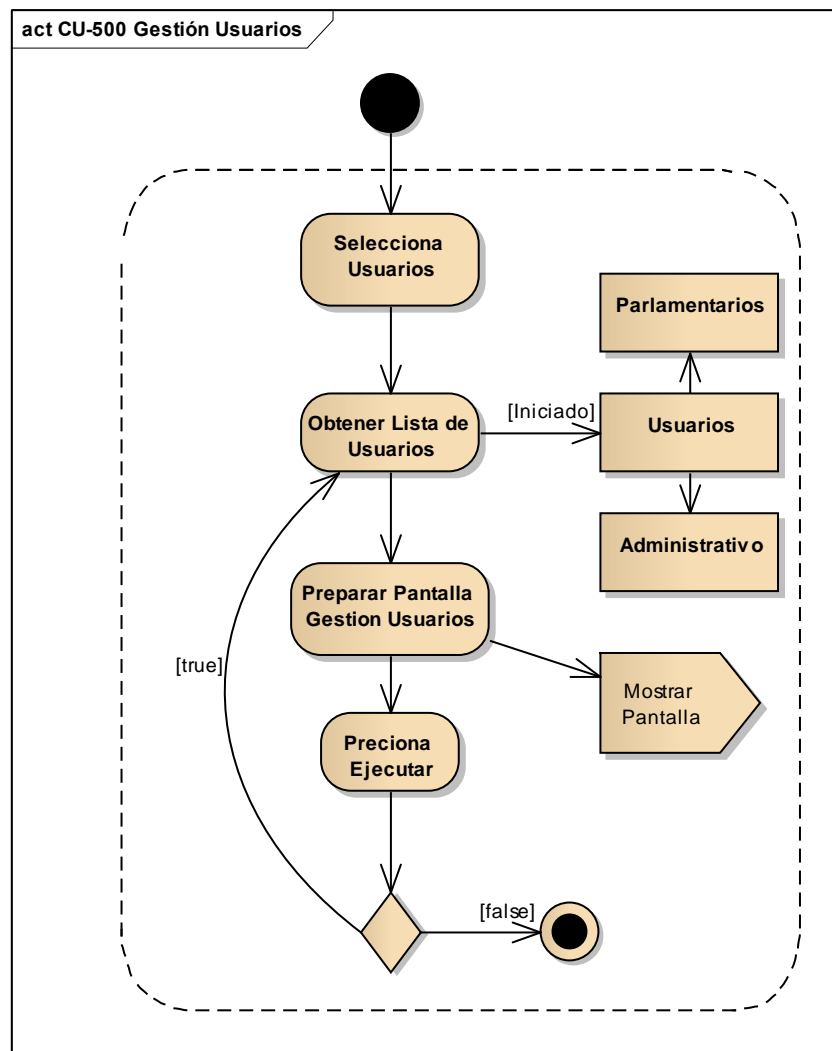


Figura 104 Diagrama de Actividad: CU-500 Gestión Usuarios

2.1.2.2.3.14.1.1.10 Diagrama de Actividad: CU-501 Adicionar Usuario

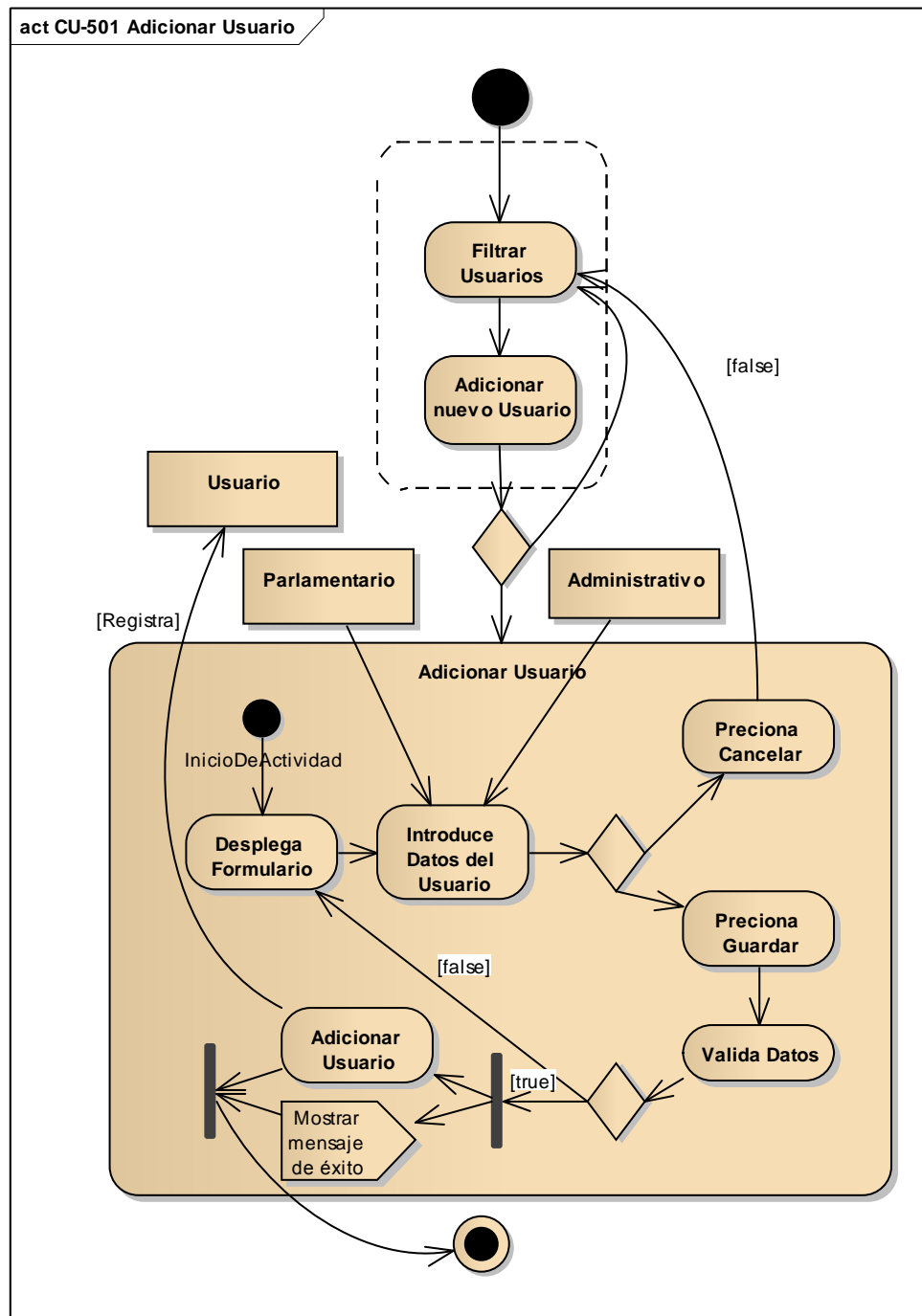


Figura 105 Diagrama de Actividad: CU-501 Adicionar Usuario

2.1.2.2.3.14.1.1.11 Diagrama de Actividad: CU-610 Gestión Hoja de Proveído

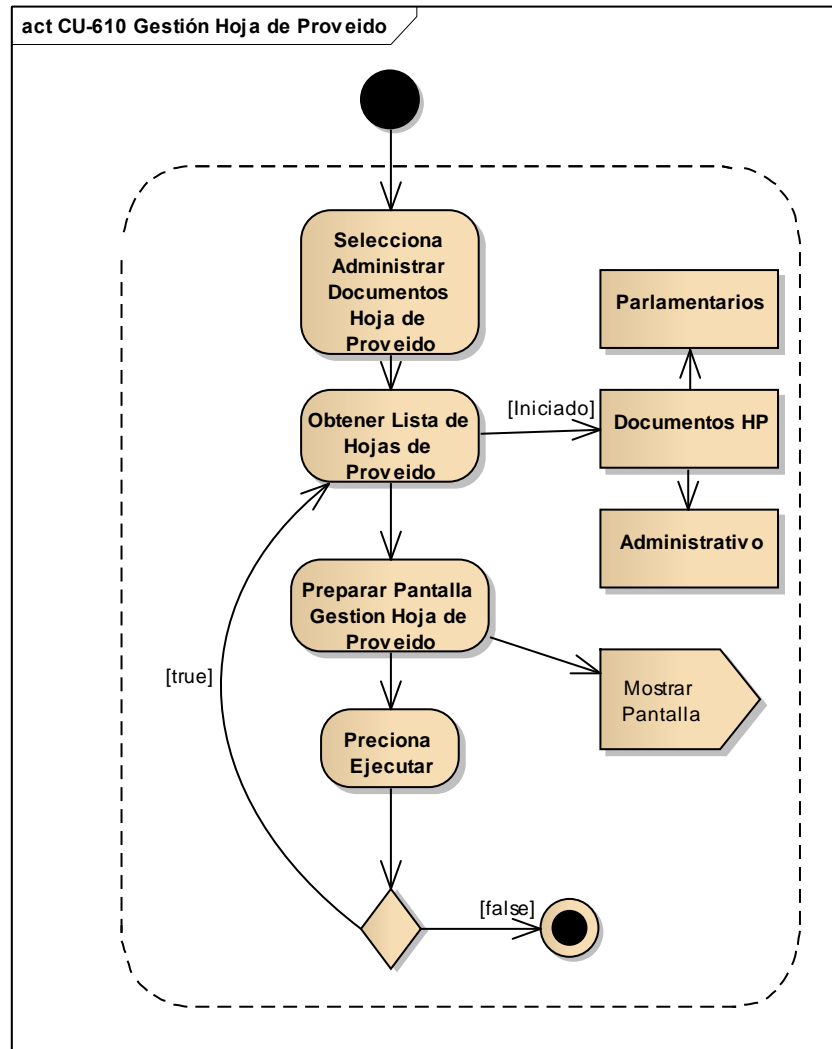


Figura 106 Diagrama de Actividad: CU-610 Gestión Hoja de Proveído

2.1.2.2.3.14.1.1.12 Diagrama de Actividad: CU-611 Ver Detalle del HP

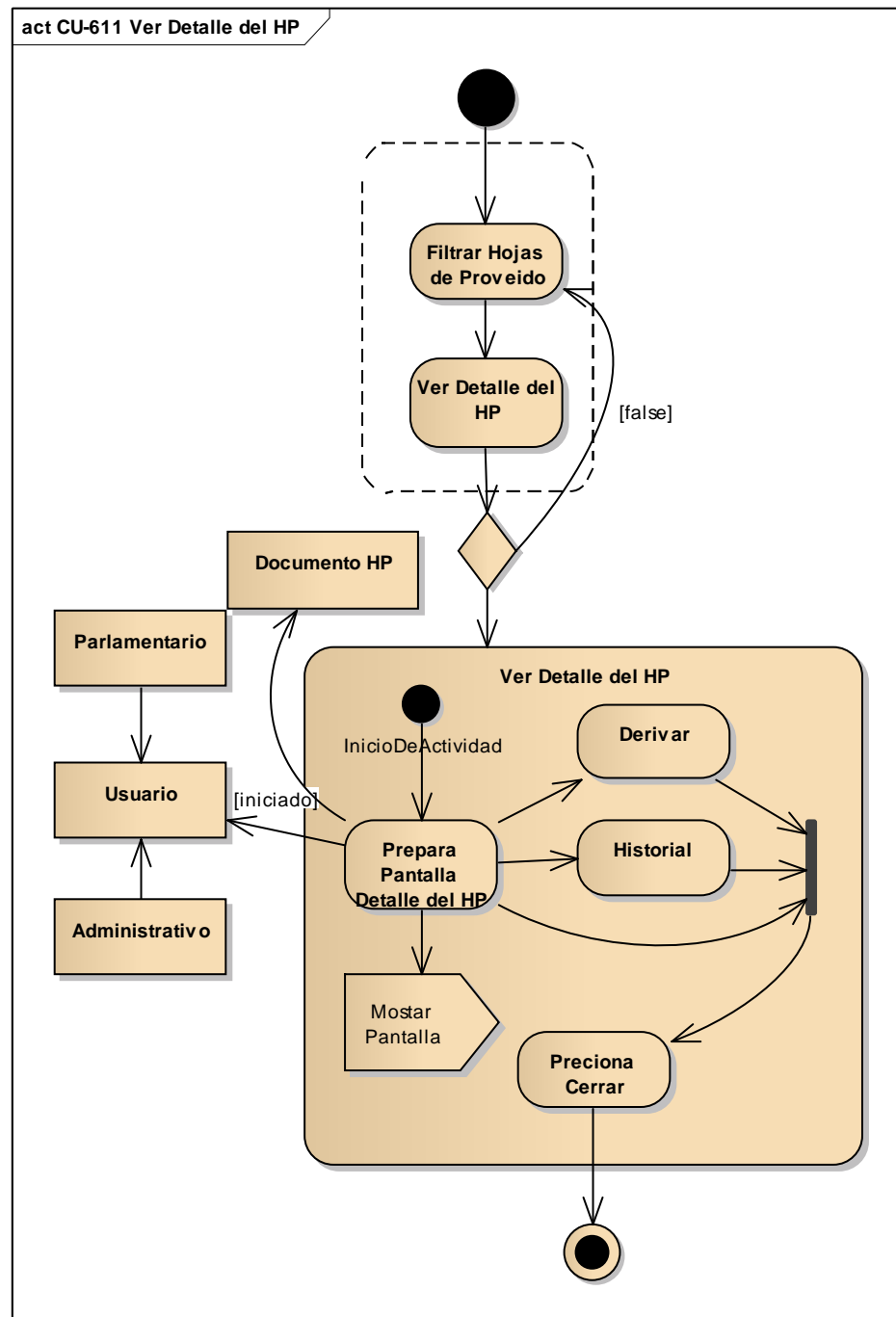


Figura 107 Diagrama de Actividad: CU-611 Ver Detalle del HP

2.1.2.2.3.14.1.1.13 Diagrama de Actividad: CU-613 Ver Historial del HP

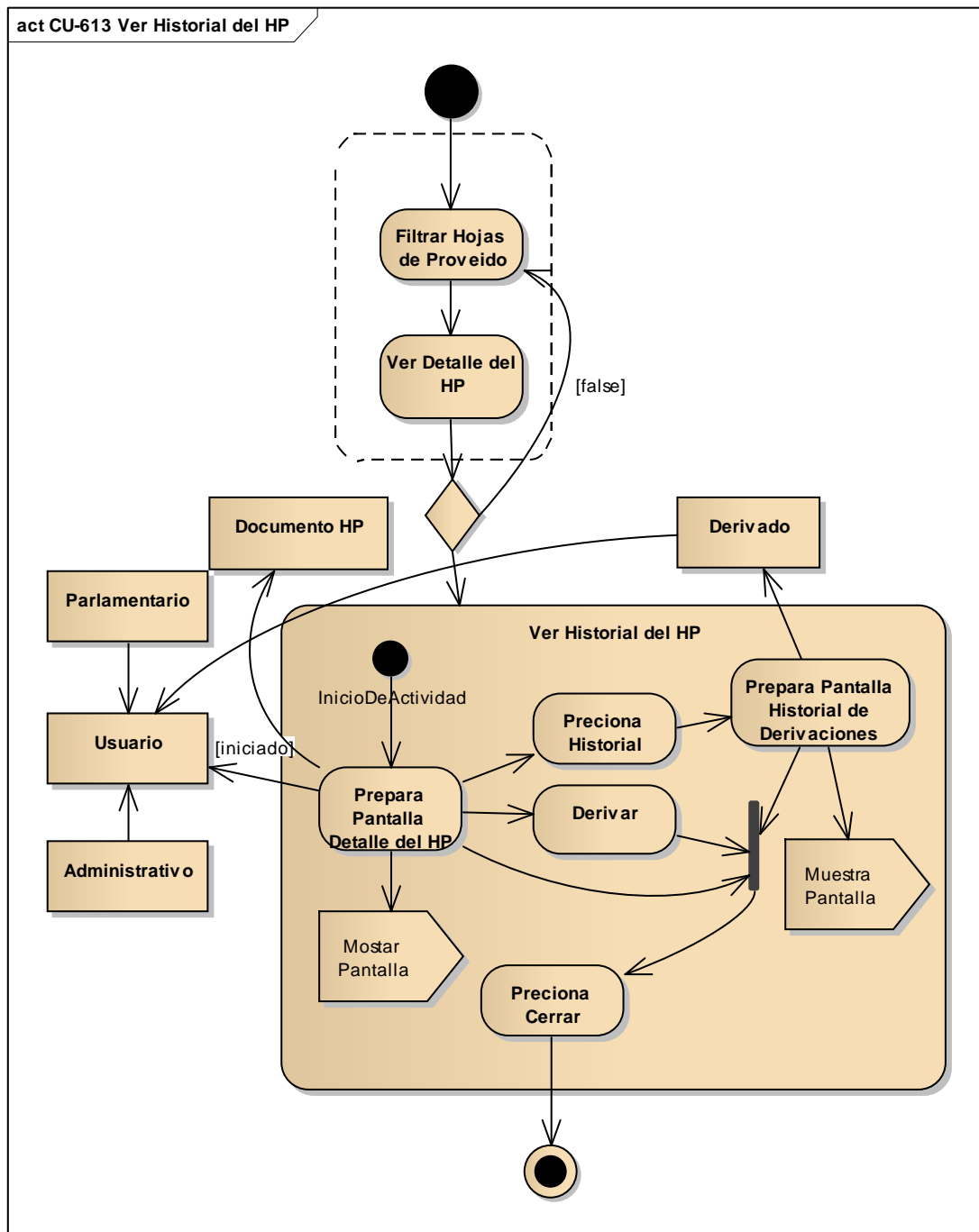


Figura 108 Diagrama de Actividad: CU-613 Ver Historial del HP

2.1.2.2.3.14.1.1.14

Diagrama de Actividad: CU-614 Derivar

HP

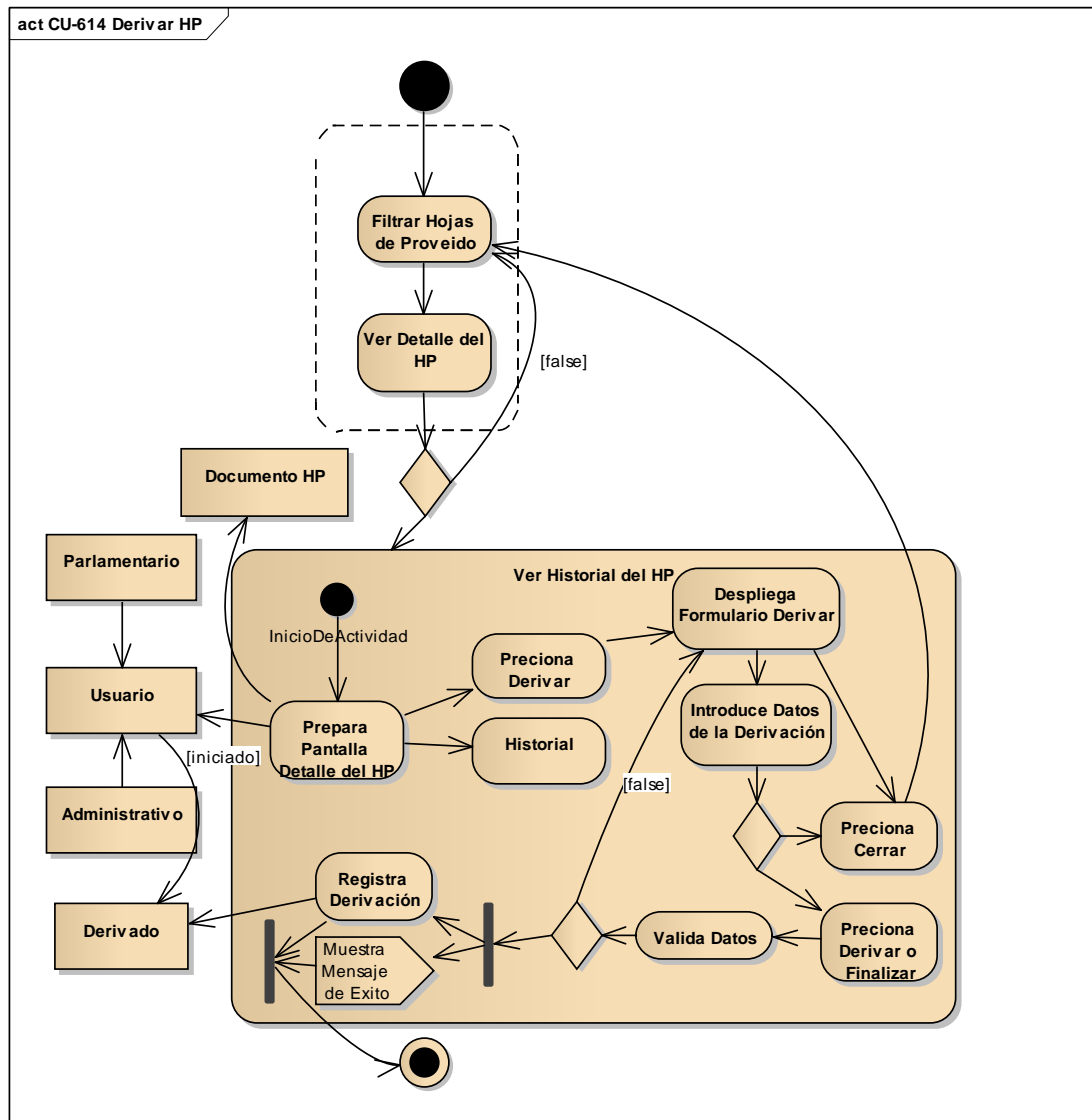


Figura 109 Diagrama de Actividad: CU-614 Derivar HP

2.1.2.2.3.14.1.15 Diagrama de Actividad: CU-612 Anular Cite de HP de HP

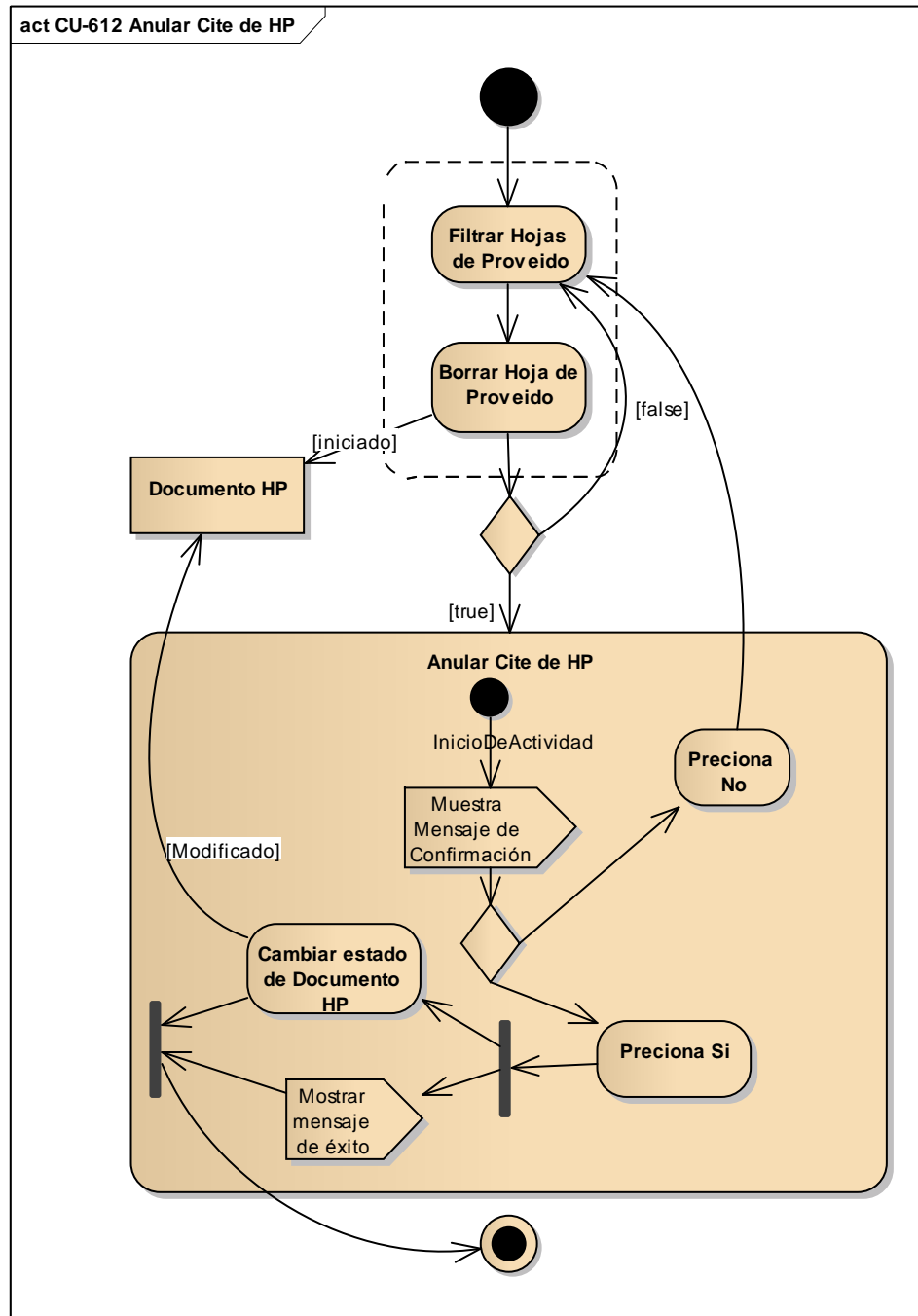


Figura 110 Diagrama de Actividad: CU-612 Anular Cite de HP

2.1.2.2.3.14.1.1.16 Diagrama de Actividad: CU-700 Generar Cite

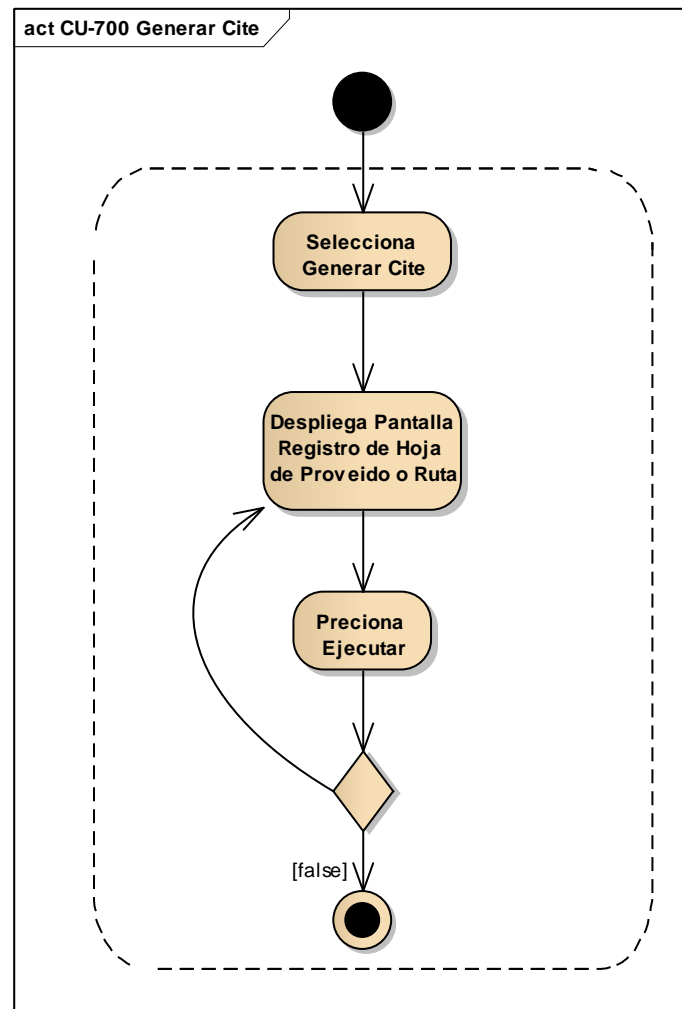


Figura 111 Diagrama de Actividad: CU-700 Generar Cite

2.1.2.2.3.14.1.1.17 CU-701 Registro de HP

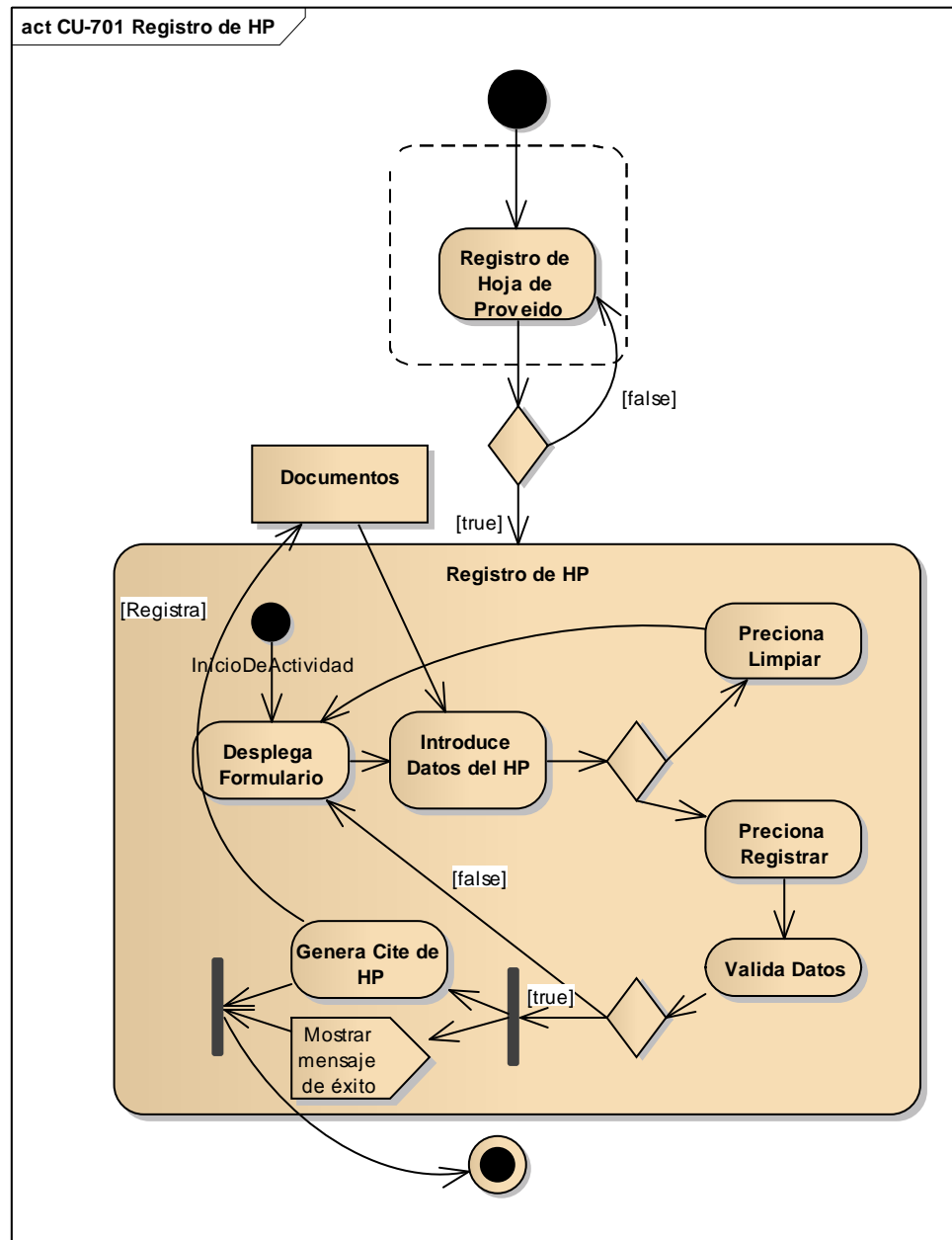


Figura 112 Diagrama de Actividad: CU-701 Registro de HP

2.1.2.2.3.14.2 Modelado de Diagrama de Secuencias

Un diagrama de secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario. En aplicaciones grandes además de los objetos se muestran también los componentes y casos de uso. El mostrar los componentes tiene sentido ya que se trata de objetos reutilizables, en cuanto a los casos de uso hay que recordar que se implementan como objetos cuyo rol es encapsular lo definido en el caso de uso.

Para mostrar la interacción con el usuario o con otro sistema se introducen en los diagramas de secuencia las **boundary classes**. En las primeras fases de diseño el propósito de introducir estas clases es capturar y documentar los requisitos de interfaz, pero no el mostrar cómo se va a implementar dicha interfaz.

Los diagramas de secuencia, formalmente diagramas de traza de eventos o de interacción de objetos, se utilizan con frecuencia para validar los casos de uso. Documentan el diseño desde el punto de vista de los casos de uso. Observando qué mensajes se envían a los objetos, componentes o casos de uso y viendo a grosso modo cuanto tiempo consume el método invocado, los diagramas de secuencia nos ayudan a comprender los cuellos de botella potenciales, para así poder eliminarlos. A la hora de documentar un diagrama de secuencia resulta importante mantener los enlaces de los mensajes a los métodos apropiados del diagrama de clases.

2.1.2.2.3.14.2.1 Diagrama de Secuencia

2.1.2.2.3.14.2.1.1 Diagrama de Secuencia: CU-001 Iniciar Sesión

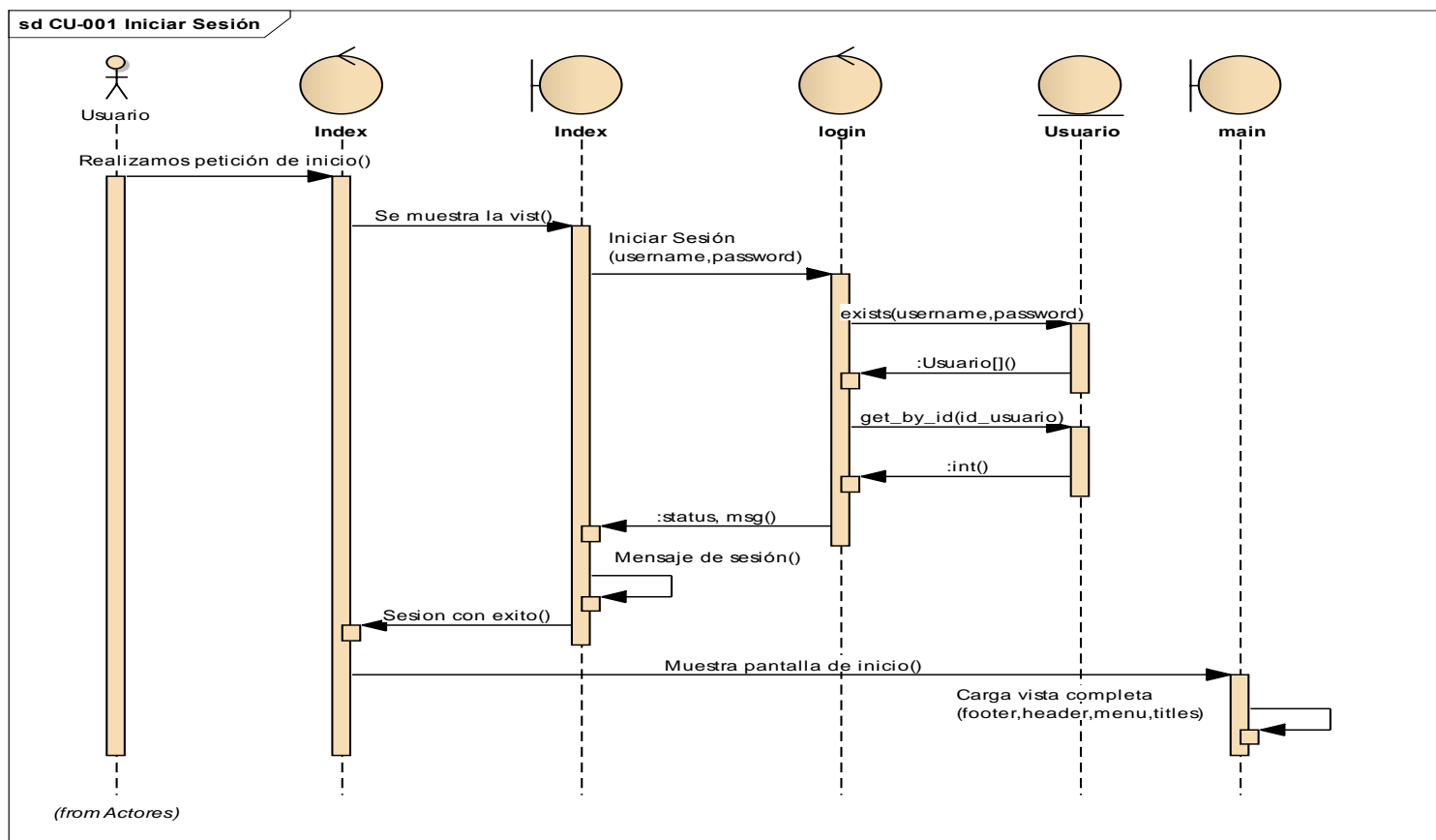


Figura 113 Diagrama de Secuencia: Caso de Uso Ingresar al Sistema

2.1.2.2.3.14.2 Diagrama de Secuencia: CU-100 Gestión Pleno

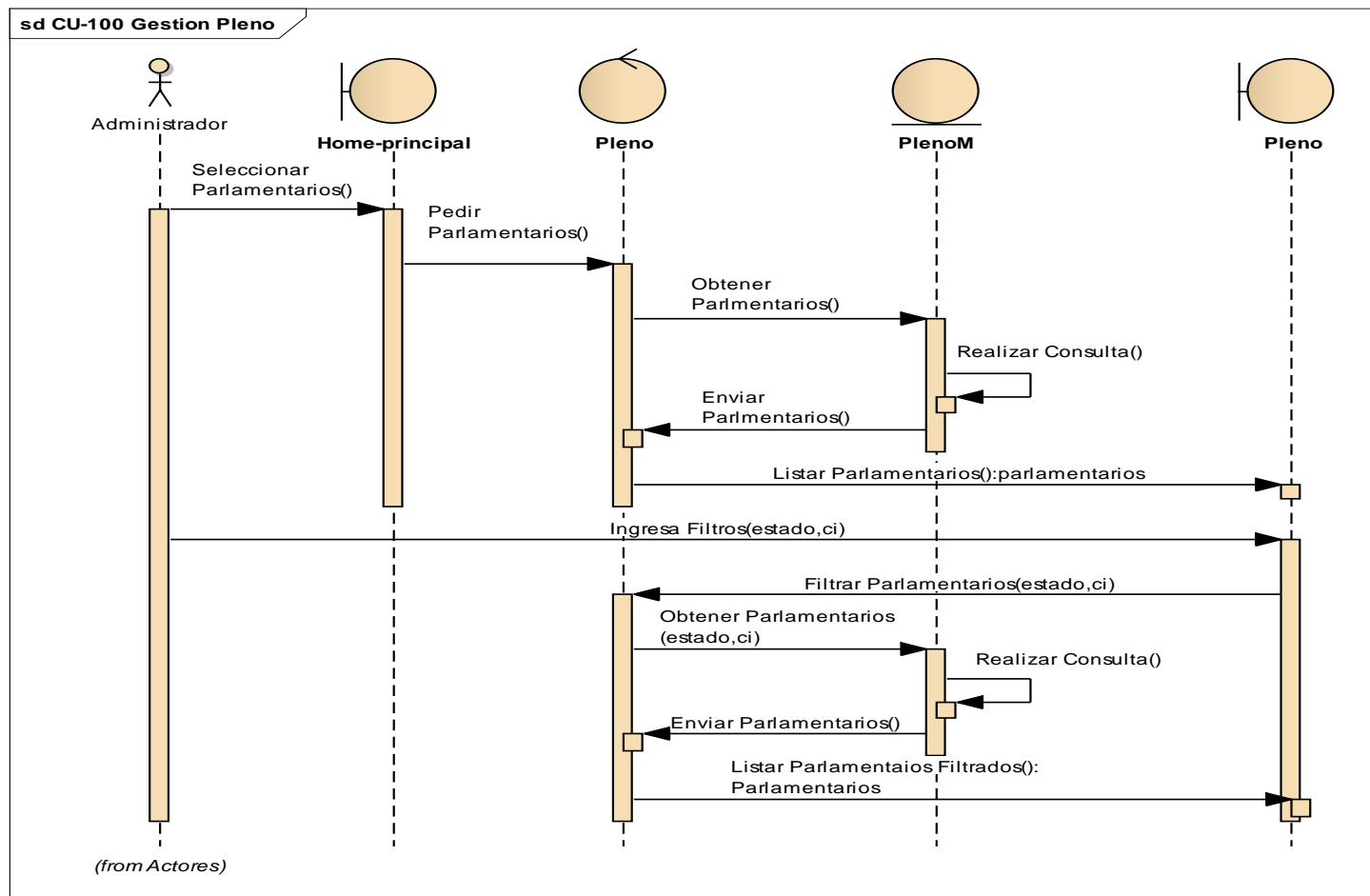


Figura 114 Diagrama de Secuencia: CU-100 Gestión Pleno

2.1.2.2.3.14.2.1.3 Diagrama de Secuencia: CU-101 Adicionar Parlamentario

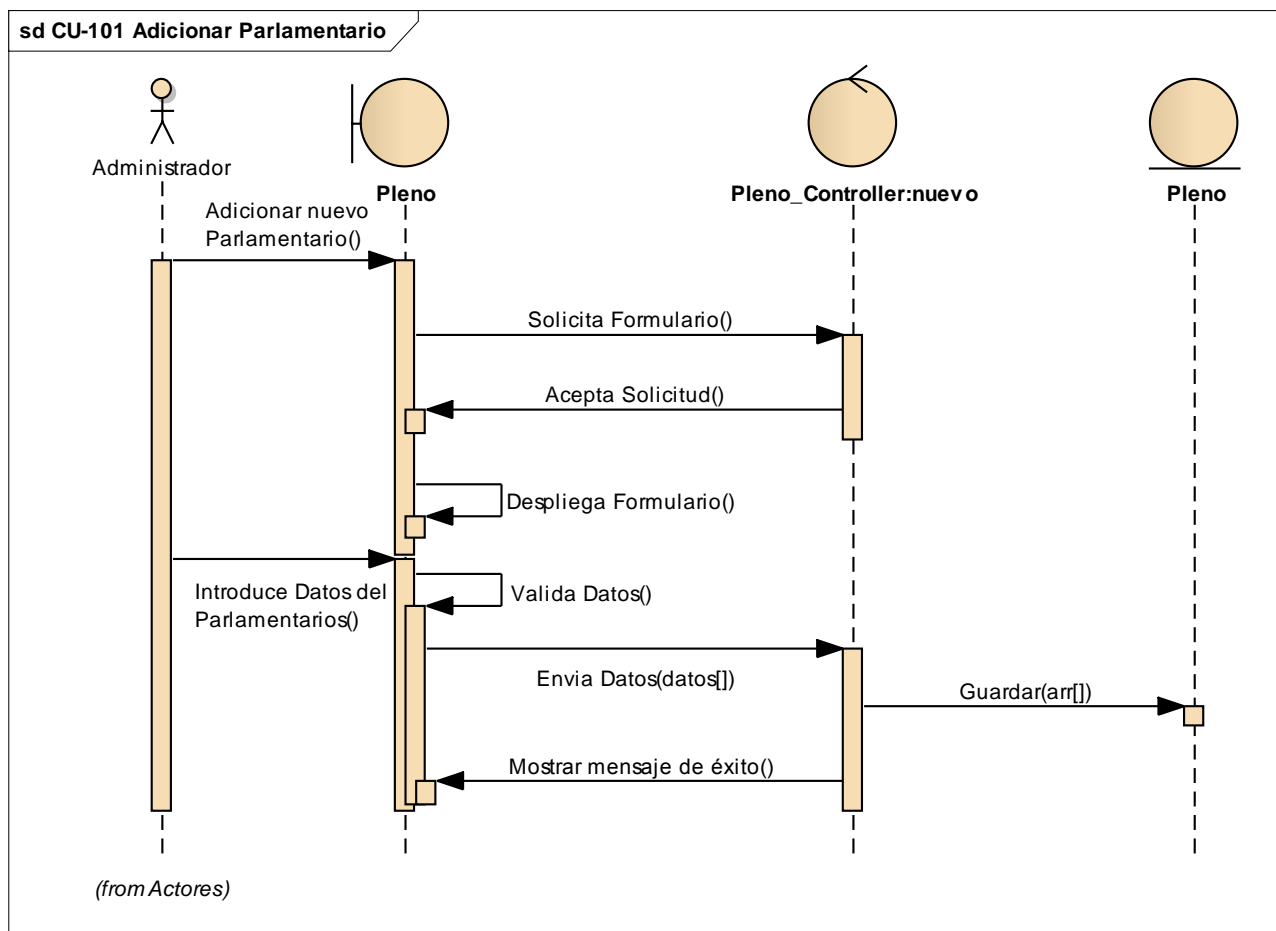


Figura 115 Diagrama de Secuencia: CU-101 Adicionar Parlamentario

2.1.2.2.3.14.2.1.4 Diagrama de Secuencia: CU-102 Modificar Parlamentario

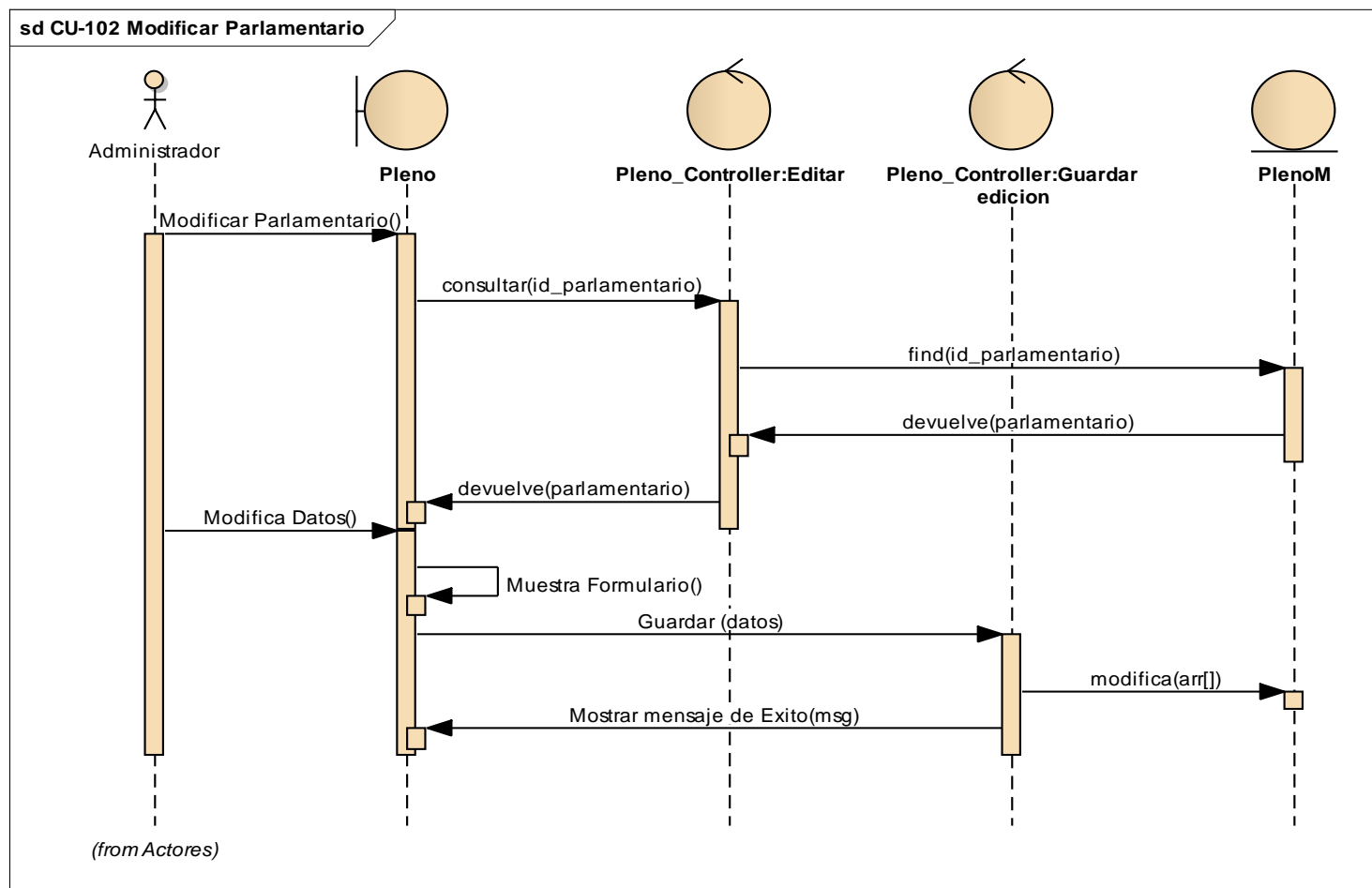


Figura 116 Diagrama de Secuencia: CU-102 Modificar Parlamentario

2.1.2.2.3.14.2.1.5 Diagrama de Secuencia: CU-103 Eliminar Parlamentario

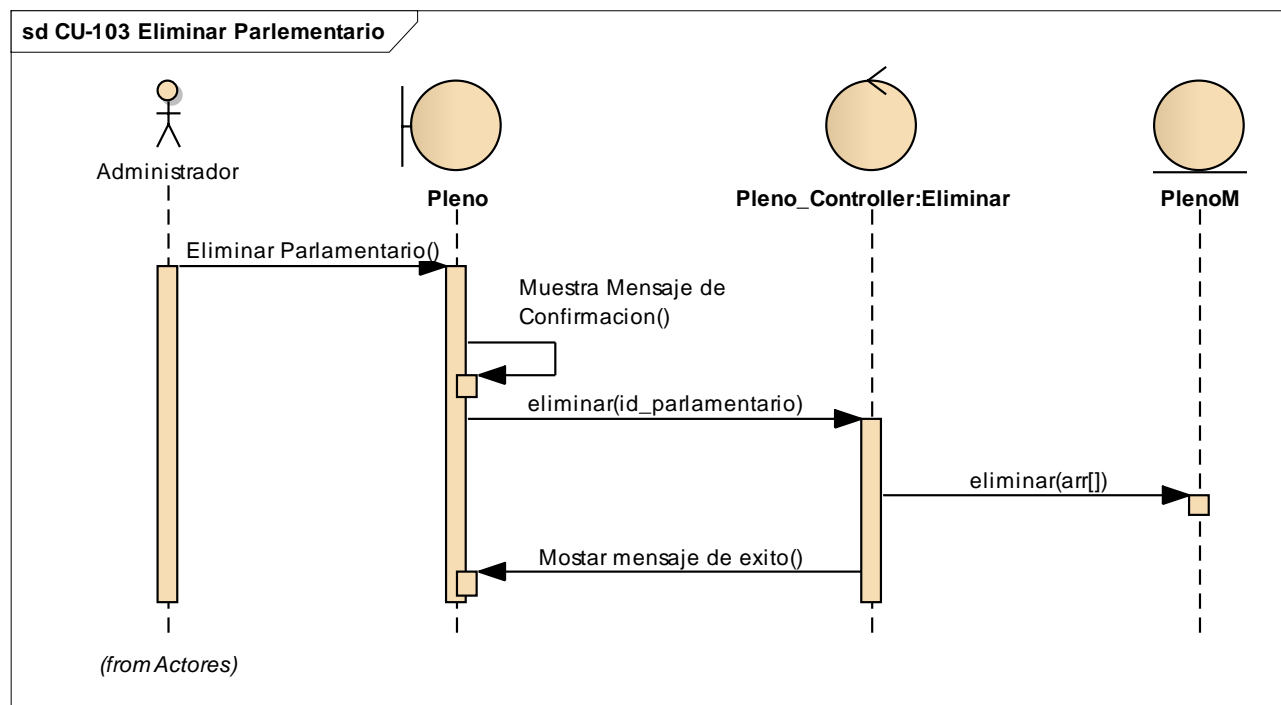


Figura 117 Diagrama de Secuencia: CU-103 Eliminar Parlamentario

2.1.2.2.3.14.2.1.6 Diagrama de Secuencia: CU-200 Gestión Directiva

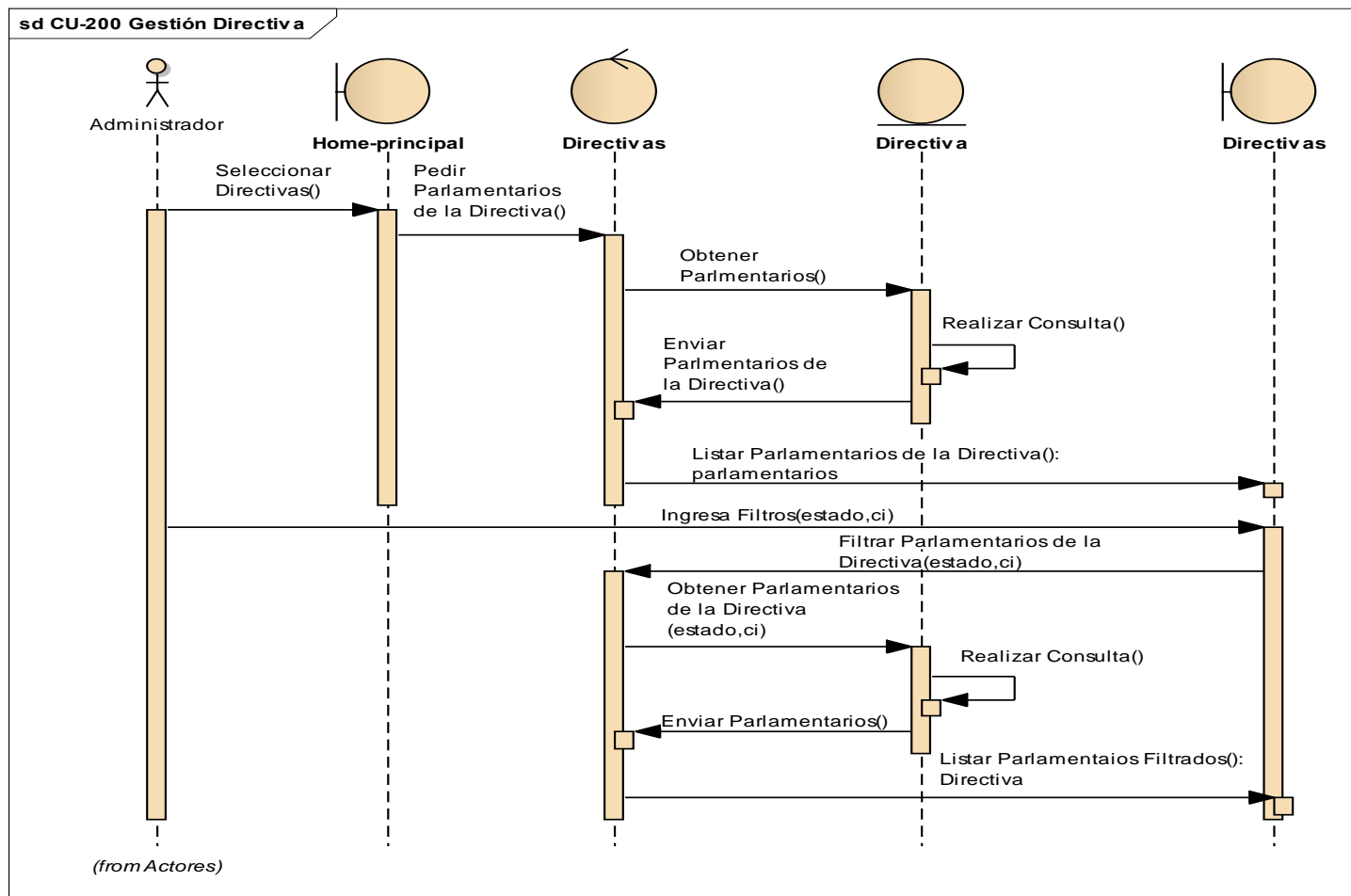


Figura 118 Diagrama de Secuencia: CU-200 Gestión Directiva

2.1.2.2.3.14.2.1.7 Diagrama de Secuencia: CU-201 Adicionar Directiva

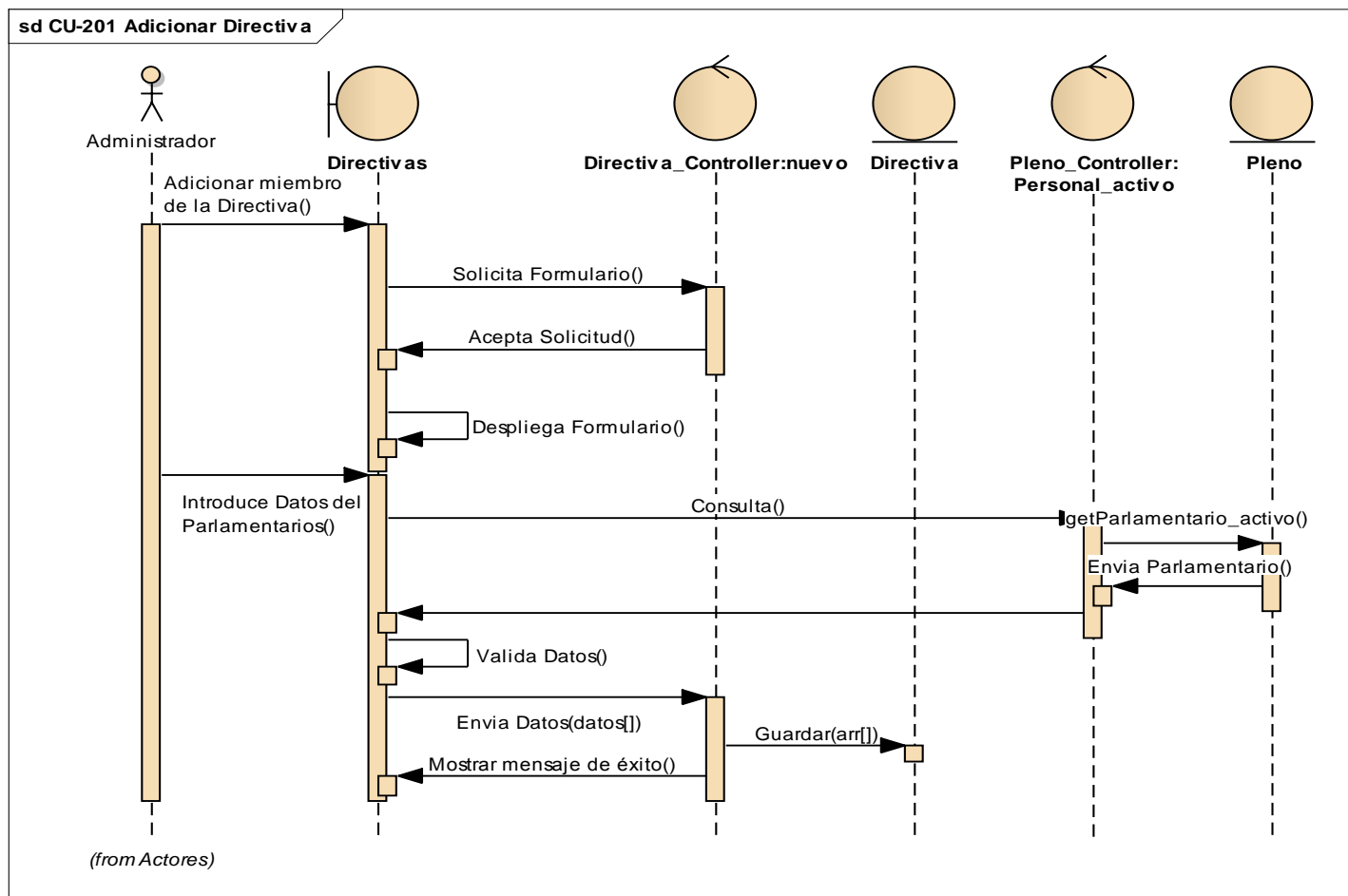


Figura 119 Diagrama de Secuencia: CU-201 Adicionar Directiva

2.1.2.2.3.14.2.1.8 Diagrama de Secuencia: CU-202 Modificar Directiva

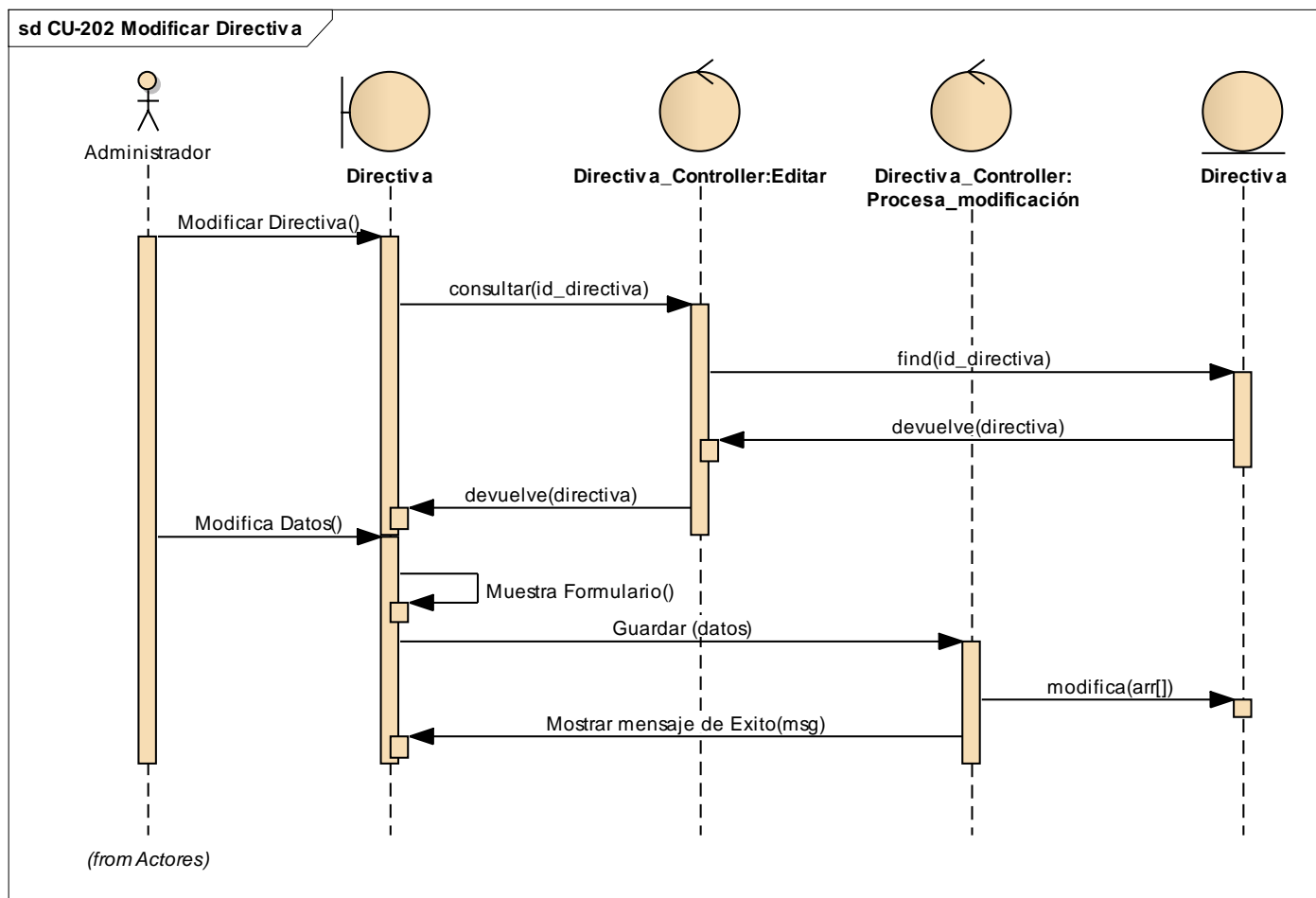


Figura 120 Diagrama de Secuencia: CU-202 Modificar Directiva

2.1.2.2.3.14.2.1.9 Diagrama de Secuencia: CU-203 Eliminar Directiva

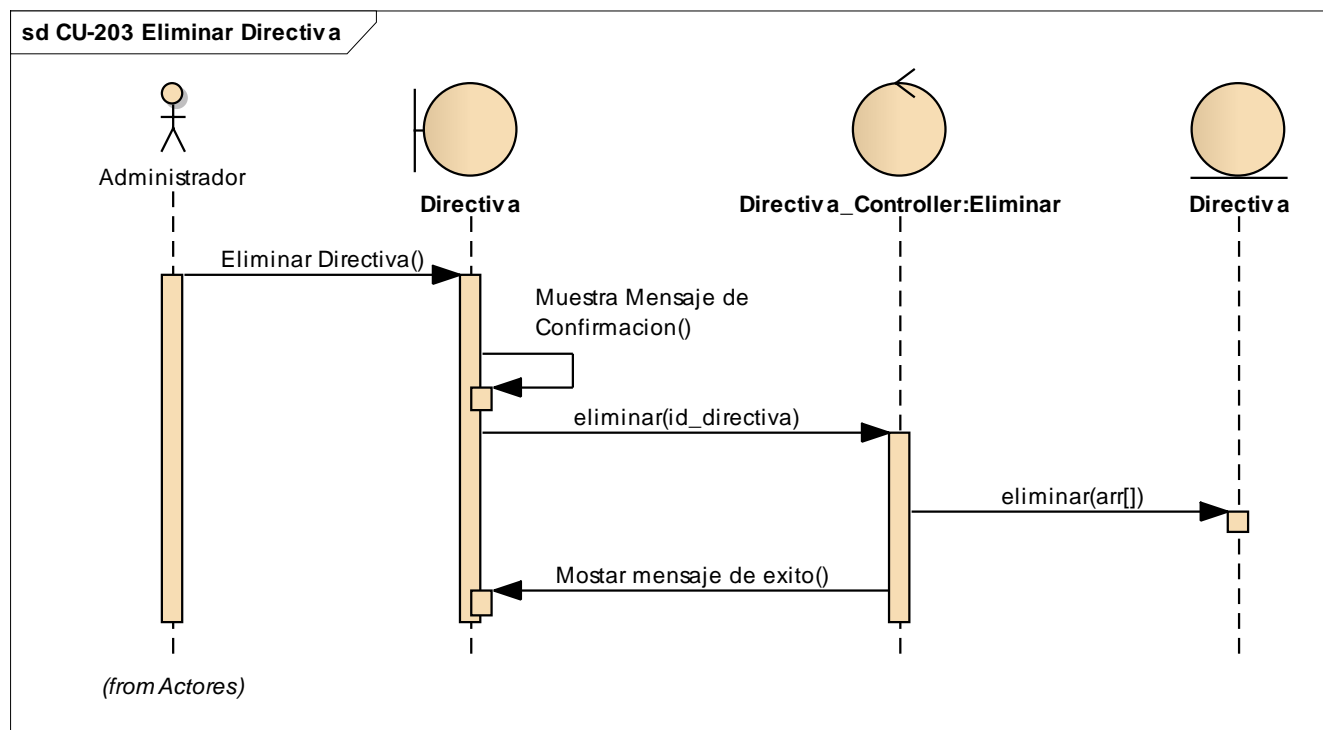


Figura 121 Diagrama de Secuencia: CU-203 Eliminar Directiva

2.1.2.2.3.14.2.1.10 Diagrama de Secuencia: CU-300 Gestión Comisiones

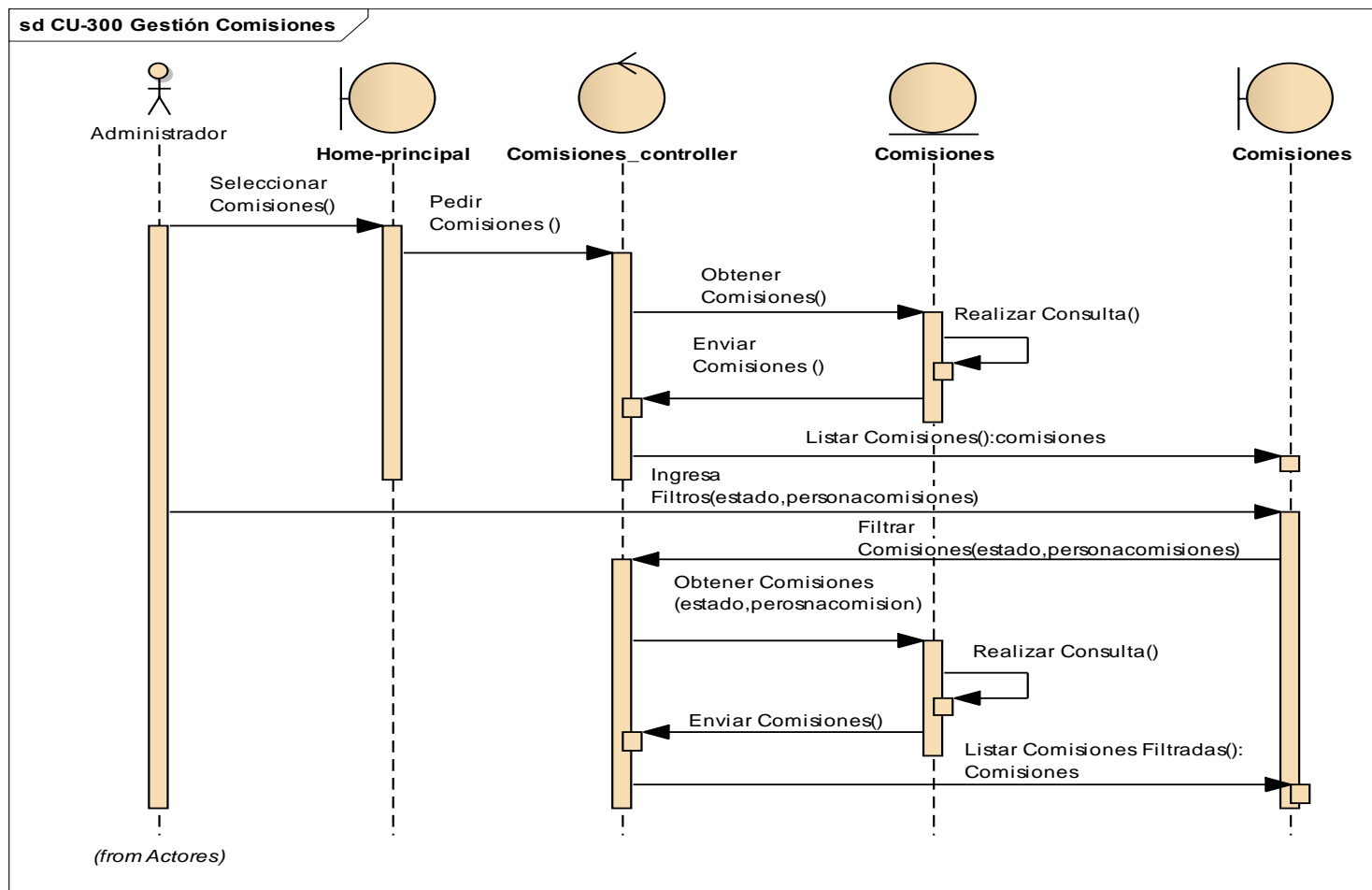


Figura 122 Diagrama de Secuencia: CU-300 Gestión Comisiones

2.1.2.2.3.14.2.1.11 Diagrama de Secuencia: CU-301 Adicionar Comisión

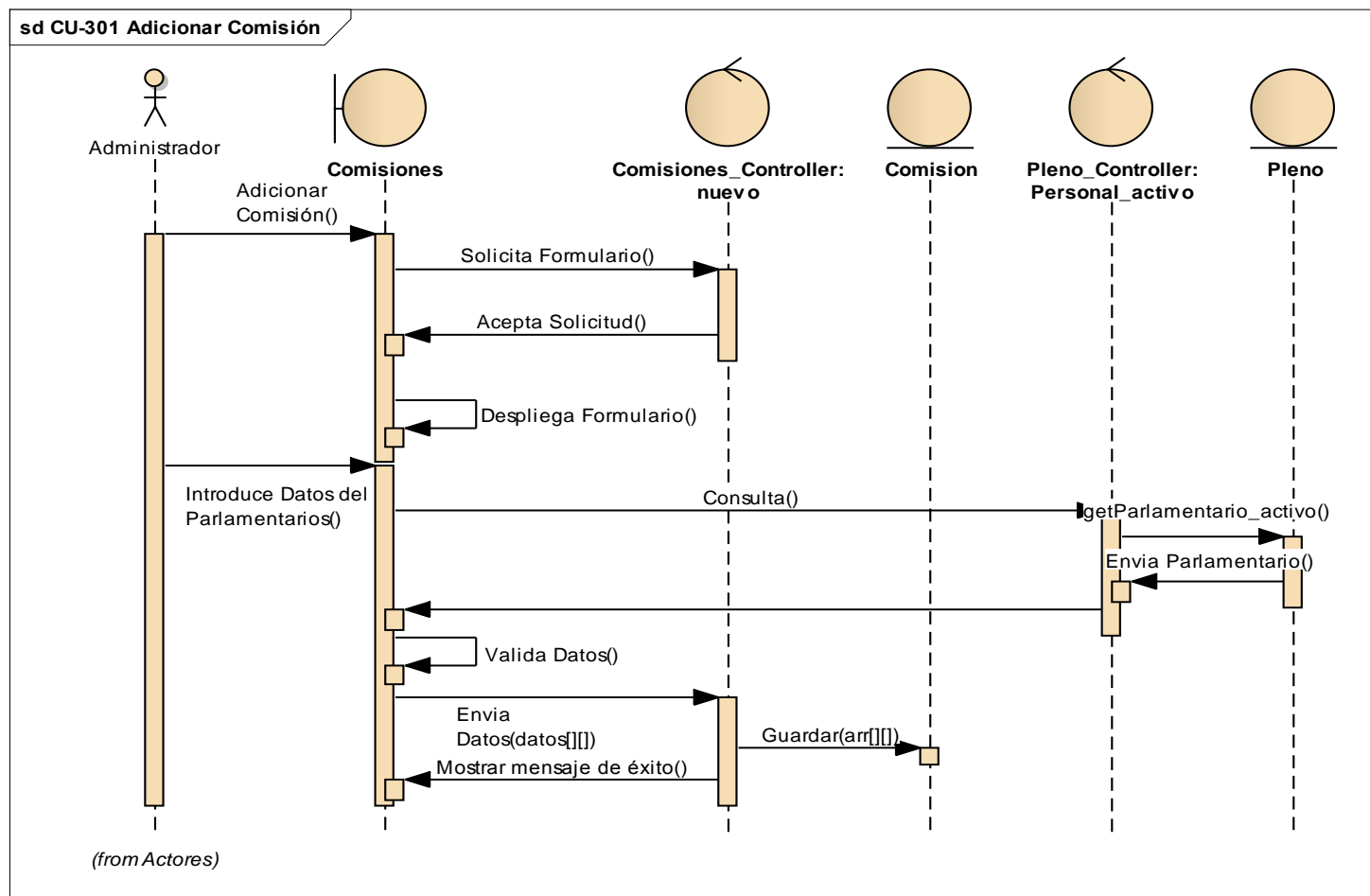


Figura 123 Diagrama de Secuencia: CU-301 Adicionar Comisión

2.1.2.2.3.14.2.1.12 Diagrama de Secuencia: CU-302 Ver Detalle Comisión

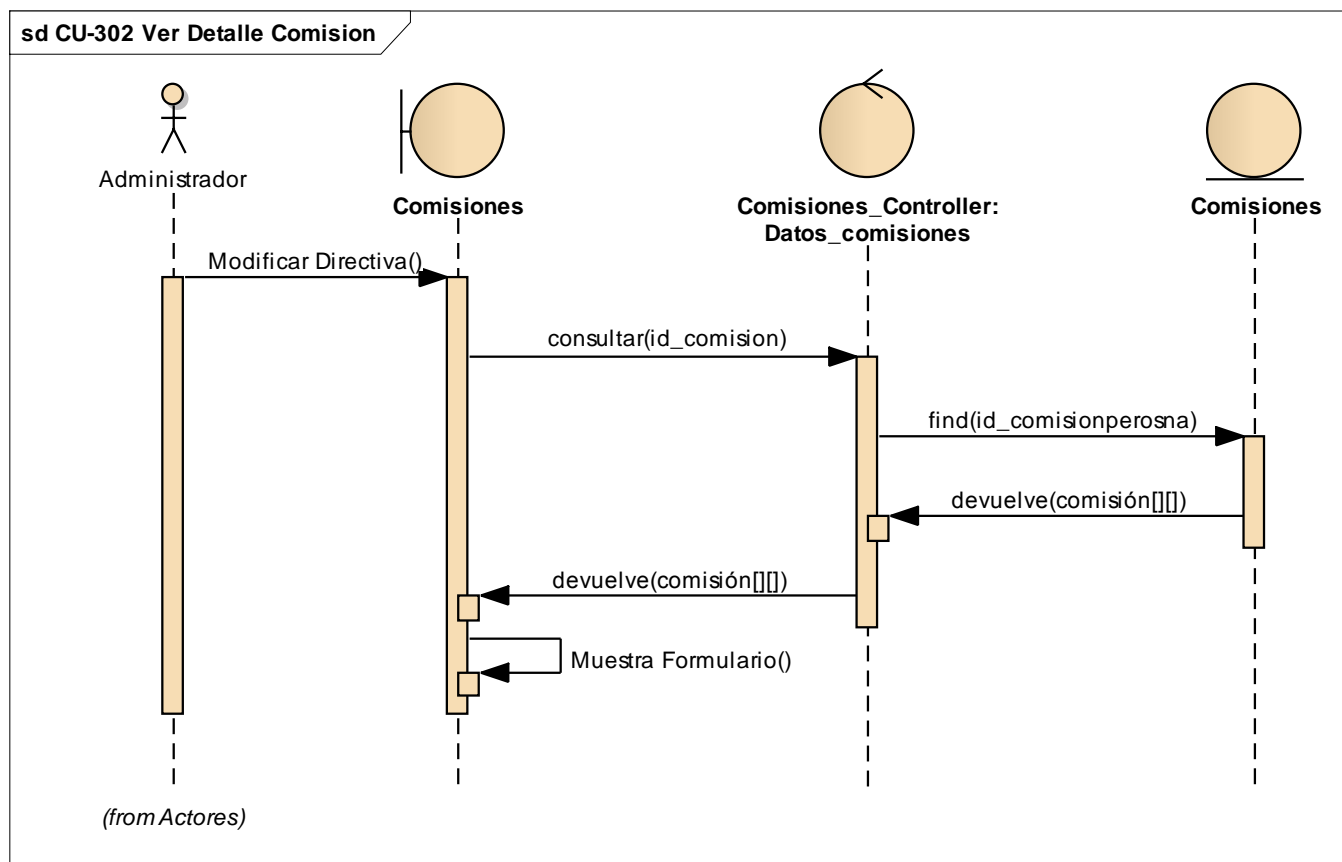


Figura 124 Diagrama de Secuencia: CU-302 Ver Detalle Comisión

2.1.2.2.3.14.2.1.13 Diagrama de Secuencia: CU-303 Eliminar Comisión

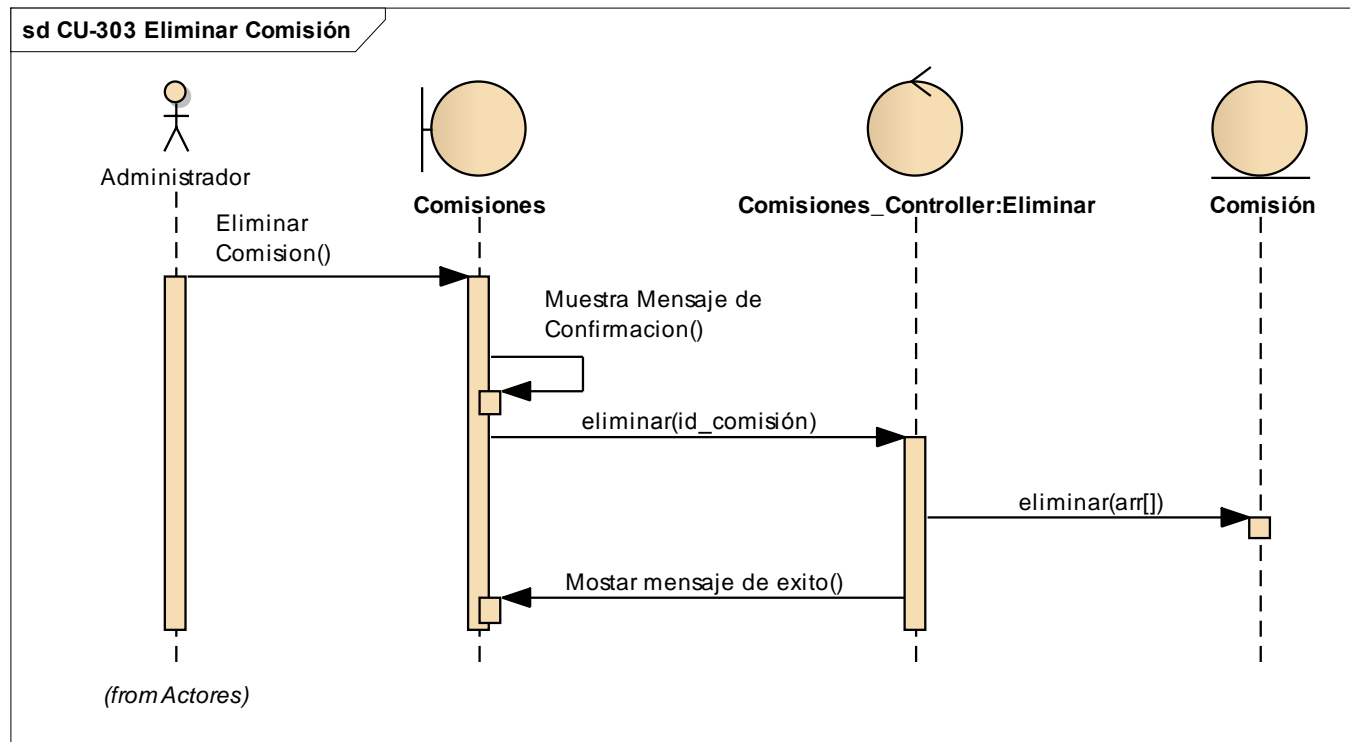


Figura 125 Diagrama de Secuencia: CU-303 Eliminar Comisión

2.1.2.2.3.14.2.1.14 Diagrama de Secuencia: CU-400 Gestión Administrativos

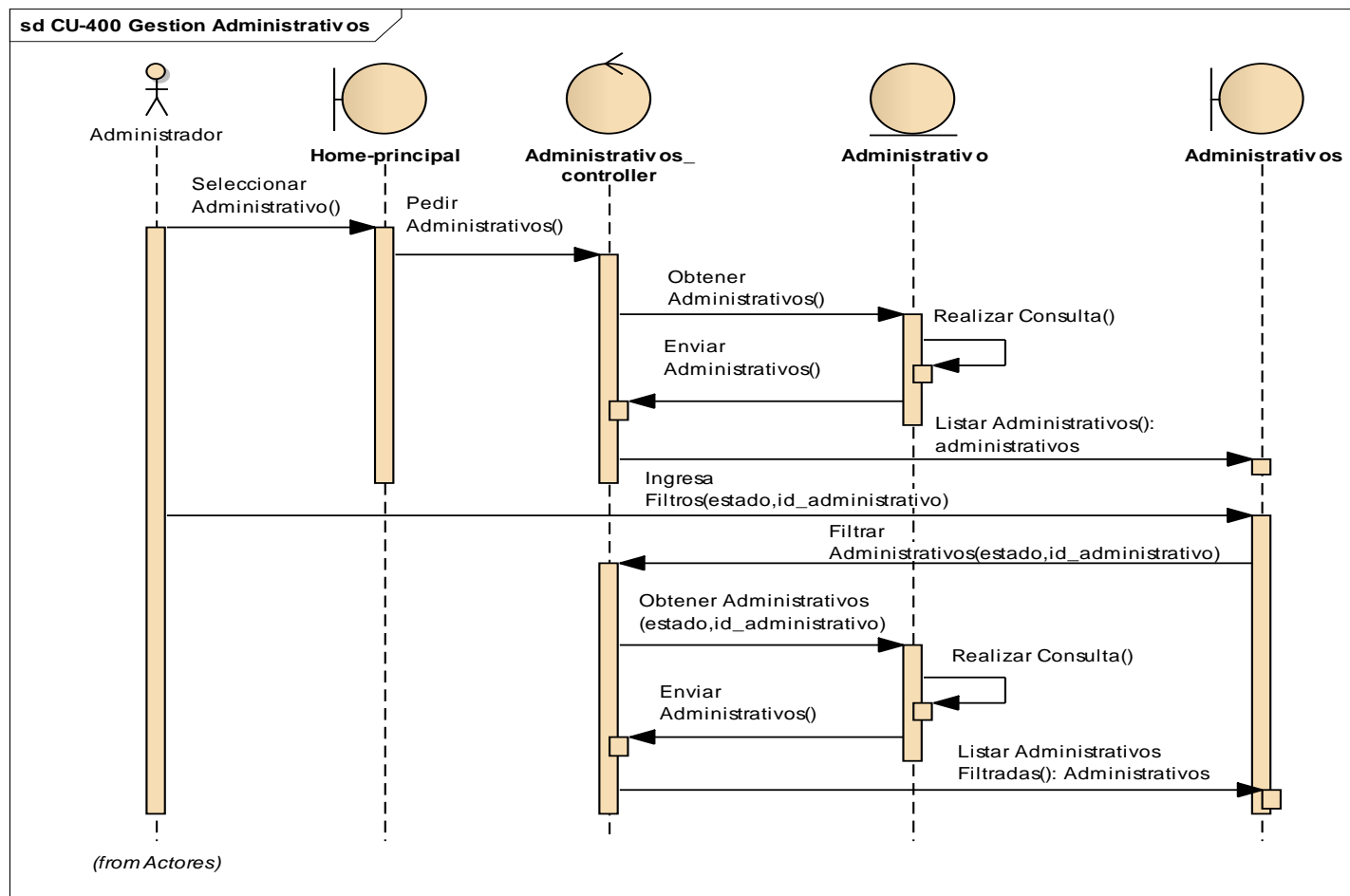


Figura 126 Diagrama de Secuencia: CU-400 Gestión Administrativos

2.1.2.2.3.14.2.1.15 Diagrama de Secuencia: CU-401 Adicionar Administrativo

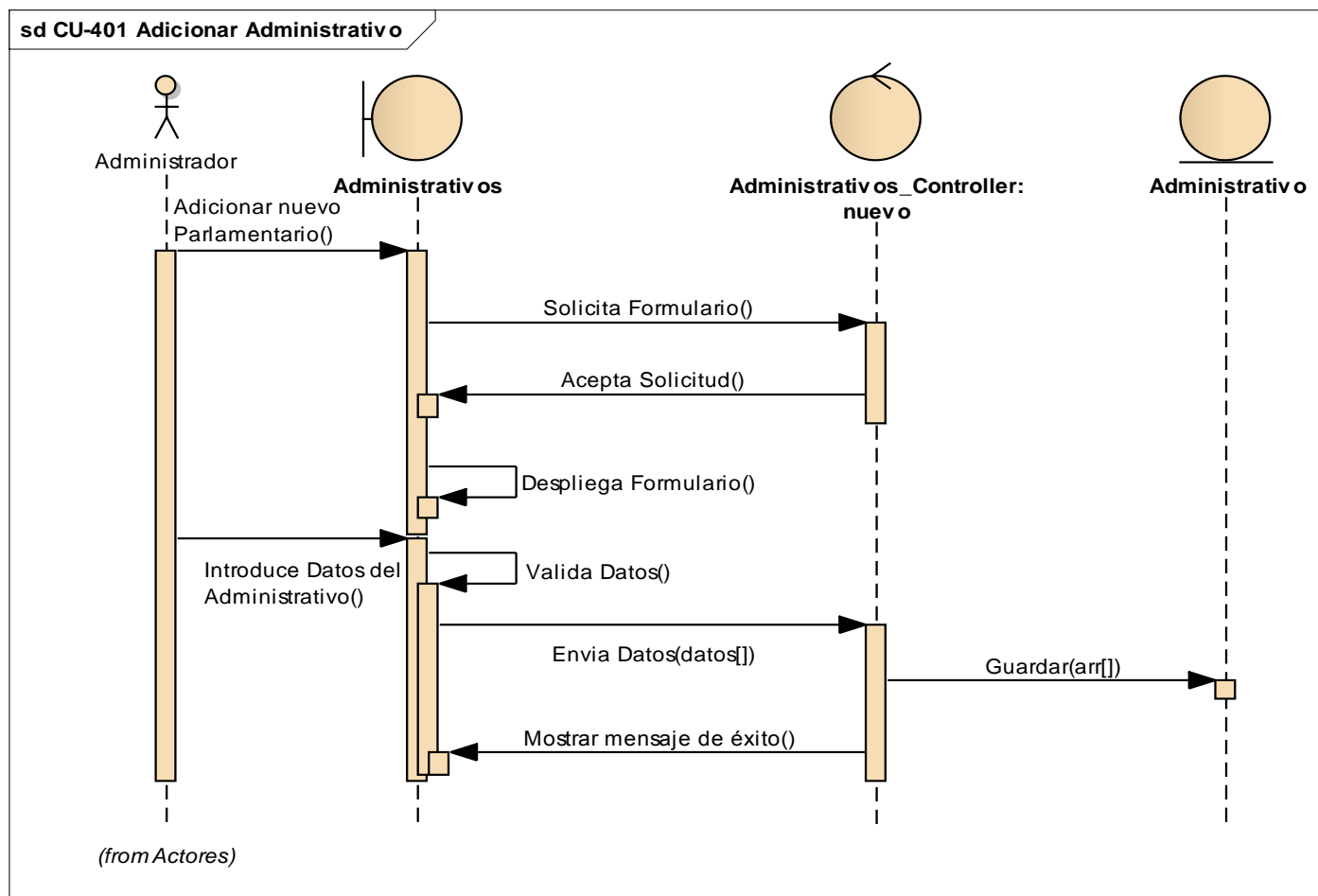


Figura 127 Diagrama de Secuencia: CU-401 Adicionar Administrativo

2.1.2.2.3.14.2.1.16 Diagrama de Secuencia: CU-402 Modificar Administrativo

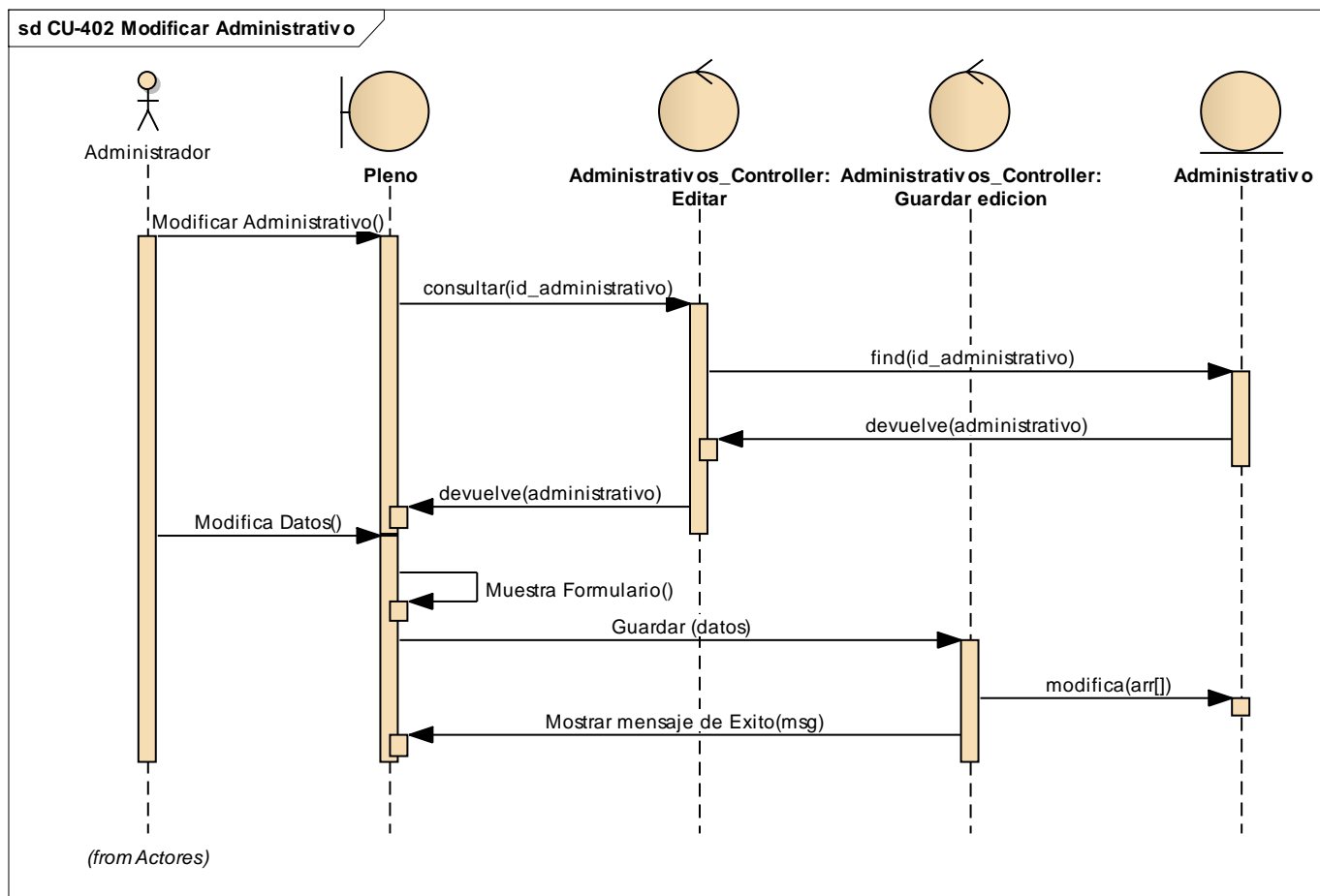


Figura 128 Diagrama de Secuencia: CU-402 Modificar Administrativo

2.1.2.2.3.14.2.1.17 Diagrama de Secuencia: CU-403 Eliminar Administrativo

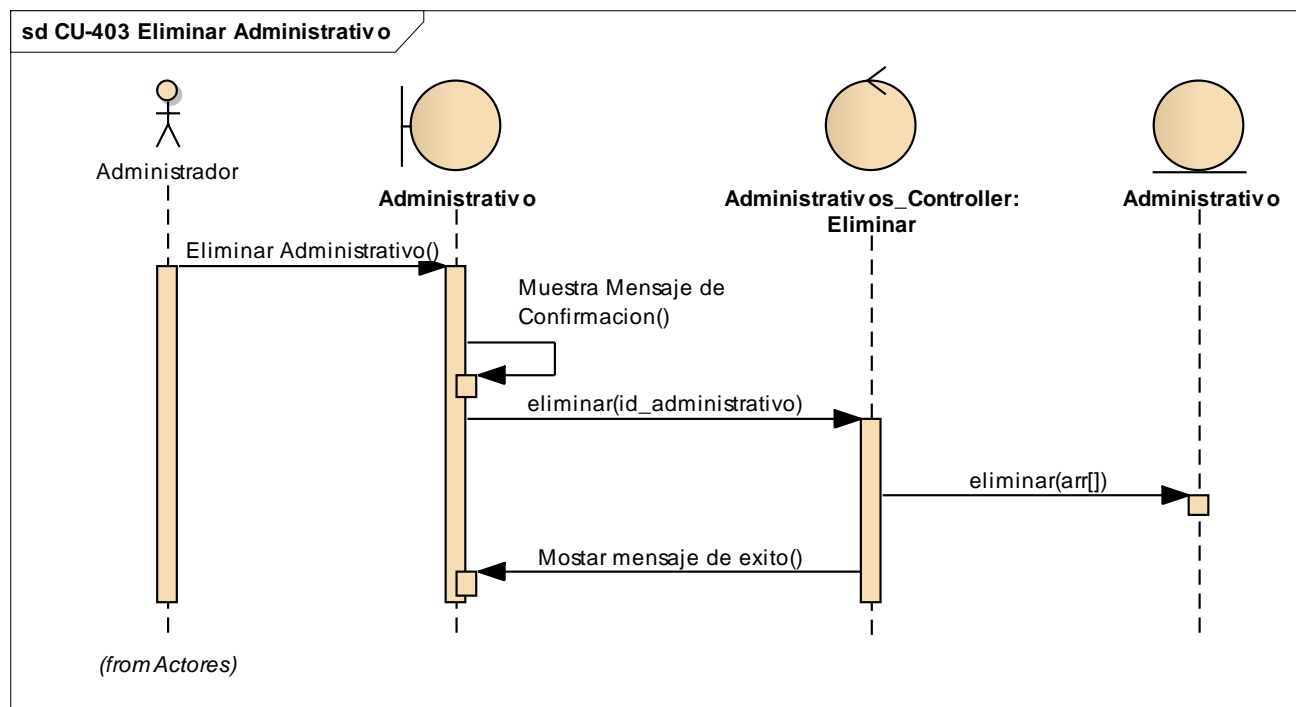


Figura 129 Diagrama de Secuencia: CU-403 Eliminar Administrativo

2.1.2.2.3.14.2.1.18 Diagrama de Secuencia: CU-500 Gestión Usuarios

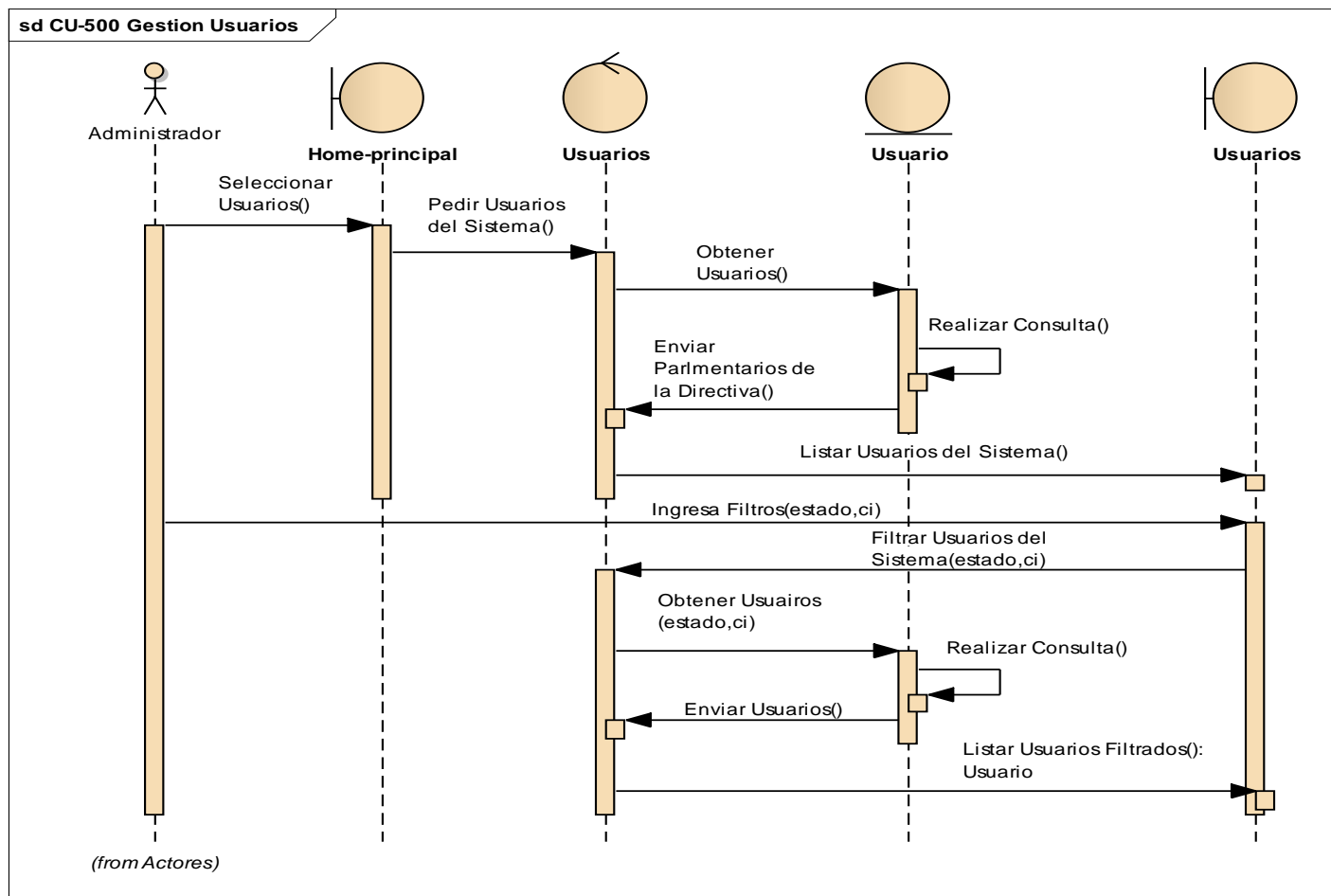


Figura 130 Diagrama de Secuencia: CU-500 Gestión Usuarios

2.1.2.2.3.14.2.1.19 Diagrama de Secuencia: CU-501 Adicionar Usuario

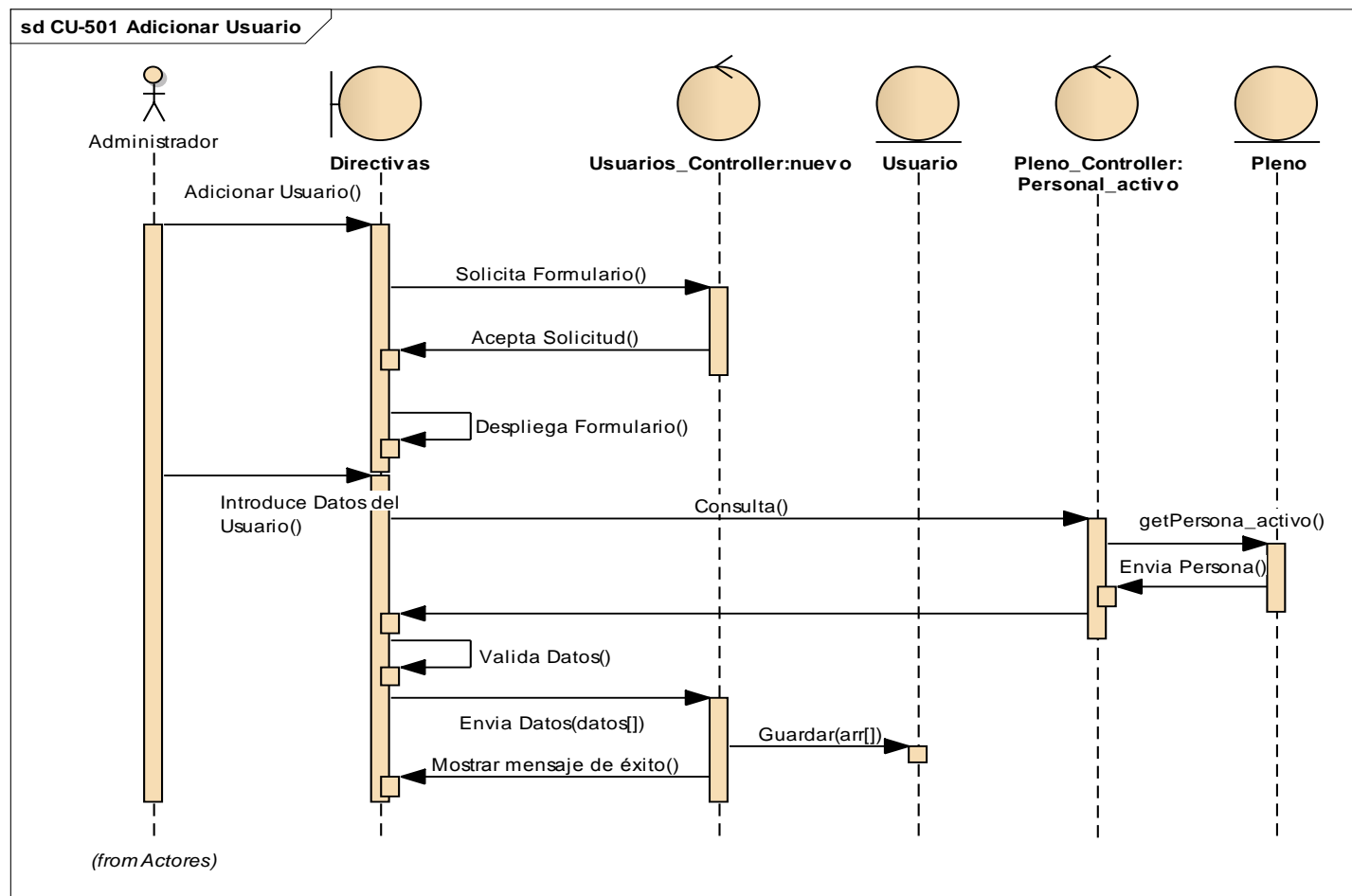
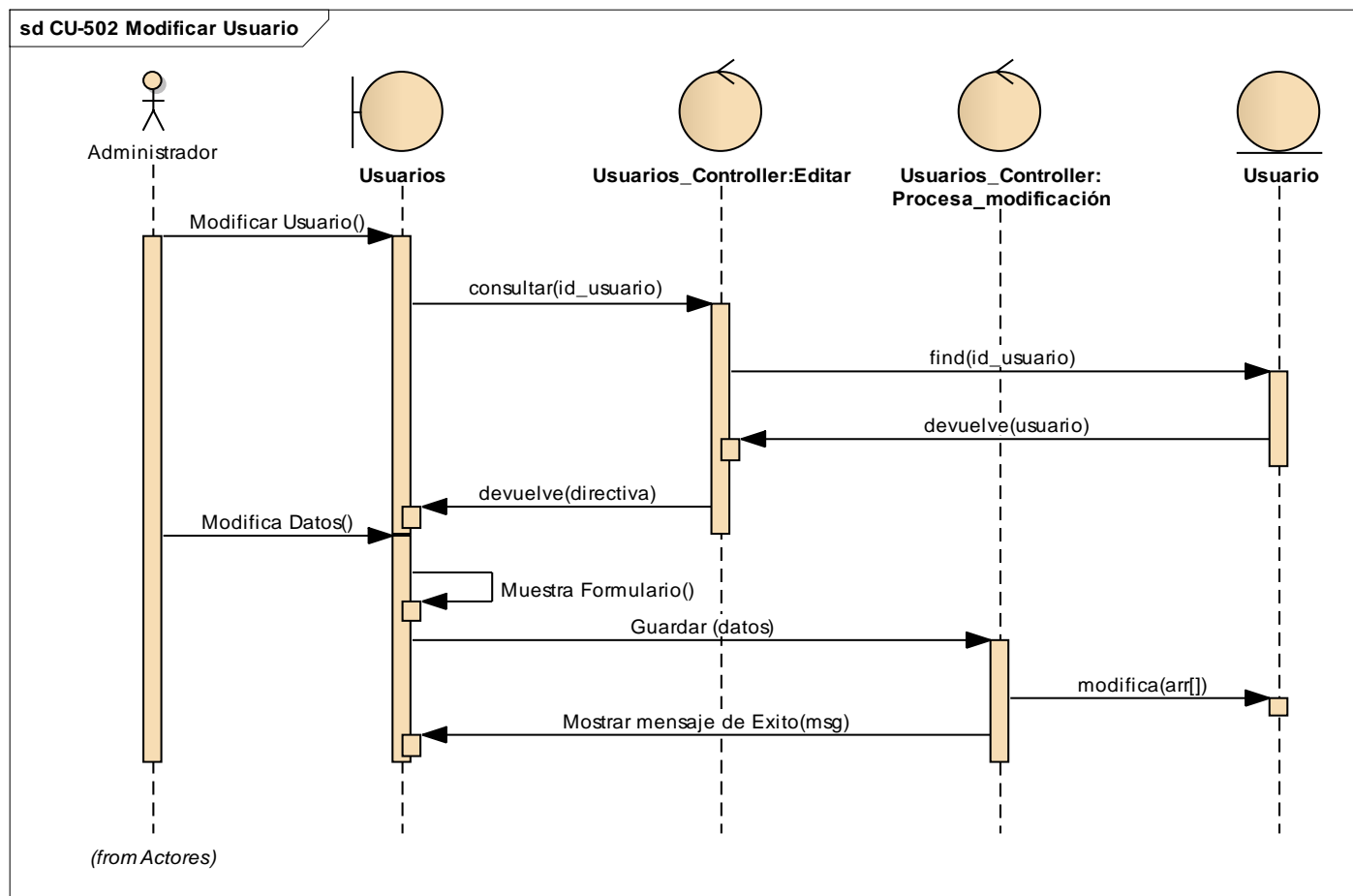


Figura 131 Diagrama de Secuencia: CU-501 Adicionar Usuario

2.1.2.2.3.14.2.1.20 Diagrama de Secuencia: CU-502 Modificar Usuario**Figura 132 Diagrama de Secuencia: CU-502 Modificar Usuario**

2.1.2.2.3.14.2.1.21 Diagrama de Secuencia: CU-504 Eliminar Usuario

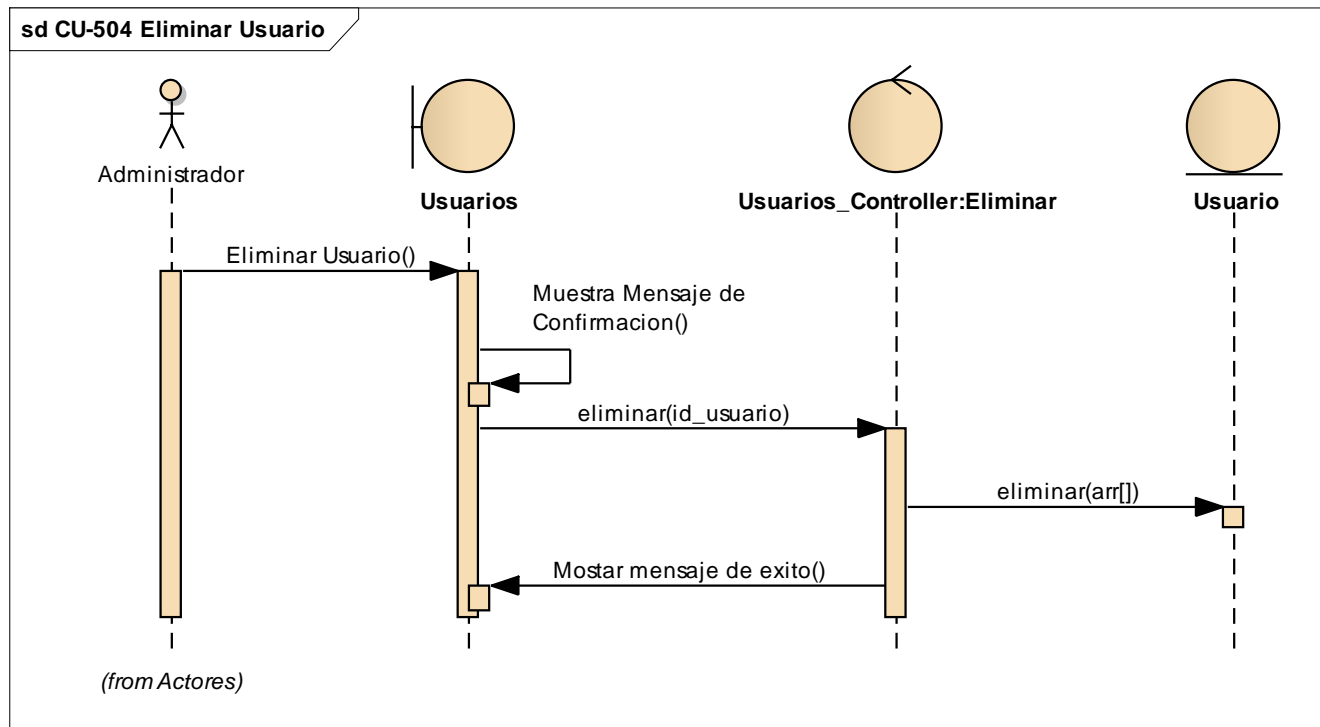


Figura 133 Diagrama de Secuencia: CU-504 Eliminar Usuario

2.1.2.2.3.14.2.1.22 Diagrama de Secuencia: CU-701 Registro de HP

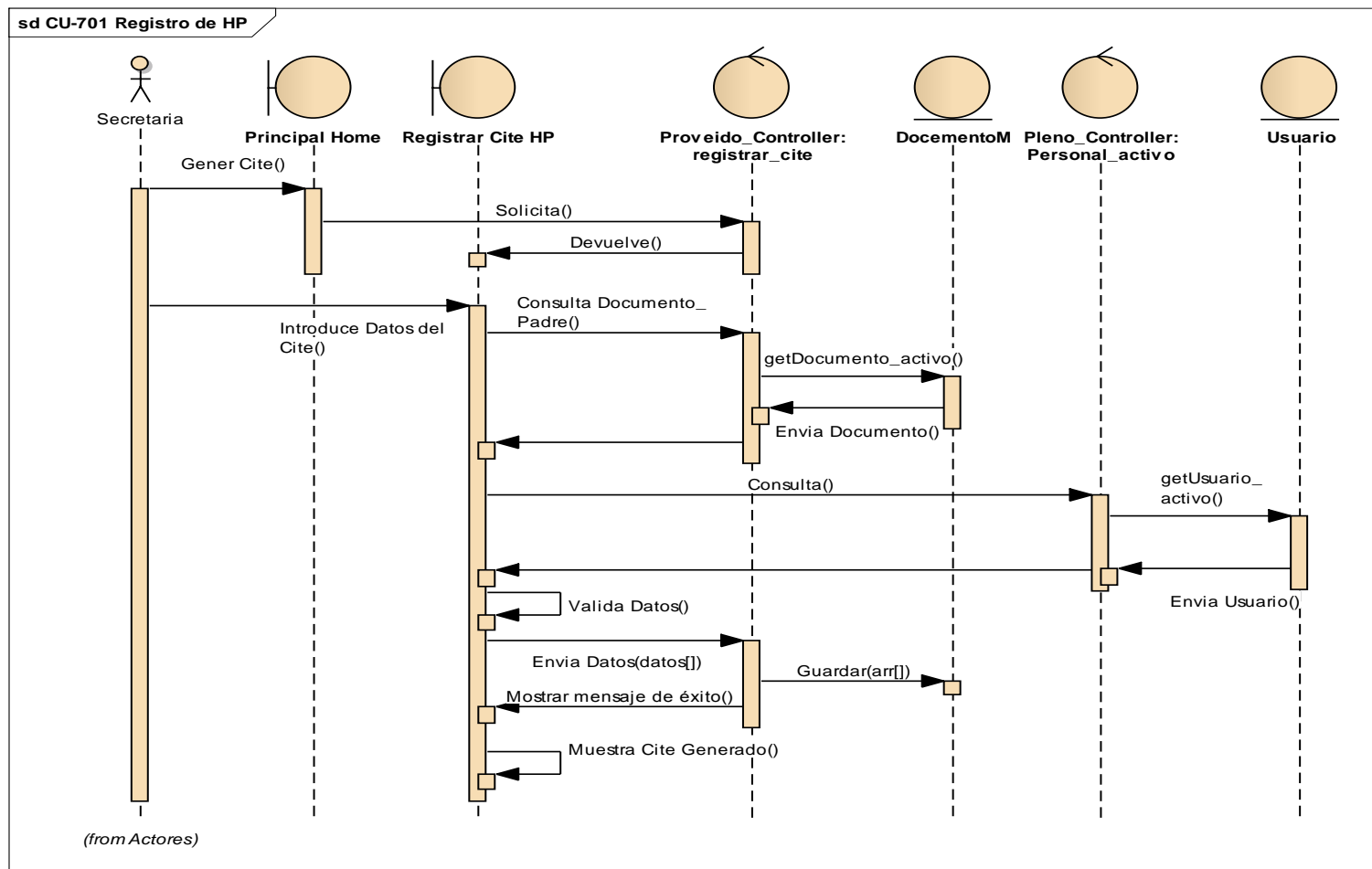


Figura 134 Diagrama de Secuencia: CU-701 Registro de HP

2.1.2.2.3.14.2.1.23

Diagrama de Secuencia: CU-702 Registro de HR

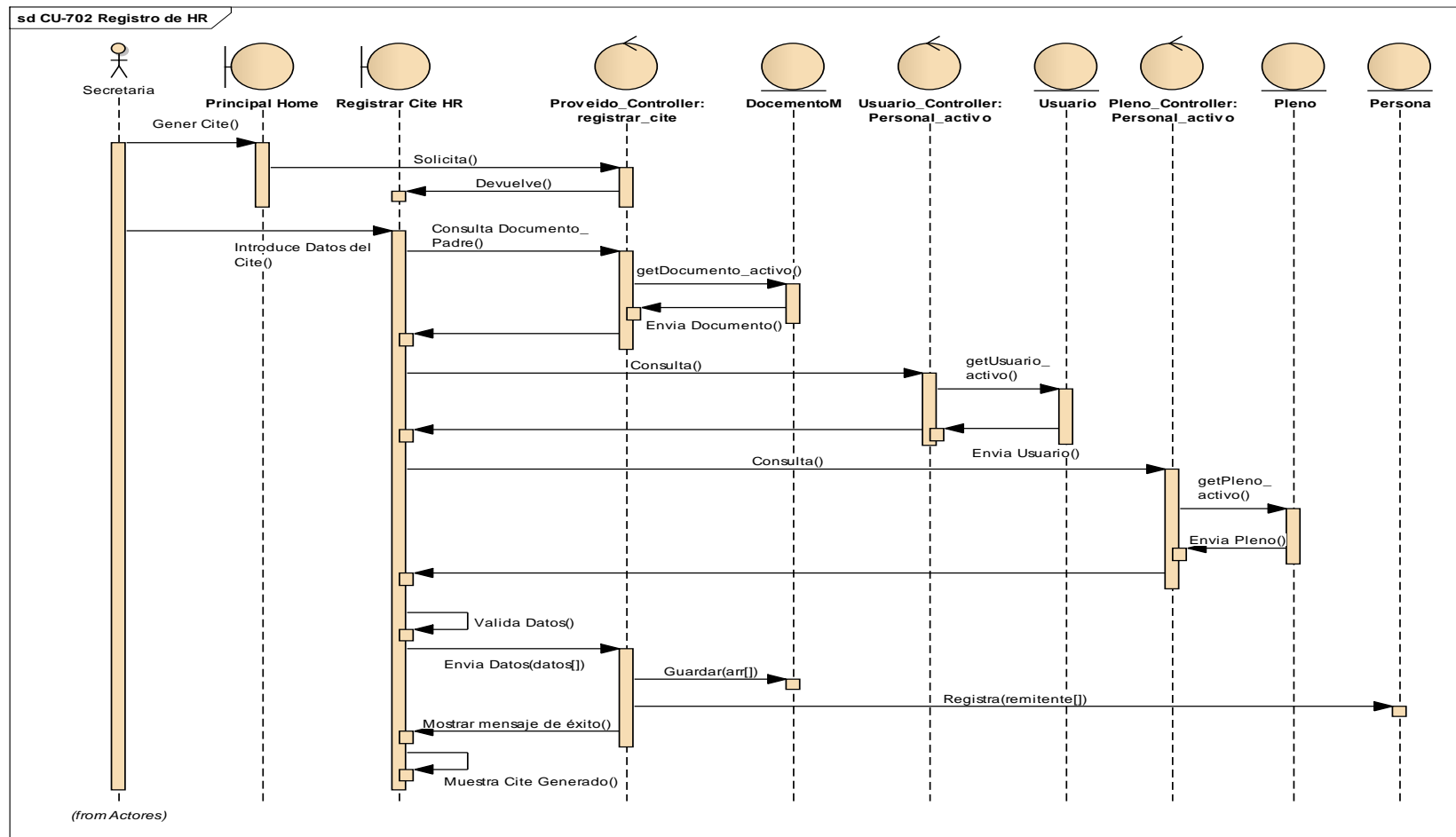


Figura 135 Diagrama de Secuencia: CU-702 Registro de HR

2.1.2.2.3.15 Modelo de Datos

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases (donde se utiliza un profile UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.).

2.1.2.2.3.15.1 Modelado de Diagrama de Clases

2.1.2.2.3.15.1.1 Introducción

El diagrama de clases es el diagrama principal para el análisis y diseño. Un diagrama de clases representa las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.

2.1.2.2.3.15.1.2 Mecanismos de Abstracción

- Clasificación / Instanciación
- Composición / Descomposición
- Agrupación / Individualización
- Especialización / Generalización

La clasificación es una de los mecanismos de abstracción más utilizados. La clase define el ámbito de definición de un conjunto de objetos, y cada objeto pertenece a una clase. Los objetos se crean por instanciación de las clases.

Cada clase se representa en un rectángulo con tres compartimientos.

- Nombre de la clase
- Atributos de la clase

- Operaciones de la clase

Los atributos de una clase no deberían ser manipulables directamente por el resto de objetos. Por esta razón se crearon niveles de visibilidad para los elementos que son:

- (-) Privado: es el más fuerte. Esta parte es totalmente invisible.
- (#) Los atributos u operaciones protegidas están visibles para las clases de herencia.
- (+) Los atributos u operaciones públicas son visibles desde otras clases y también por clases de herencia.

2.1.2.2.3.15.1.3 Diagrama de Clases

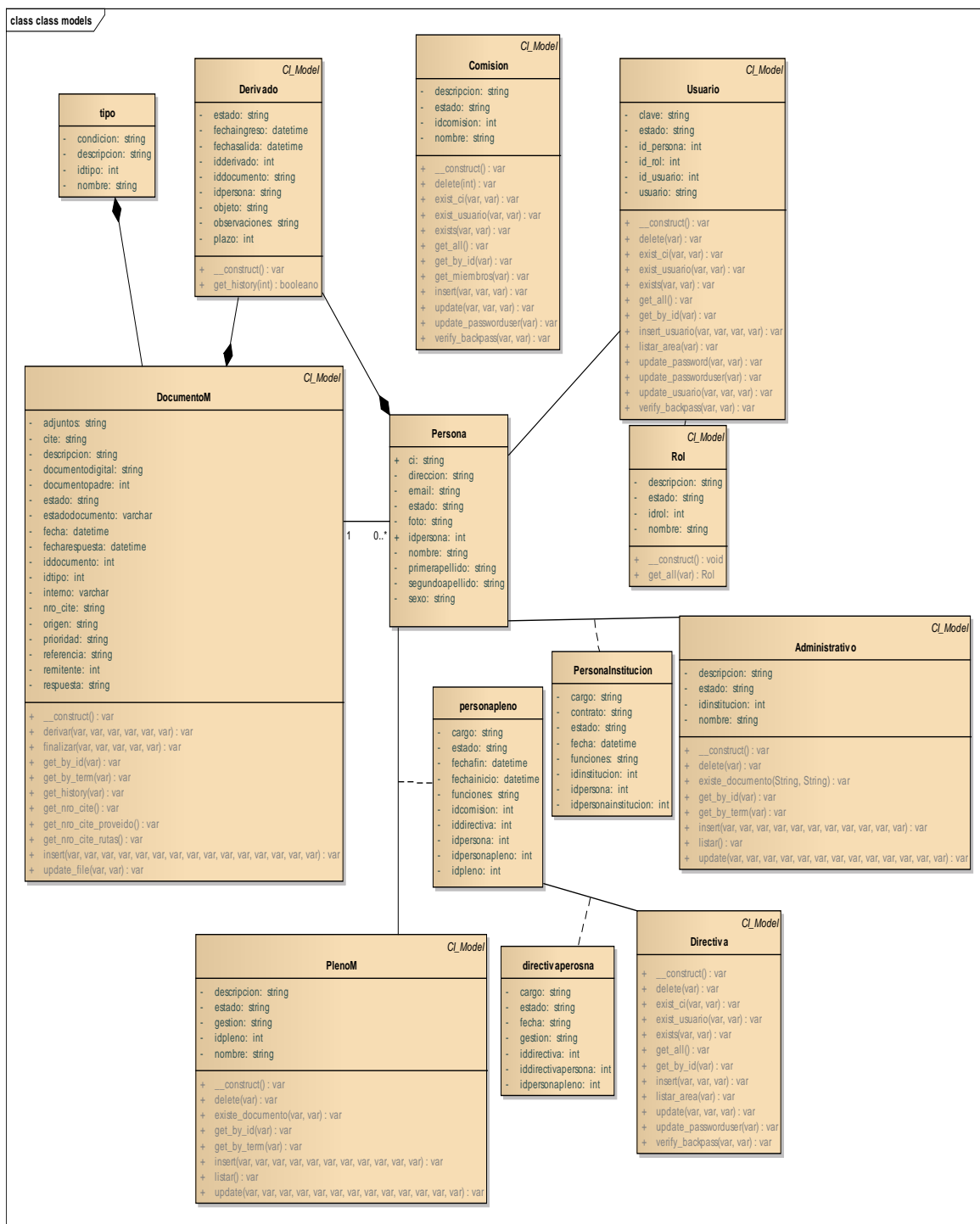


Figura 136 Diagrama de Clases

2.1.2.2.3.15.2 Modelo Entidad Relación

2.1.2.2.3.15.2.1 Introducción

Cuando se utiliza una base de datos para gestionar información, se está plasmando una parte del mundo real en una serie de tablas, registros y campos ubicados en un ordenador; creándose un modelo parcial de la realidad. Antes de crear físicamente estas tablas en el ordenador se debe realizar un modelo de datos.

Se suele cometer el error de ir creando nuevas tablas a medida que se van necesitando, haciendo así el modelo de datos y la construcción física de las tablas simultáneamente. El resultado de esto acaba siendo un sistema de información parcheado, con datos dispersos que terminan por no cumplir adecuadamente los requisitos necesarios.

2.1.2.2.3.15.2.2 Modelo Entidad / Relación

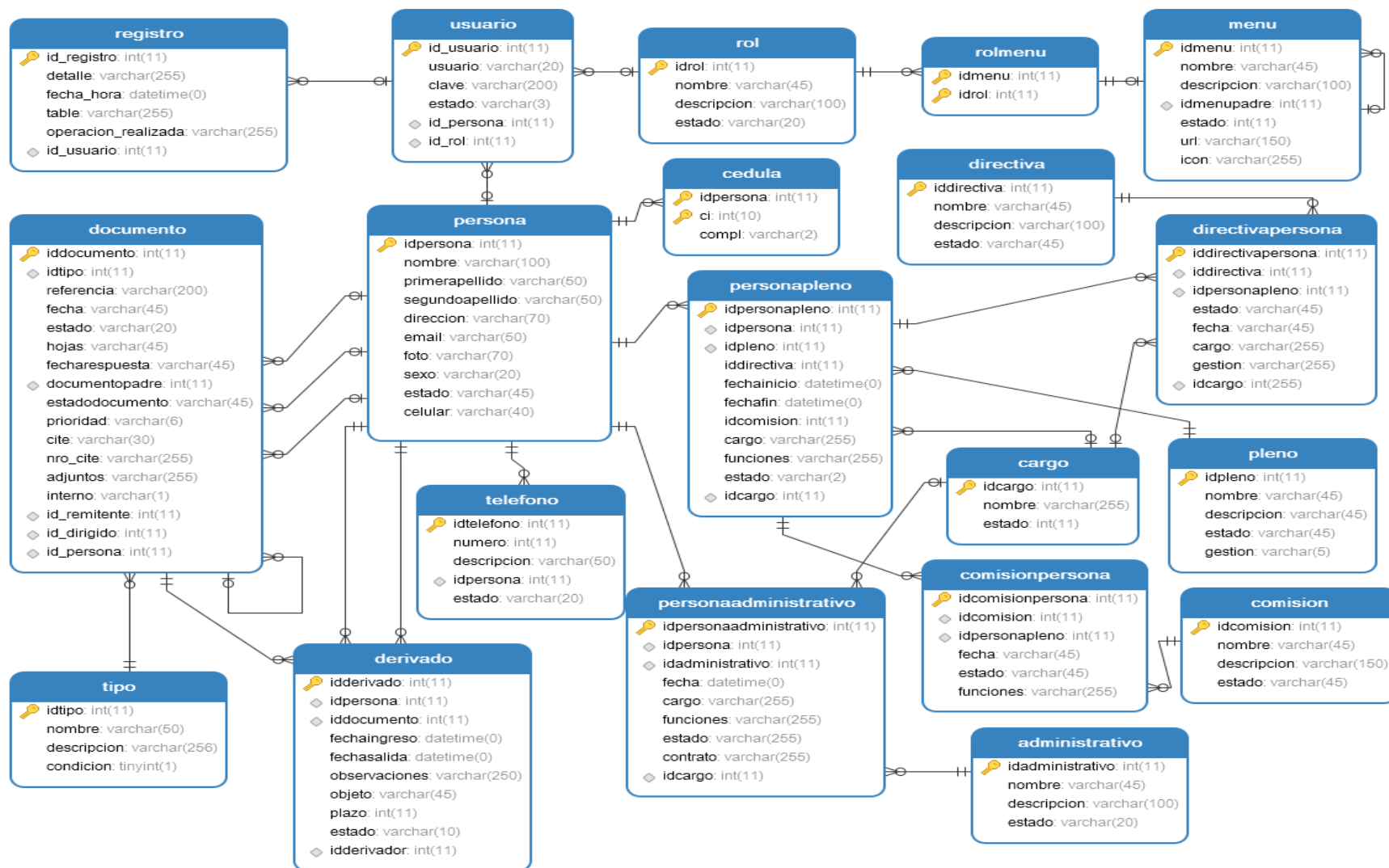


Figura 137 Diagrama Entidad Relación

2.1.2.2.3.15.2.3 Creación de la Base de Datos

```
DROP TABLE IF EXISTS `administrativo`;
CREATE TABLE `administrativo` (
  `idadministrativo` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `descripcion` varchar(100) CHARACTER SET utf8 COLLATE utf8_general_ci
NOT NULL,
  `estado` varchar(20) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  PRIMARY KEY (`idadministrativo`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 2 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;
```

```
DROP TABLE IF EXISTS `cargo`;
CREATE TABLE `cargo` (
  `idcargo` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(255) CHARACTER SET latin1 COLLATE latin1_swedish_ci
NULL DEFAULT NULL,
  `estado` int(11) NULL DEFAULT NULL,
  PRIMARY KEY (`idcargo`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 8 CHARACTER SET = latin1
COLLATE = latin1_swedish_ci ROW_FORMAT = Compact;
```

```
DROP TABLE IF EXISTS `cedula`;
CREATE TABLE `cedula` (
  `idpersona` int(11) NOT NULL,
  `ci` int(10) NOT NULL,
  `compl` varchar(2) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT NULL,
  PRIMARY KEY (`ci`, `idpersona`) USING BTREE,
  INDEX `fk_cedula_persona_idx`(`idpersona`) USING BTREE,
  CONSTRAINT `cedula_ibfk_1` FOREIGN KEY (`idpersona`) REFERENCES
`persona` (`idpersona`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8_general_ci
ROW_FORMAT = Compact;
```

```
DROP TABLE IF EXISTS `comision`;
CREATE TABLE `comision` (
  `idcomision` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
```

```

`descripcion` varchar(150) CHARACTER SET utf8 COLLATE utf8_general_ci
NULL DEFAULT NULL,
`estado` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL DEFAULT '1',
PRIMARY KEY (`idcomision`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;

```

```

DROP TABLE IF EXISTS `comisionpersona`;
CREATE TABLE `comisionpersona` (
  `idcomisionpersona` int(11) NOT NULL AUTO_INCREMENT,
  `idcomision` int(11) NOT NULL,
  `idpersonaplano` int(11) NOT NULL,
  `fecha` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `estado` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `funciones` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci
NULL DEFAULT NULL,
PRIMARY KEY (`idcomisionpersona`) USING BTREE,
INDEX `fk_comisionpersona_comision_idx` (`idcomision`) USING BTREE,
INDEX `fk_comisionpersona_personaplano` (`idpersonaplano`) USING BTREE,
CONSTRAINT `fk_comisionpersona_comision` FOREIGN KEY (`idcomision`)
REFERENCES `comision` (`idcomision`) ON DELETE NO ACTION ON UPDATE
NO ACTION,
CONSTRAINT `fk_comisionpersona_personaplano` FOREIGN KEY
(`idpersonaplano`) REFERENCES `personaplano` (`idpersonaplano`) ON DELETE
NO ACTION ON UPDATE NO ACTION
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;

```

```

DROP TABLE IF EXISTS `derivado`;
CREATE TABLE `derivado` (
  `idderivado` int(11) NOT NULL AUTO_INCREMENT,
  `idpersona` int(11) NOT NULL,
  `iddocumento` int(11) NOT NULL,
  `fechaingreso` datetime(0) NOT NULL,
  `fechasalida` datetime(0) NULL DEFAULT NULL,
  `observaciones` varchar(250) CHARACTER SET utf8 COLLATE utf8_general_ci
NULL DEFAULT NULL,
  `objeto` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `plazo` int(11) NULL DEFAULT NULL,
  `estado` varchar(10) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT 'ACTIVO',

```

```

`idderivador` int(11) NOT NULL,
PRIMARY KEY (`idderivado`) USING BTREE,
INDEX `fk_derivado_perosna_idx`(`idpersona`) USING BTREE,
INDEX `fk_derivado_documento_idx`(`iddocumento`) USING BTREE,
INDEX `idderivador`(`idderivador`) USING BTREE,
CONSTRAINT `derivado_ibfk_1` FOREIGN KEY (`idderivador`) REFERENCES
`persona` (`idpersona`) ON DELETE RESTRICT ON UPDATE RESTRICT,
CONSTRAINT `fk_derivado_documento` FOREIGN KEY (`iddocumento`)
REFERENCES `documento` (`iddocumento`) ON DELETE NO ACTION ON
UPDATE NO ACTION,
CONSTRAINT `fk_derivado_persona` FOREIGN KEY (`idpersona`)
REFERENCES `persona` (`idpersona`) ON DELETE NO ACTION ON UPDATE
NO ACTION
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;

```

```

DROP TABLE IF EXISTS `directiva`;
CREATE TABLE `directiva` (
  `iddirectiva` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `descripcion` varchar(100) CHARACTER SET utf8 COLLATE utf8_general_ci
NULL DEFAULT NULL,
  `estado` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT NULL,
  PRIMARY KEY (`iddirectiva`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 2 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;

```

```

DROP TABLE IF EXISTS `directivapersona`;
CREATE TABLE `directivapersona` (
  `iddirectivapersona` int(11) NOT NULL AUTO_INCREMENT,
  `iddirectiva` int(11) NOT NULL,
  `idpersonapleno` int(11) NOT NULL,
  `estado` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `fecha` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `cargo` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT NULL,
  `gestion` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL DEFAULT '2018',
  `idcargo` int(255) NULL DEFAULT NULL,
  PRIMARY KEY (`iddirectivapersona`) USING BTREE,
INDEX `fk_directivapersona_directiva_idx`(`iddirectiva`) USING BTREE,

```

```

INDEX `fk_directivapersona_personapleno_idx`(`idpersonapleno`) USING
BTREE,
INDEX `idcargo`(`idcargo`) USING BTREE,
INDEX `idcargo_2`(`idcargo`) USING BTREE,
CONSTRAINT `directivapersona_ibfk_1` FOREIGN KEY (`idcargo`)
REFERENCES `cargo` (`idcargo`) ON DELETE RESTRICT ON UPDATE
RESTRICT,
CONSTRAINT `fk_directivapersona_directiva` FOREIGN KEY (`iddirectiva`)
REFERENCES `directiva` (`iddirectiva`) ON DELETE NO ACTION ON UPDATE
NO ACTION,
CONSTRAINT `fk_directivapersona_personapleno` FOREIGN KEY
(`idpersonapleno`) REFERENCES `personapleno` (`idpersonapleno`) ON DELETE
NO ACTION ON UPDATE NO ACTION
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;

```

```

DROP TABLE IF EXISTS `documento`;
CREATE TABLE `documento` (
  `iddocumento` int(11) NOT NULL AUTO_INCREMENT,
  `idtipo` int(11) NOT NULL,
  `referencia` varchar(200) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `fecha` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `estado` varchar(20) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL DEFAULT 'I',
  `hojas` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `fecharespuesta` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci
NULL DEFAULT NULL,
  `documentopadre` int(11) NULL DEFAULT NULL,
  `estadodocumento` varchar(45) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL COMMENT 'poceso y finalizado\r\n',
  `prioridad` varchar(6) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT NULL,
  `cite` varchar(30) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT NULL,
  `nro_cite` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT NULL,
  `adjuntos` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT NULL,
  `interno` varchar(1) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `id_remitente` int(11) NULL DEFAULT NULL,
  `id_dirigido` int(11) NULL DEFAULT NULL,

```

```

`id_persona` int(11) NULL DEFAULT NULL,
PRIMARY KEY (`iddocumento`) USING BTREE,
INDEX `fk_articulo_categoria_idx`(`idtipo`) USING BTREE,
INDEX `fk_documento_documento_idx`(`documentopadre`) USING BTREE,
INDEX `id_remitente`(`id_remitente`) USING BTREE,
INDEX `documento_ibfk_3`(`id_persona`) USING BTREE,
INDEX `documento_ibfk_2`(`id_dirigido`) USING BTREE,
CONSTRAINT `documento_ibfk_1` FOREIGN KEY (`id_remitente`)
REFERENCES `persona` (`idpersona`) ON DELETE CASCADE ON UPDATE
RESTRICT,
CONSTRAINT `documento_ibfk_2` FOREIGN KEY (`id_dirigido`)
REFERENCES `persona` (`idpersona`) ON DELETE CASCADE ON UPDATE
RESTRICT,
CONSTRAINT `documento_ibfk_3` FOREIGN KEY (`id_persona`)
REFERENCES `persona` (`idpersona`) ON DELETE CASCADE ON UPDATE
RESTRICT,
CONSTRAINT `fk_documento_documento` FOREIGN KEY (`documentopadre`)
REFERENCES `documento` (`iddocumento`) ON DELETE CASCADE ON
UPDATE NO ACTION,
CONSTRAINT `fk_documento_tipo` FOREIGN KEY (`idtipo`) REFERENCES
`tipo` (`idtipo`) ON DELETE CASCADE ON UPDATE NO ACTION
) ENGINE = InnoDB AUTO_INCREMENT = 22 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;

```

```

DROP TABLE IF EXISTS `menu`;
CREATE TABLE `menu` (
  `idmenu` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
  NULL,
  `descripcion` varchar(100) CHARACTER SET utf8 COLLATE utf8_general_ci
  NULL DEFAULT NULL,
  `idmenupadre` int(11) NULL DEFAULT NULL,
  `estado` int(11) NOT NULL,
  `url` varchar(150) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
  NULL,
  `icon` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
  DEFAULT NULL,
  PRIMARY KEY (`idmenu`) USING BTREE,
  INDEX `fk_idmenupadre_idx`(`idmenupadre`) USING BTREE,
  CONSTRAINT `fk_idmenupadre` FOREIGN KEY (`idmenupadre`)
  REFERENCES `menu` (`idmenu`) ON DELETE NO ACTION ON UPDATE NO
  ACTION,
  CONSTRAINT `menu_ibfk_1` FOREIGN KEY (`idmenu`) REFERENCES
  `rolmenu` (`idmenu`) ON DELETE RESTRICT ON UPDATE RESTRICT

```

```
) ENGINE = InnoDB AUTO_INCREMENT = 16 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;
```

```
DROP TABLE IF EXISTS `persona`;
CREATE TABLE `persona` (
  `idpersona` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(100) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `primerapellido` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci
NULL DEFAULT NULL,
  `segundoapellido` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci
NULL DEFAULT NULL,
  `direccion` varchar(70) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT NULL,
  `email` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT NULL,
  `foto` varchar(70) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT NULL,
  `sexo` varchar(20) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT 'M',
  `estado` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL DEFAULT 'I',
  `celular` varchar(40) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT NULL,
  PRIMARY KEY (`idpersona`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 18 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;
```

```
DROP TABLE IF EXISTS `personaadministrativo`;
CREATE TABLE `personaadministrativo` (
  `idpersonaadministrativo` int(11) NOT NULL AUTO_INCREMENT,
  `idpersona` int(11) NOT NULL,
  `idadministrativo` int(11) NOT NULL,
  `fecha` datetime(0) NOT NULL,
  `cargo` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT NULL,
  `funciones` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci
NULL DEFAULT NULL,
  `estado` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT 'I',
  `contrato` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
DEFAULT 'EVENTUAL',
  `idcargo` int(11) NULL DEFAULT NULL,
  PRIMARY KEY (`idpersonaadministrativo`) USING BTREE,
```

```

INDEX `fk_personainstitucion_institucion_idx`(`idadministrativo`) USING
BTREE,
INDEX `fk_personainstitucion_persona_idx`(`idpersona`) USING BTREE,
INDEX `idcargo`(`idcargo`) USING BTREE,
CONSTRAINT `fk_personainstitucion_institucion` FOREIGN KEY
(`idadministrativo`) REFERENCES `administrativo`(`idadministrativo`) ON
DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `fk_personainstitucion_persona` FOREIGN KEY (`idpersona`)
REFERENCES `persona`(`idpersona`) ON DELETE NO ACTION ON UPDATE
NO ACTION,
CONSTRAINT `personaadministrativo_ibfk_1` FOREIGN KEY (`idcargo`)
REFERENCES `cargo`(`idcargo`) ON DELETE RESTRICT ON UPDATE
RESTRICT
) ENGINE = InnoDB AUTO_INCREMENT = 2 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;

```

```

DROP TABLE IF EXISTS `personaplano`;
CREATE TABLE `personaplano` (
  `idpersonaplano` int(11) NOT NULL AUTO_INCREMENT,
  `idpersona` int(11) NOT NULL,
  `idpleno` int(11) NOT NULL DEFAULT 1,
  `iddirectiva` int(11) NULL DEFAULT NULL,
  `fechainicio` datetime(0) NOT NULL,
  `fechafin` datetime(0) NULL DEFAULT NULL,
  `idcomision` int(11) NULL DEFAULT NULL,
  `cargo` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
  DEFAULT NULL,
  `funciones` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci
  NULL DEFAULT NULL,
  `estado` varchar(2) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
  DEFAULT '1',
  `idcargo` int(11) NULL DEFAULT NULL,
  PRIMARY KEY (`idpersonaplano`) USING BTREE,
  INDEX `fk_personaplano_pleno_idx`(`idpleno`) USING BTREE,
  INDEX `fk_personaplano_persona_idx`(`idpersona`) USING BTREE,
  INDEX `idcargo`(`idcargo`) USING BTREE,
  CONSTRAINT `fk_personaplano_persona` FOREIGN KEY (`idpersona`)
  REFERENCES `persona`(`idpersona`) ON DELETE NO ACTION ON UPDATE
  NO ACTION,
  CONSTRAINT `fk_personaplano_pleno` FOREIGN KEY (`idpleno`)
  REFERENCES `pleno`(`idpleno`) ON DELETE NO ACTION ON UPDATE NO
  ACTION,
  CONSTRAINT `personaplano_ibfk_1` FOREIGN KEY (`idcargo`) REFERENCES
  `cargo`(`idcargo`) ON DELETE RESTRICT ON UPDATE RESTRICT

```



```
) ENGINE = InnoDB AUTO_INCREMENT = 3 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;
```

```
DROP TABLE IF EXISTS `pleno`;
CREATE TABLE `pleno` (
  `idpleno` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
  NULL,
  `descripcion` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci
  NULL DEFAULT NULL,
  `estado` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
  NULL,
  `gestion` varchar(5) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
  NULL,
  PRIMARY KEY (`idpleno`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 2 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;
```

```
DROP TABLE IF EXISTS `registro`;
CREATE TABLE `registro` (
  `id_registro` int(11) NOT NULL AUTO_INCREMENT,
  `detalle` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
  DEFAULT NULL,
  `fecha_hora` datetime(0) NULL DEFAULT CURRENT_TIMESTAMP(0),
  `table` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
  DEFAULT NULL,
  `operacion_realizada` varchar(255) CHARACTER SET utf8 COLLATE
  utf8_general_ci NULL DEFAULT NULL,
  `id_usuario` int(11) NULL DEFAULT NULL,
  PRIMARY KEY (`id_registro`) USING BTREE,
  INDEX `id_usuario` (`id_usuario`) USING BTREE,
  CONSTRAINT `registro_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES
  `usuario` (`id_usuario`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE = InnoDB AUTO_INCREMENT = 694 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;
```

```
DROP TABLE IF EXISTS `rol`;
CREATE TABLE `rol` (
  `idrol` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(45) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
  NULL,
  `descripcion` varchar(100) CHARACTER SET utf8 COLLATE utf8_general_ci
  NULL DEFAULT NULL,
  `estado` varchar(20) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
  NULL,
```

```

PRIMARY KEY (`idrol`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 11 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;

```

```

DROP TABLE IF EXISTS `rolmenu`;
CREATE TABLE `rolmenu` (
  `idmenu` int(11) NOT NULL,
  `idrol` int(11) NOT NULL,
  PRIMARY KEY (`idmenu`, `idrol`) USING BTREE,
  INDEX `fk_rolmenu_menu_idx` (`idmenu`) USING BTREE,
  INDEX `fk_rolmenu_rol_idx` (`idrol`) USING BTREE,
  CONSTRAINT `rolmenu_ibfk_1` FOREIGN KEY (`idrol`) REFERENCES `rol`
(`idrol`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8_general_ci
ROW_FORMAT = Compact;

```

```

DROP TABLE IF EXISTS `telefono`;
CREATE TABLE `telefono` (
  `idtelefono` int(11) NOT NULL AUTO_INCREMENT,
  `numero` int(11) NOT NULL,
  `descripcion` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci
NULL DEFAULT NULL,
  `idpersona` int(11) NOT NULL,
  `estado` varchar(20) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  PRIMARY KEY (`idtelefono`) USING BTREE,
  INDEX `fk_telefono_persona_idx` (`idpersona`) USING BTREE,
  CONSTRAINT `fk_telefono_persona` FOREIGN KEY (`idpersona`)
REFERENCES `persona` (`idpersona`) ON DELETE NO ACTION ON UPDATE
NO ACTION
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;

```

```

DROP TABLE IF EXISTS `tipo`;
CREATE TABLE `tipo` (
  `idtipo` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL,
  `descripcion` varchar(256) CHARACTER SET utf8 COLLATE utf8_general_ci
NULL DEFAULT NULL,
  `condicion` tinyint(1) NOT NULL,
  PRIMARY KEY (`idtipo`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 9 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Compact;

```

```

DROP TABLE IF EXISTS `usuario`;
CREATE TABLE `usuario` (
  `id_usuario` int(11) NOT NULL AUTO_INCREMENT,
  `usuario` varchar(20) CHARACTER SET latin1 COLLATE latin1_swedish_ci
  NULL DEFAULT NULL,
  `clave` varchar(200) CHARACTER SET latin1 COLLATE latin1_swedish_ci
  NULL DEFAULT NULL,
  `estado` varchar(3) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL
  DEFAULT '1',
  `id_persona` int(11) NULL DEFAULT NULL,
  `id_rol` int(11) NULL DEFAULT NULL,
  PRIMARY KEY (`id_usuario`) USING BTREE,
  INDEX `id_persona`(`id_persona`) USING BTREE,
  INDEX `id_rol`(`id_rol`) USING BTREE,
  CONSTRAINT `usuario_ibfk_1` FOREIGN KEY (`id_persona`) REFERENCES
  `persona` (`idpersona`) ON DELETE RESTRICT ON UPDATE RESTRICT,
  CONSTRAINT `usuario_ibfk_2` FOREIGN KEY (`id_rol`) REFERENCES `rol`
  (`idrol`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE = InnoDB AUTO_INCREMENT = 8 CHARACTER SET = latin1
COLLATE = latin1_swedish_ci ROW_FORMAT = Compact;

```

2.1.2.2.3.15.2.4 Diccionario de Datos

2.1.2.2.3.15.2.4.1 Comisión

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idcomision (<i>Primaria</i>)	int(11)	No			
nombre	varchar(45)	No			Nombre de la Comisión
descripcion	varchar(150)	Sí	<i>NULL</i>		
estado	varchar(45)	No	1		

2.1.2.2.3.15.2.4.2 Comisionpersona

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idcomisionpersona (<i>Primaria</i>)	int(11)	No			

idcomision	int(11)	No		comision -> idcomision	
idpersonaplano	int(11)	No		personaplano -> idpersonaplano	
fecha	varchar(45)	No			Fecha de inicio de la comisión
estado	varchar(45)	No			
funciones	varchar(255)	Sí	NULL		Trabajo encomendado

2.1.2.2.3.15.2.4.3 Derivado

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idderivado (<i>Primaria</i>)	int(11)	No			
idpersona	int(11)	No		persona -> idpersona	
iddocumento	int(11)	No		documento -> iddocumento	
fechaingreso	datetime	No			
fechasalida	datetime	Sí	NULL		
observaciones	varchar(250)	Sí	NULL		
objeto	varchar(45)	No			Para que se esta derivando
plazo	int(11)	Sí	NULL		Tiempo limite de respuesta
estado	varchar(10)	Sí	ACTIVO		
idderivador	int(11)	No		persona -> idpersona	

2.1.2.2.3.15.2.4.4 Directiva

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
iddirectiva (<i>Primaria</i>)	int(11)	No			
nombre	varchar(45)	No			
descripcion	varchar(100)	Sí	NULL		
estado	varchar(45)	Sí	NULL		

2.1.2.2.3.15.2.4.5 Directivapersona

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
iddirectivapersona (<i>Primaria</i>)	int(11)	No			
iddirectiva	int(11)	No		directiva -> iddirectiva	
idpersonaplano	int(11)	No		personaplano -> idpersonaplano	
estado	varchar(45)	No			
fecha	varchar(45)	No			
cargo	varchar(255)	Sí	NULL		Cargo que ocupa e la directiva
gestion	varchar(255)	No	2019		
idcargo	int(255)	Sí	NULL	cargo -> idcargo	

2.1.2.2.3.15.2.4.6 Documento

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
iddocumento (<i>Primaria</i>)	int(11)	No			

idtipo	int(11)	No		tipo -> idtipo	
referencia	varchar(200)	No			Título del documento
fecha	varchar(45)	No			
estado	varchar(20)	No	1		
hojas	varchar(45)	No			Cantidad de hojas que tiene el documento
fecharespuesta	varchar(45)	Sí	NULL		
documentopadre	int(11)	Sí	NULL	documento -> iddocumento	Referencia a otro documento previamente registrado
estadodocumento	varchar(45)	No			proceso y finalizado
prioridad	varchar(6)	Sí	NULL		Tipo de prioridad alta media baja
cite	varchar(30)	Sí	NULL		Sigla única que identifica al documento
nro_cite	varchar(255)	Sí	NULL		Numero del cite correlativo dependiendo de si es un HP o HR
adjuntos	varchar(255)	Sí	NULL		Documento referencial en digital
interno	varchar(1)	No			1 interno y 2 externo
id_remitente	int(11)	Sí	NULL	persona -> idpersona	
id_dirigido	int(11)	Sí	NULL	persona -> idpersona	

id_persona	int(11)	Sí	NULL	persona idpersona	->
------------	---------	----	------	----------------------	----

2.1.2.2.3.15.2.4.7 Administrativo

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idadministrativo (<i>Primaria</i>)	int(11)	No			
nombre	varchar(45)	No			Nombre del área referencial
descripcion	varchar(100)	No			
estado	varchar(20)	No			

2.1.2.2.3.15.2.4.8 Menu

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idmenu (<i>Primaria</i>)	int(11)	No		rolmenu idmenu	->
nombre	varchar(45)	No			
descripcion	varchar(100)	Sí	NULL		
idmenupadre	int(11)	Sí	NULL	menu idmenu	->
estado	int(11)	No			
url	varchar(150)	No			
icon	varchar(255)	Sí	NULL		

2.1.2.2.3.15.2.4.9 Persona

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idpersona (<i>Primaria</i>)	int(11)	No			
nombre	varchar(100)	No			
primerapellido	varchar(50)	Sí	<i>NULL</i>		
segundoapellido	varchar(50)	Sí	<i>NULL</i>		
direccion	varchar(70)	Sí	<i>NULL</i>		
email	varchar(50)	Sí	<i>NULL</i>		Correo electronico
foto	varchar(70)	Sí	<i>NULL</i>		Foto digital
sexo	varchar(20)	Sí	M		
estado	varchar(45)	No	1		
celular	varchar(40)	Sí	<i>NULL</i>		

2.1.2.2.3.15.2.4.10 Personapleno

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idpersonaplenu (<i>Primaria</i>)	int(11)	No			
idpersona	int(11)	No		persona idpersona ->	
idpleno	int(11)	No	1	pleno -> idpleno	
iddirectiva	int(11)	Sí	<i>NULL</i>		
fechainicio	datetime	No			Fecha de inicio del cargo
fechafin	datetime	Sí	<i>NULL</i>		Fecha fin del cargo
cargo	varchar(255)	Sí	<i>NULL</i>		

funciones	varchar(255)	Sí	NULL		
estado	varchar(2)	Sí	1		
idcargo	int(11)	Sí	NULL	cargo -> idcargo	

2.1.2.2.3.15.2.4.11 Pleno

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idpleno (<i>Primaria</i>)	int(11)	No			
nombre	varchar(45)	No			
descripcion	varchar(45)	Sí	NULL		
estado	varchar(45)	No			
gestion	varchar(5)	No			

2.1.2.2.3.15.2.4.12 Rol

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idrol (<i>Primaria</i>)	int(11)	No			
nombre	varchar(45)	No			
descripcion	varchar(100)	Sí	NULL		
estado	varchar(20)	No			

2.1.2.2.3.15.2.4.13 Rolmenu

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idmenu (<i>Primaria</i>)	int(11)	No			
idrol (<i>Primaria</i>)	int(11)	No		rol -> idrol	

2.1.2.2.3.15.2.4.14 Tipo

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idtipo (<i>Primaria</i>)	int(11)	No			
nombre	varchar(50)	No			
descripcion	varchar(256)	Sí	<i>NULL</i>		
condicion	tinyint(1)	No			

2.1.2.2.3.15.2.4.15 Usuario

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
id_usuario (<i>Primaria</i>)	int(11)	No			
usuario	varchar(20)	Sí	<i>NULL</i>		
clave	varchar(200)	Sí	<i>NULL</i>		
estado	varchar(3)	Sí	1		
id_persona	int(11)	Sí	<i>NULL</i>	persona -> idpersona	
id_rol	int(11)	Sí	<i>NULL</i>	rol -> idrol	

2.1.2.2.3.15.2.4.16 Registro

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
id_registro (<i>Primaria</i>)	int(11)	No			
detalle	varchar(255)	Sí	<i>NULL</i>		
fecha_hora	datetime	Sí	CURRENT_TIMESTAMP		
table	varchar(255)	Sí	<i>NULL</i>		
operacion_realizada	varchar(255)	Sí	<i>NULL</i>		
id_usuario	int(11)	Sí	<i>NULL</i>	usuario ->	

				id_usuario	
--	--	--	--	------------	--

2.1.2.2.3.15.2.4.17 Personaadministrativo

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idpersonaadministrativo (<i>Primaria</i>)	int(11)	No			
idpersona	int(11)	No		persona idpersona ->	
idadministrativo	int(11)	No		administrativo -> idadministrati vo	
fecha	datetime	No			
cargo	varchar(255)	Sí	NULL		
funciones	varchar(255)	Sí	NULL		
estado	varchar(255)	Sí	1		
contrato	varchar(255)	Sí	EVENTUAL		
idcargo	int(11)	Sí	NULL	cargo idcargo ->	

2.1.2.2.3.15.2.4.18 Cargo

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idcargo (<i>Primaria</i>)	int(11)	No			
nombre	varchar(255)	Sí	NULL		
estado	int(11)	Sí	NULL		

2.1.2.2.3.15.2.4.19 Cedula

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
idpersona (<i>Primaria</i>)	int(11)	No		persona idpersona ->	
ci (<i>Primaria</i>)	int(10)	No			
compl	varchar(2)	Sí	<i>NULL</i>		Complemento de la cedula de identidad

2.1.2.2.3.16 Modelo de Implementación

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo es sólo una versión preliminar al final de la fase de Elaboración, posteriormente tiene bastante refinamiento).

2.1.2.2.3.16.1 Modelado de Diagrama de Componentes

2.1.2.2.3.16.1.1 Diagrama de Componente

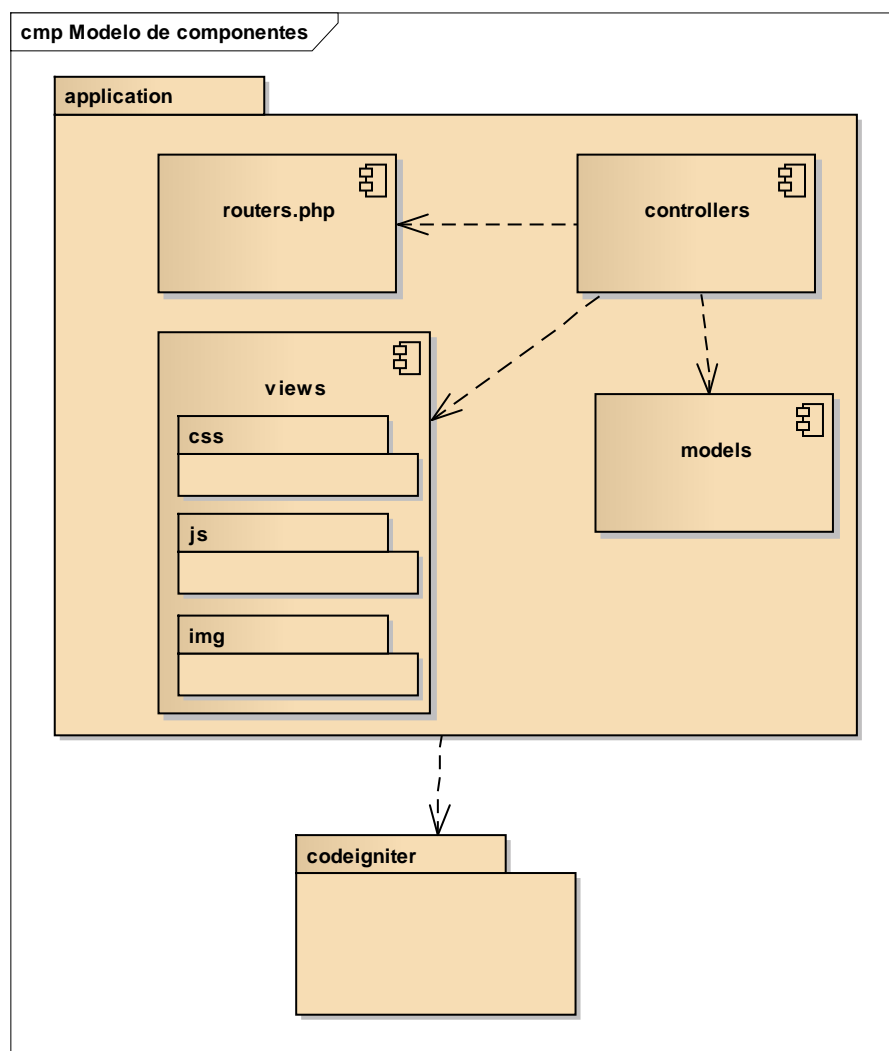


Figura 138 Diagrama de Componentes

2.1.2.2.3.17 Modelo de Despliegue

Este modelo muestra el despliegue la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.

2.1.2.2.3.17.1 Modelado de Diagrama de Despliegue

2.1.2.2.3.17.1.1 Diagrama de despliegue

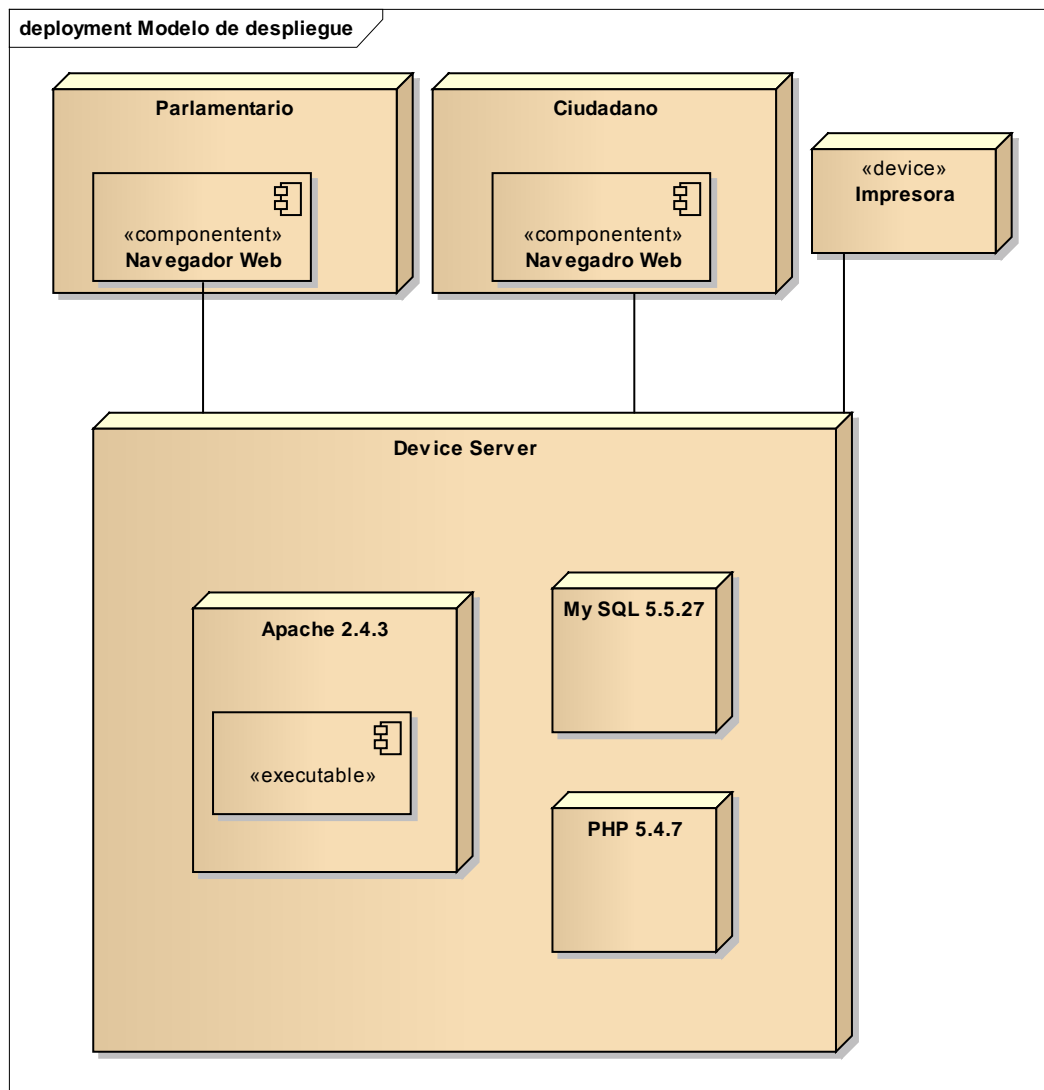


Figura 139 Diagrama de despliegue

2.1.2.2.3.18 Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.

2.1.2.2.3.18.1 Plan de Pruebas

2.1.2.2.3.18.1.1 Descripción de Aspectos Generales

Esta sección establece el alcance y el objetivo del Plan de Pruebas. Es aquí donde se describen los aspectos fundamentales del esfuerzo que se hará para probar cada uno de los módulos que conforman el sistema descrito en este Plan de Desarrollo de Software, independiente las características y tamaño que ésta pueda tener.

2.1.2.2.3.18.1.1.1 Objetivo

Este Plan de Pruebas fue desarrollado con el fin de guiar el proceso de pruebas del proyecto Gestión documental en la Brigada Parlamentaria de Tarija Mejorada, desarrollado por nuestro grupo de trabajo, esto con el fin de asegurar una excelente calidad del software desarrollado, encontrando para ello errores que puedan perjudicar en el funcionamiento de este, los cuales una vez corregidos, se podrá elaborar un documento que presente las evaluaciones correspondientes en el plazo más corto posible.

2.1.2.2.3.18.1.1.2 Entorno o Marco

La necesidad de aplicar este plan de pruebas radica en la necesidad de obtener un software de calidad, que cumpla con los requerimientos de la Brigada Parlamentaria de Tarija, además que sea aprobado por la comunidad de usuarios finales del sistema.

2.1.2.2.3.18.1.1.3 Arquitectura Técnica

Para la realización de las pruebas se tomará en cuenta los módulos desarrollados en la fase de construcción y los diagramas realizados en la fase de elaboración.

2.1.2.2.3.18.1.1.4 Especificaciones del Software y Hardware

Corresponde a una lista individualizada de todo el hardware y el software que utiliza la aplicación, incluyendo proveedores y versiones.

HARDWARE
Computadora con la suficiente capacidad de soportar el software

Tabla 59 Hardware

SOFTWARE		
Aplicación	Versión	Descripción
Sistema de control de documentación de la Brigada Parlamentaria de Tarija	0.1	Sistema desarrollado por el grupo de trabajo.
Enterprise Architect	8.0.863	Herramienta de Diseño y Desarrollo de Diagramas UML para RUP
MySql	5.1.53	Motor de Base de Datos Relacional.

Tabla 60 Software

OTROS

Descripción	Versión	Descripción
Documentación	0.1	Documentación de Desarrollo de Software

Tabla 61 Otros

2.1.2.2.3.18.1.1.5 Alcance

Este plan describe las pruebas del sistema, que serán aplicadas a los componentes del Sistema de control de documentación de la Brigada Parlamentaria de Tarija.

Se asume que antes de probar cada módulo habrá una revisión informal y solo el código que ha sido revisado como exitoso será probado.

Las unidades de prueba serán realizadas a través del programa manejador de pruebas que ejecuta el chequeo de los límites y las pruebas básicas de caja negra.

Las siguientes medidas de desempeño serán probadas:

Validación correcta de las Entradas de Datos.

Tiempo de respuesta de carga del Sitio Web.

Tiempo de respuesta y Pool de conexiones a la Base de Datos.

2.1.2.2.3.18.1.1.6 Referencias

Glosario del Proyecto.

Plan de Desarrollo de Software.

2.1.2.2.3.18.2 Descripción de las Pruebas Planeadas

2.1.2.2.3.18.2.1 Descripción de las pruebas incluidas

Las siguientes pruebas serán realizadas a todo el Sistema de Informático:

- Pruebas de Pool de Conexiones.

- Pruebas funcionales.
- Pruebas de Validación de Datos.
- Pruebas de Perfiles de desempeño.

2.1.2.2.3.18.3 Planeación de la Ejecución de las Pruebas

La siguiente es una descripción de los tipos y técnicas de pruebas que se harán sobre el Sistema Informático, junto con la manera en la que éstas se ejecutarán.

2.1.2.2.3.18.4 Tipos y Técnicas de pruebas

2.1.2.2.3.18.4.1 Pruebas de Pool de Conexiones

Las pruebas de pooling de conexiones serán realizadas para asegurar que los datos no son corruptos, dentro de las estructuras de datos internas. Estas pruebas son realizadas de forma independiente de la Interfaz de Usuario, en forma de pruebas de caja negra.

Objetivo de la Técnica:	Verificar la las conexiones realizadas a la base de datos y determinar el rendimiento del Sistema y del Motor de Base de Datos (MySQL).
Técnica:	Analizar la carga a la conexión mediante la capa abstracta PDO, simulando usuarios conectados.
Hitos:	Rastreo de cuello de botella en la conexión a la base de datos, realizada por el sistema.
Criterios de Éxito:	Las conexiones se realizan con fluidez y una por usuario.

Tabla 62 Pruebas de Pool de Conexiones

2.1.2.2.3.18.4.2 Pruebas Funcionales

Las pruebas funcionales serán realizadas para verificar que todos los requerimientos funcionales se cumplen satisfactoriamente. Estas serán cumplidas a través de pruebas de caja negra.

Objetivo de la Técnica:	Verificar requerimientos funcionales del sistema.
Técnicas:	Verificar que los requerimientos funcionales expuestos en los casos de uso se cumplen.
Hitos:	Casos de Uso del Sistema Informático.
Herramientas Requeridas:	Herramientas de backup y recuperación. Herramientas de Instalación y Monitoreo (registro, disco duro, CPU, memoria, etc.) Herramientas de Generación de Datos
Criterio de éxito:	Los siguientes elementos son probados exitosamente: <ul style="list-style-type: none"> - Todos los casos de uso del Sistema Informático. - La prueba funcional fue realizada juntamente con la persona encargada de su Administración.

Tabla 63 Pruebas Funcionales

2.1.2.2.3.18.4.3 Pruebas de Validación de Datos

Las pruebas de validación de datos verifican la interacción del usuario con el software y la información que introduce. El objetivo de las pruebas de la interfaz de usuario es asegurar que dicha interfaz proporciona al usuario el acceso y validación a todos los datos introducidos por el Usuario con motivo de dar Integridad a la información.

Objetivo de la Técnica:	Verificar y probar la Interfaz de entrada Usuario - Sistema Verificar todas las posibles combinaciones de caracteres para entradas del sistema.
Técnica:	Crear o modificar pruebas para cada ventana para verificar todas las entradas de datos.
Hitos:	El testear verificara todas las entradas del Sistema.
Herramientas Requeridas:	-
Criterio de Éxito:	Todas las entradas han sido verificadas mediante combinaciones de caracteres para dar con los posibles errores del Sistema.

Tabla 64 Pruebas de Validación de Datos

2.1.2.2.3.18.4.4 Pruebas de Perfiles de Desempeño

Los perfiles de desempeño prueban tiempos de respuesta, tasas de transacción y otros requerimientos sensibles al tiempo son medidos y evaluados. El objetivo de los perfiles de desempeño es verificar el desempeño con el que los requerimientos se han cumplido. Un perfil de desempeño es implementado y ejecutado para refinar el desempeño y comportamiento de los elementos objetivo de las pruebas.

Objetivo de la Técnica:	Determinar el comportamiento del objeto de la prueba bajo: <ul style="list-style-type: none"> - Carga de trabajo anticipada. - Peor caso de la carga de trabajo.
Técnica:	Concurrencia de visita al Sistema Informático. Prueba de Stress al servidor Web.

Hitos:	Cálculo de tiempos de software.
Herramientas requeridas:	
Criterio de éxito:	Los tiempos de respuesta son los esperados para el número de usuarios que acceden al Sistema Informático.

Tabla 65 Pruebas de Perfiles de Desempeño

2.1.2.2.3.18.4.5 Ejecución de las Pruebas

2.1.2.2.3.18.4.5.1 Demostración y Resultados de la prueba de Validación de Datos

2.1.2.2.3.18.4.5.2 Partición Equivalente : Casos de Prueba Caja Negra

Interfaz: Iniciar Sesión

Username: Alfanumérico 20 caracteres

Password: Alfanumérico 20 caracteres

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Username Tamaño: Username	1. Alfanumérico 2. $5 \leq \text{Caracteres} \leq 20$	3. Vacío 4. Caracteres > 50 y Caracteres < 5
Tipo: Password Tamaño: Password	5. Alfanumérico 6. $5 \leq \text{Caracteres} \leq 20$	7. Vacío 8. Caracteres > 50 y Caracteres < 5

Tabla 66. Caso de Prueba Caja Negra: Iniciar Sesión

Casos de Prueba válidas

CP1	
Username	Password
Ae7143945	administrador

Cubre las clases de equivalencias válidas: 1-2-5-6

Casos de Pruebas Inválidas

CP2	
Username	Password
adm	

Cubre las clases de equivalencias inválidas: 7

CP3	
Username	Password
	123

Cubre las clases de equivalencias inválidas: 3

2.1.2.2.3.18.4.5.3 Prueba de Perfil de Desempeño

El componente principal de Unit Testing es denominado *Plan de Prueba* o *Test Plan*, en él se definen todos los aspectos relacionados con una prueba de carga, como : parámetros empleados por requisición, tipo de reportes a generarse con los resultados obtenidos, la posible reutilización de requisiciones compuestas por usuarios, entre otros aspectos.

2.1.2.2.3.19 Solicitud de Cambio

2.1.2.2.3.19.1 Introducción

Cuando construimos software de computadora surgen cambios. Debido a esto, necesitamos controlarlos eficazmente, en la cual es un conjunto de actividades

diseñadas para controlar el cambio identificando los productos del trabajo que probablemente cambien estableciendo relaciones entre ellos definiendo mecanismos para gestionar distintas versiones de estos productos controlando los cambios realizados y auditando e informando de los cambios realizados.

Los cambios propuestos para los artefactos se formalizan mediante este documento.

Mediante este documento se hace un seguimiento de los defectos detectados, solicitud de mejoras o cambios en los requisitos del producto. Así se provee un registro de decisiones de cambios, de su evaluación de impacto, y se asegura que éstos sean conocidos por el director del proyecto. Los cambios se establecen respecto de la última baseline (el estado del conjunto de los artefactos en un momento determinado del proyecto) establecida. En nuestro caso al final de cada iteración se establecerá una baseline o Línea Base.

2.1.2.2.3.19.2 Descripción

Las solicitudes de cambio se presentan de acuerdo a las presentaciones parciales de proyecto. Cada línea base presenta los siguientes campos:

Nombre: En este campo se anota el nombre de aquel elemento del cual se desea la modificación, actualización o mejoramiento.

Tipo: Se indica el tipo de documento del que se desea el cambio.

Fecha: Se indica la fecha de solicitud del cambio.

Fase / Iteración: Se indica la fase / iteración en la que se realizó el cambio.

Fecha de Actualización: Se indica la fecha que se realizó el cambio.

Encargado: Se indica el nombre del rol responsable de autorizar, realizar y verificar el cambio una vez finalizado este.

Informe del Cambio: Se describe los resultados obtenidos provenientes de la modificación del artefacto.

2.1.2.2.3.19.3 Control de Solicitudes de Cambio

2.1.2.2.3.20 Lista de Riesgos

2.1.2.2.3.20.1 Introducción

2.1.2.2.3.20.1.1 Alcance

Este plan es emprendido ante la fase de elaboración para asegurar que ninguno de los riesgos identificados sea direccionado durante la misma fase de elaboración. Apenas el plan de gestión de riesgos haya sido documentado, el proceso de prevención de riesgos estará ocupado para monitorear y controlar la probabilidad y el impacto de los riesgos sobre el proyecto.

2.1.2.2.3.20.2 Proceso de Manipulación de Riesgos

En esta sección se establecen todos los aspectos que son necesarios para que cada riesgo que se pueda identificar dentro del proyecto en desarrollo sea debidamente documentado, evaluado y mitigado tanto como sea necesario. Para ello se establecen los procedimientos que en este proyecto deben ser ejecutados al momento de identificar un determinado riesgo, el mismo está acorde a las políticas de la organización.

2.1.2.2.3.20.3 Roles Encargados de Gestionar los Riesgos

Se establecen los roles participantes para el manejo de riesgos en su proyecto, los cuales van a ser los encargados de gestionar los riesgos y conforme a cada uno se señalan cuáles son sus funciones específicas.

Roles para el Manejo de Riesgos	Funciones
Jefe del Proyecto	Debe establecer un plan donde se debe contemplar los riesgos que sean identificados para el proyecto, adicionalmente dicho plan puede contener las descripciones, análisis,

	<p>prioridades y estrategias que sirvan para minimizar el impacto que los riesgos puedan llegar a tener.</p> <p>Asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios.</p> <p>Mantiene al equipo del proyecto enfocado en los objetivos.</p> <p>Gestión de riesgos. Planificación y control del proyecto.</p> <p>Identificar y valorar preliminarmente los riesgos encontrados.</p>
--	---

Tabla 67 Roles Encargados de Gestionar los Riesgos

2.1.2.2.3.20.4 Escala de Ponderación de los Riesgos

Se establecen un sistema de puntuación que sirva para establecer la calificación a los diversos riesgos, como propuesta tenemos:

Puntaje	Calificación	Descripción General
0% - 20%	Muy Bajo	Riesgo muy improbable, la probabilidad de ocurrencia es bajísima y es muy posible que no ocurra.
20% - 40%	Bajo	Riesgo de baja probabilidad de ocurrencia, es posible que la no ocurrencia de este.

40% - 60%	Moderado	Riesgo medianamente probable, puede ocurrir como puede no ocurrir.
60% - 80%	Alto	Riesgo muy probable, podría ocurrir, es muy probable la ocurrencia en el proceso de desarrollo.
80% - 100%	Muy alto	Riesgo altamente probable para ocurrir dado que las circunstancias que generarían este evento son muy probables.

Tabla 68 Escala de Ponderación de los Riesgo

2.1.2.2.3.20.5 Lista de Riesgos

Categoría de Riesgo			Identificación de Riesgos		Análisis de Riesgo		Planificación de Riesgos			Supervisión de Riesgos
Proyecto	Producto	Negocio	Nº	Riesgo	Ponderación	Efecto	Estrategias		Plan de contingencia	Políticas de Supervisión
							Anulación	Disminución o Preventiva		
*	*	*	1	La ausencia del Director del proyecto o que se desvíe del objetivo de terminar el proyecto.	20%	No poder llevar a cabo el proyecto y reprobar la materia .	-	*Ser consciente por sí mismo y trazarse el objetivo para poder llevar a su término el proyecto.	-	Realizar un control y seguimiento a los objetivos del proyecto.

	*		2	Que el personal cambie en la institución.	20%	Que no se brinde la misma ayuda al equipo de desarrollo.	-	*Conversaciones con el nuevo personal.	-	Realizar un control y seguimiento a posibles cambios del personal de la institución.
*	*		3	Los usuarios rehúsan utilizar el sistema.	20%	Los usuarios no quieren que su sistema sea	-	*En conversación con los usuarios finales, indicar las ventajas que	-	Realizar un control y seguimiento a los usuarios.

						automa tizado.		presentara el software que se desarrollara. *Preparar un manual de usuario comprensible que pueda estimular a los usuarios a usar el sistema a desarrollar.		
	*		4	Los usuarios no tengan	50%	El sistema sea mal	-	*Aconseja al usuario interesado a	-	Realizar un control y seguimiento a los

				conocimient o en el manejo del sistema.		utilizad o o solo utilizad o en parte.		cerca de capacitación para realizarse en el presente o futuro. <input type="checkbox"/> *Prever para que el usuario final tenga el conocimiento suficiente para poder manejar el sistema		usuarios en el manejo del sistema.
*	*	*	5	Realizar una mala	50%	El product	-	*Compara con otros	-	Realizar un control y

				estimación del tamaño del software.		o no será entregado en la fecha acordada.		proyectos al nuestro para tener una vista general de una estimación del tamaño del proyecto. <input type="checkbox"/> *Realizar lectura de recursos bibliográficos disponibles.		seguimiento a las estimaciones del proyecto.
*	*	*	6	Realizar una mala estimación	25%	El software no	-	*Obtener un mayor conocimiento	-	Realizar un control y seguimiento a los

				de los riesgos.		llegue a satisfacer las necesidades del usuario .		de las acciones se realizan en la empresa cliente, las acciones que realizara nuestro equipo de desarrollo y tener en claro el objetivo y el alcance de nuestro proyecto. <input type="checkbox"/> Obtener ayuda de expertos,		a los posibles riesgos.
--	--	--	--	-----------------	--	---	--	--	--	-------------------------

								libros y otros recursos que puedan ayudarnos en el desarrollo del plan de gestión de riesgos.		
*	*		7	No tener buen conocimiento en el manejo de las herramientas de trabajo.	30%	El software pueda tener fallas y no pueda ser entregado a	-	*Utilizar herramientas conocidas. <input type="checkbox"/> *Asegurarse que las personas miembros del equipo de desarrollo	-	Realizar un control y seguimiento a las herramientas con las que se va a trabajar.

						tiempo.		cuentan con el conocimiento necesario y suficiente acerca del uso de las herramientas a utilizar en el proyecto.		
	*		8	Las Interfaces y el diseño de las pantallas no sean correctos.	30%	Las interfaces para el usuario lleguen a ser desagradables.	-	*Establecer con anticipación las interfaces y el entorno gráfico a utilizar.	-	Realizar un control y seguimiento a las interfaces o pantallas que vayan de acuerdo a la facilidad del usuario.

						dables.		<input type="checkbox"/> *Hacer uso de herramientas que puedan otorgar una mejor apariencia a las interfaces. <input type="checkbox"/> *Obtener conocimiento de buenas normas para el diseño de interfaces.		
*	*	*		No llegar a la culminación	40%	La empresa quede	-	*Estimar la calendarización del	-	Realizar un control y seguimiento al

				del sistema en la fecha indicada.		perjudi cada.		proyecto haciendo uso de técnicas, métodos y herramientas adecuadas que nos faciliten una correcta estimación.		calendario del proyecto.
*		*	9	El coste del proyecto llegue a aumentarse durante el desarrollo.	25%	El proyecto puede cancelarse temporal o definitivamente	-	*Hacer uso de técnicas de Ingeniería de Software para realizar una buena estimación del coste.	-	Realizar un control y seguimiento a los costes del proyecto.

						e, por el incremento del precio para el usuario .		<input type="checkbox"/> *Obtener información actualizada del coste del desarrollo proyectos similares: personal, recursos de hardware y otros.		
	*		10	La institución no cuenta con el Hardware y Software	20%	El sistema no pueda ser implem	-	*Aconsejar y plantear equipo de hardware y recursos de software	-	Realizar un control y seguimiento al software y hardware de la institución.

				necesarios.		entado.		necesario para el funcionamien to del sistema con anticipación.		
*	*		11	El personal está enfermo y no disponible en momentos críticos en el desarrollo del proyecto.	40%	La empres a quede perjudi cada y no se present e el proyect o en la fecha acordad a	-	-	Buscar temporalme nte un cajero	-

						(retraso).				
*	*		12	Posible Falla en algún dispositivo de almacenami ento físico al resguardar de documentac ión del proyecto	15%	Perdida de todo la informa ción avanza da del proyect o.	-	Realizar respaldos y copias de seguridad en dispositivos de almacenamie nto externo.	Acudir a software de recuperació n de datos.	Establecer mantenimiento constante de los ordenadores en los que se desarrolla el proyecto.
*	*		13	Se proponen cambios en	60%	El product o no	-	Revisar y hacer un análisis de	-	Realizar un control y seguimiento a

				los requerimientos que requieren rehacer el diseño.		será entregado en la fecha acordada por los cambios a realizar.		cada uno de los requerimientos establecidos y llegar a un acuerdo con el cliente.		cada uno de los requerimientos.
*	*		14	Posible corte de corriente eléctrica y/o desconexión a Internet	30%	Existe un porcentaje de error en el registro de	-	Revisar minuciosamente y manualmente cada registro asociado con las ventas, egresos e	Volver al método anterior de registro manual de pedidos en planillas	Contar con un motor generador de corriente eléctrica

						pedidos y perjudi cará al seguimi ento del flujo de caja		ingresos		
--	--	--	--	--	--	--	--	----------	--	--

Tabla 69 Lista de Riesgos

2.1.2.2.4 Evolución del Plan de Desarrollo del Software

El Plan de Desarrollo del Software se revisará semanalmente y se refinará antes del comienzo de cada iteración.

2.1.2.3 Organización del Proyecto

2.1.2.3.1 Participantes del Proyecto

Director del Proyecto: Antonio Lujan Huanca Mamani

2.1.2.3.2 Interfaces Externas

El entorno gráfico del sistema está pensado en los usuarios, se tomará en cuenta la comodidad y la facilidad de manejo.

El sistema cuenta con una interfaz diseñada en base a módulos debido a que el sistema es muy complejo, requiere bastante precisión, y sobre todo para otorgar una mejor comunicación en el proceso de desarrollo entre el personal responsable del mismo.

Todas las funcionalidades del sistema requieren de una validación de seguridad debido a que se maneja información de uso institucional. La tediosa carga de datos que se realiza bajo una conexión de Internet, debe ser totalmente transparente para el Usuario.

El rendimiento de las interfaces nombradas con anterioridad recae sobre todo en el lenguaje a utilizar y la versión del mismo.

El sistema contara de manera general con:

- Pantalla de Acceso.
- Pantalla de Menú Principal.

2.1.2.3.3 Roles y Responsabilidades

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP.

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos. Planificación y control del proyecto.
Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.
Programador	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario.
Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos,

	preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue.
--	---

Tabla 70 Roles y Responsabilidades

2.1.2.4 Gestión del Proceso

2.1.2.4.1 Estimaciones del Proyecto

2.1.2.4.1.1 Introducción

El objetivo principal del plan de medidas es generar información que sea útil para gestionar el proyecto y tomar decisiones en base a datos objetivos.

2.1.2.4.1.2 Propósito

Se tomarán medidas sobre el proyecto de Gestión Administrativa “Mejorada” Sistema de control de documentación de la Brigada Parlamentaria de Tarija que se está desarrollando.

2.1.2.4.1.3 Alcance

Este plan se limita al proyecto de Gestión Administrativa “Mejorada” Sistema de control de documentación de la Brigada Parlamentaria de Tarija

2.1.2.4.1.4 Visión General

Se van a identificar la información que se desea obtener y qué métricas miden esa información, describiendo con detalle algunas de ellas.

2.1.2.4.2 Objetivos y sub-objetivos de la gestión

Se deben identificar las necesidades de información para el proyecto de Gestión Administrativa “Mejorada”. Varias de estas medidas son necesarias para gestionar riesgos, otras para gestionar el proyecto midiendo tiempos y recursos, otras para comprobar los costes y otras para comprobar la calidad, por ejemplo del diseño realizado antes de comenzar la implementación. Puede haber más casos en los que se

necesite hacer medidas, pero como ejemplo, en este documento sólo veremos algunas métricas representativas.

2.1.2.4.3 Métricas

Categoría de información	Concepto medible	Métrica
Tiempo y progreso	Tiempo y esfuerzo de desarrollo	COCOMO
Tamaño del producto	Tamaño físico	Número de líneas de código
	Tamaño Funcional	Puntos de función Número de casos de uso

Tabla 71 Métricas

2.1.2.4.4 Descripción de las métricas

Nombre	COCOMO
Definición	Métrica de tiempo y esfuerzo de desarrollo
Objetivo	Ayudar a realizar la planificación temporal y de recursos, así como a realizar la gestión del proyecto.
Proceso de Análisis	Aplicar tablas y formulas del método COCOMO para obtener estimaciones de esfuerzo y tiempo.

Tabla 72 Métrica: Cocomo

Nombre	Basado en Casos de Uso
Definición	Métrica de tamaño, Esfuerzo y Funcionalidad

Objetivo	Medir el tamaño de Producto y Funcional de la aplicación y Estimar Esfuerzo
Proceso de Análisis	Aplicar tablas con valores y hacer cálculos siguiendo el método de análisis de estimación basada en Casos de Uso.

Tabla 73 Basado en Casos de Uso

Nombre	Puntos de Función
Definición	Métrica de tamaño y complejidad
Objetivo	Medir el tamaño y complejidad de la aplicación
Proceso de Análisis	Aplicar tablas con valores y hacer cálculos siguiendo el método de análisis de puntos de función.

Tabla 74 Puntos de Función

2.1.2.4.5 Utilización de las Métricas

2.1.2.4.5.1 COCOMO

El tamaño del software varía de unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles de líneas (medio).

Se utilizan dos ecuaciones para determinar el esfuerzo de personal y el tiempo de desarrollo. El coste es:

$$K_m = 2.4 S_k^{1.05}$$

Dónde:

K_m se expresa en personas-mes.

S_k es el tamaño expresado en miles de líneas de código fuente.

El tiempo de desarrollo se da por:

$$t_d = 2.5 K_m^{0.38}$$

Donde K_m se obtiene de la ecuación anterior y t_d es el tiempo de desarrollo en meses.

El N° de Personas Medio se da por:

$$P_e = K_m / t_d$$

Donde K_m se obtiene de la ecuación anterior y t_d se obtiene de la ecuación anterior y P_e es el número de personas promedio.

$$K_m = (2.4)(15)^{1.05} = 41.22 \text{ personas-mes}$$

$$T_d = 2.5 (41.22)^{0.38} = 10.27 \text{ mes}$$

$$P_e = 41.22 / 10.27 = 4.01 \text{ personas}$$

Lo cual se estima que el proyecto se llevara a cabo en 10 meses con 4 personas, Por tanto, el proyecto se realizará en 250 días sin considerar los días domingos y feriados. En este proyecto sin embargo lo desarrolla solo una persona que es el Jefe de Proyecto que tiene que realizar la función de las cuatro personas.

2.1.2.4.5.2 Basado en Casos de Uso

Se cuenta con los puntos de Casos de Uso ajustados y se calcula que para cada punto de casos de uso se requieren 2 horas-hombre. Por lo tanto el esfuerzo se obtiene aplicando la siguiente fórmula:

$$UCP = \text{Punto de Caso de Uso}$$

$$E = UCP * 2 = \text{horas-hombre}$$

$$E = 132 * 2 = 264 \text{ horas-hombre}$$

Usar la contabilización de los Factores ambientales para ajustar las horas-hombre que se requieren por punto de caso de uso, este valor de esfuerzo según la siguiente tabla representa a la actividad de programación, por tanto se debe realizar un nuevo cálculo agregando las actividades restantes relacionadas con el desarrollo del software.

Por lo tanto debería calcularse el esfuerzo de la siguiente manera:

Actividad	Porcentaje	Horas-hombre
Análisis	10%	$10\% * 264 = 26.4$
Diseño	20%	$20\% * 264 = 52.8$
Programación	40%	$40\% * 264 = 105.6$
Pruebas	15%	$15\% * 264 = 39.6$
Sobrecarga (otras actividades)	15%	$15\% * 264 = 39.6$
TOTAL de esfuerzo	100%	$\Sigma = 264$

Tabla 75 Cálculo de Esfuerzo

El tiempo es de 264 horas – hombre por cada Caso de Uso.

En la cual Si realizamos el siguiente calculo

$$264 \text{ horas-hombre} / 24 \text{ horas} = 11 \text{ días por cada Caso de Uso}$$

Es una Estimación que no tiene mucha relevancia por considerar solo los requisitos del cliente capturados en los casos de uso

2.1.2.4.5.3 Puntos de Función

Los puntos de función que obtienen utilizando una función empírica basando en medidas cuantitativas del dominio de información del software y valoraciones subjetivos de la complejidad del software.

Los puntos de función se calculan mediante la siguiente fórmula:

$$PF = CT * (0.65 + 0.1 * \sum Fi)$$

Dónde:

PF= Punto de función

CT= Cuenta Total

Fi= Cuenta de los valores de ajuste de complejidad

Los Puntos de Función se derivan de medidas Directas del dominio de la información

Calculo de la cuenta total:

Parámetro	Cuenta	Factor de Ponderación			Subtotal
		Simple	Medio	Complejo	
Número de entradas de usuario	15	3	4	6	60
Número de salidas de usuario	20	4	5	7	140
Número de peticiones de usuario	15	3	4	6	90
Numero de archivos	3	7	10	15	2
Numero de interfaces externas	2	5	7	10	10
Cuenta Total					351

Tabla 76 Factor de ponderación

Valores de Ajuste de Complejidad

0	1	2	3	4	5
Sin	Incidental	Moderado	Medio	Significati	Esencial

influencia				VO	
------------	--	--	--	----	--

Tabla 77 Valores de ajuste de complejidad

Nº	Parámetros a Evaluar	Valor
1	Comunicación de datos	5
2	Funciones de procesamiento distribuidos	2
3	Objetivos de Performance	4
4	Ejecución del sistema en un entorno operativo utilitario	4
5	Transacciones de datos sobre múltiples entradas	1
6	Entrada de datos	5
7	Copia de seguridad y recuperación de datos fiable	5
8	Actualización de archivos en forma interactiva	5
9	Complejidad de procesamiento interno	3
10	Reusabilidad de código	4
11	Facilidad de instalación	2
12	Facilidad operacional	5
13	Soporte de múltiples instalaciones	4
14	Facilidad de cambio y manejo	5
	S-Total	54

Tabla 78 Parámetros a evaluar

$$PF = \text{Cuenta-Total} * [0.65 + (0.01 * S\text{-Total})]$$

$$PF = 351 * [0.65 + (0.01 * 54)]$$

$$PF = 417.69$$

$$PF = 418$$

La siguiente tabla proporciona estimaciones del número de líneas de código que se necesitan para construir un punto de función en varios lenguajes de programación:

Lenguaje	LDC/PF
Ensamblador	320
C	128
Java	120
Fortran	105
Ada	70
4GL	20
Lenguajes de Iconos	6

Tabla 79 Estimación de cantidad líneas de código en función al lenguaje de programación

En base a esta tabla se pueden establecer las líneas de código (LDC) que sería la estimación para nuestro proyecto.

El lenguaje de programación que desarrollamos para nuestro proyecto es PHP el cual se tomara de Java por hacerlo haci la mayoría entonces tendríamos unas 120 LDC por PF:

$$LDC = 120 * 418$$

$$LDC = 50160$$

Para simplificar el proceso de estimación y utilizar una forma más común para su modelo de estimación, Putman y Myers sugieren un conjunto de ecuaciones obtenidas de la ecuación del software:

$$t = 8.14 (LDC / P)^{0.43} \quad (1)$$

$$E = 180 B t^3 \quad (2)$$

Donde

E = esfuerzo en personas – mes

t = duración del proyecto en meses a años.

B = factor especial de destrezas. Para programas mayores a 60 KLDC, B > 0.39.

P = parámetro de productividad. Para aplicaciones comerciales de sistemas, P > 28000

Aplicando las ecuaciones (1) y (2) a nuestro proyecto obtenemos:

Dónde: B = 1.03 y P = 28000 para nuestro caso.

$$t = 8.14 (50160 / 28000)^{0.43}$$

$$t = 10,46 \text{ meses.}$$

El tiempo estimado el proyecto es de 10 meses.

$$E = 180 * 1.03 * (0.82)^3$$

$$E = 102.22 \text{ personas – mes.}$$

Entonces 102.22 personas – mes./ 10 mes.= 10.22 personas.

El esfuerzo estimado el proyecto según el análisis de Punto de Función es de 10 personas por mes. Pero vale la pena mencionar que solo una persona desarrolla el sistema.

2.1.2.4.6 Plan del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

2.1.2.4.6.1 Plan de las Fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase (para las fases de Construcción y Transición es sólo una aproximación muy preliminar).

Fase	Nro. Iteraciones	Duración
Fase de Inicio	1	50
Fase de Elaboración	1	64
Fase de Construcción	1	61
Fase de Transición	1	53

Tabla 80 Plan de las Fases

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Descripción	Hito
--------------------	-------------

Fase de Inicio	<p>En esta fase desarrollarán los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del cliente /usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.</p>
Fase de Elaboración	<p>En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera release de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también se incluye como hito. La primera iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos. Ambas iteraciones tendrán una duración de una semana.</p>

Fase de Construcción	Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis / Diseño. El producto se construye en base a 2 iteraciones, cada una produciendo una reléase a la cual se le aplican las pruebas y se valida con el cliente / usuario. Se comienza la elaboración de material de apoyo al usuario. El hito que marca el fin de esta fase es la versión de la reléase 2.0, con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada a los usuarios para pruebas beta.
Fase de Transición	En esta fase se prepararán dos releases para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto.

Tabla 81 Hitos

2.1.2.4.6.2 Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto. La siguiente figura ilustra este enfoque, en ella

lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.

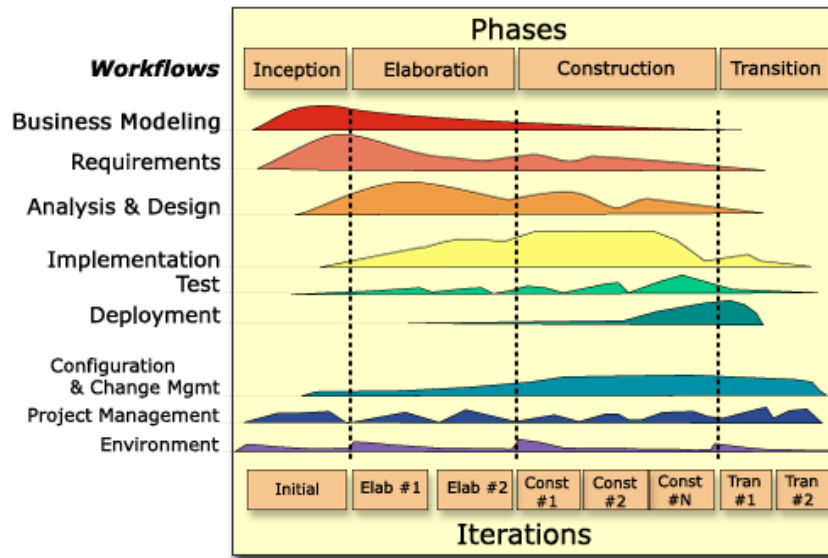


Figura 140 Calendario del proyecto

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

Disciplinas / Artefactos generados o modificados durante la Fase de Inicio	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio		
Requisitos		

Glosario		
Visión		
Modelo de Casos de Uso		siguiente fase
Especificación de Casos de Uso		siguiente fase
Especificaciones Adicionales		siguiente fase
Análisis/Diseño		
Modelo de Análisis/Diseño		siguiente fase
Modelo de Datos		siguiente fase
Implementación		
Prototipos de Interfaces de Usuario		siguiente fase
Modelo de Implementación		siguiente fase
Pruebas		

Casos de Pruebas Funcionales		siguiente fase
Despliegue		
Modelo de Despliegue		siguiente fase
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 1.0 y planes de las Iteraciones		
Ambiente	Durante todo el proyecto	

Tabla 82 Disciplina / Artefactos fase de inicio

Disciplinas / Artefactos generados o modificados durante la Fase de Elaboración	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio		aprobado
Requisitos		
Glosario		aprobado

Visión		aprobado
Modelo de Casos de Uso		
Especificación de Casos de Uso		
Especificaciones Adicionales		
Análisis / Diseño		
Modelo de Análisis / Diseño		Revisar en cada iteración
Modelo de Datos		Revisar en cada iteración
Implementación		
Prototipos de Interfaces de Usuario		Revisar en cada iteración
Modelo de Implementación		Revisar en cada iteración
Pruebas		
Casos de Pruebas Funcionales		Revisar en cada iteración
Despliegue		

Modelo de Despliegue		Revisar en cada iteración
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 2.0 y planes de las Iteraciones		Revisar en cada iteración
Ambiente	Durante todo el proyecto	

Tabla 83 Disciplina / Artefactos fase de elaboración

2.1.2.4.7 Seguimiento y Control del Proyecto

2.1.2.4.7.1 Control de Plazos

Fase	Nº de Iteración	Inicio	Fin
Inicio	Primera	05-03-2018	06-04-2018
Elaboración	Primera	06-04-2018	25-05-2018
Construcción	Primera	16-07-2018	05-10-2018
Transición	Primera	06-10-2018	09-11-2018

Tabla 84 Control de plazos

2.1.2.4.7.2 Control de Calidad

2.1.2.4.7.2.1 Introducción

El control de calidad es un modelo planeado y sistemático de todas las acciones necesarias para proporcionar la confianza de que el artículo o producto se ajuste a los requisitos técnicos establecidos (IEE83).

La preparación de un plan de control de calidad del software para cada proyecto de software es una de las principales responsabilidades del grupo de control de calidad del software.

El control de calidad realizará las siguientes funciones:

Durante el análisis y diseño, se presentaran un plan de verificación del software y un plan de prueba de aceptación. El plan de verificación describe los métodos que se ocuparan para revisar que los documentos de diseño satisfagan los requisitos, y que el código fuente sea consistente con las especificaciones de requisitos y con la documentación del diseño.

El plan de prueba del código fuente es un componente importante del plan de verificación del software.

El plan de prueba de aceptación incluye casos de prueba, resultados esperados y capacidades demostradas por cada caso de prueba. A menudo, el personal de control de calidad trabajara con el cliente para desarrollar un solo plan de prueba de aceptación. En otros casos el cliente desarrollara un plan de prueba de aceptación independiente del plan de control de calidad. De cualquier forma, el personal de control de calidad debe desarrollar un plan de prueba de aceptación doméstico.

Al terminar los planes de verificación y de aceptación se realizara una revisión de verificación del software para evaluar cuan adecuados son los planes.

Durante la evolución del producto, se realizaran Auditorias en el proceso para verificar que los productos de trabajo sean consistentes y estén completos.

Los elementos que sufrirán auditoría por consistencia incluyen especificaciones de interfaces para hardware, software y personas; diseño interno contra especificaciones funcionales; código fuente contra documentación. En la práctica, solo ciertas porciones críticas del sistema pueden someterse a auditorías intensivas.

Antes de la entrega del sistema, se realizan una auditoría funcional y una auditoría física.

La primera reconfirma el cumplimiento de todos los requisitos. La auditoría física verifica que el código fuente y todos los documentos asociados estén completos, sean consistentes tanto internamente, como uno con otro, y que estén listos para enviarse. El resumen de verificación del software se repara para describir los resultados de todas las revisiones, auditorías y pruebas efectuadas por el personal de control de calidad, a través del ciclo de desarrollo.

Dicho personal, a veces se encarga de los acuerdos para los recorridos, inspecciones, y revisiones de logros principales. Además, el personal de control de calidad conduce el proyecto póstumo, escribe es documento del legado del proyecto, y proporciona una custodia a largo plazo de los registros del proyecto.

El grupo de control de calidad trabajara con el grupo de desarrollo para obtener el plan de pruebas y el código fuente, que especifica los objetivos de las pruebas, los criterios para la terminación de las pruebas, el plan de integración del sistema, los métodos que se usaran en módulos particulares, además, entradas de prueba particulares y resultados esperados.

El plan de pruebas de código fuente tiene cuatro tipos de pruebas: pruebas de función, de desempeño, de tensión, y estructuradas. Las dos primeras se basan en las especificaciones de requisitos y se diseñaron para demostrar que el sistema satisface sus requisitos los que a su vez se deben redactar en términos cuantificables y que se puedan probar.

Los casos de prueba funcional especifican condiciones operativas comunes valores de entradas comunes y resultados esperados comunes, también prueban el comportamiento dentro, sobre, y más allá de las fronteras funcionales.

Las pruebas de desempeño se proyectan para verificar el tiempo de respuesta, rendimiento, la utilización de memorias primarias y secundarias y las tasas de tráfico en los canales de datos y los enlaces de comunicación.

Las pruebas de tensión se diseñan para sobrecargar un sistema de varias maneras.

Las pruebas de estructura se relacionan con el examen de la lógica interna de procesamiento de un sistema de software. Las rutinas particulares llamadas y las rutas lógicas recorridas a lo largo de las rutinas son los objetos importantes.

2.1.2.4.7.2.2 Propósito

- Detectar problemas.
- Delimitar el área problemática.
- Estimar factores que probablemente provoquen el problema.
- Determinar si el efecto tomado como problema es verdadero o no.
- Prevenir errores debido a omisión, rapidez o descuido.
- Confirmar los efectos de mejora.
- Detectar desfases.
- Realizar pruebas en cada versión.

2.1.2.4.7.2.3 Objetivos

- Aumentar la satisfacción del Cliente.
- Equilibrar el esfuerzo en múltiples demandas.
- Obtener el mejor producto.
- Disfrutar de una ventaja competitiva.
- Disponer de métricas objetivas de valoración.
- Ahorrar tiempo y dinero.

2.1.2.4.7.2.4 Resumen de las actividades de garantía de calidad

Listado de las actividades de garantía de la calidad que se llevarán a cabo durante el proyecto.

2.1.2.4.7.2.4.1 Seguimiento de la administración del SQA

ID	Propósito	Sub Actividad	Detalle / Indicación
1	Criterios de Entrada	- Ninguna.	Plan de Desarrollo y proceso de Software y Personal.
2	Revisión	<ul style="list-style-type: none"> - Examinar estructura gerencial de la organización. - Identificar tareas de cada integrante de la gerencia. - Definir responsabilidades a cada integrante de la 	Verificar consistencia de la estructura organizacional con las responsabilidades asignadas en Plan de desarrollo de Software.
3	Criterios de Salida	- Estructura de la administración revisada.	Estructura organizacional de la gerencia óptima para el proyecto.

Tabla 85 Seguimiento de la administración del SQA

2.1.2.4.7.2.4.2 Seguimiento de la Documentación

ID	Propósito	Sub Actividad	Detalle / Indicación
1	Criterios de Entrada	- Ninguna.	Plan de Desarrollo de Software.

2	Revisión	<ul style="list-style-type: none"> - Revisión y análisis del plan de documentación. - Buscar discrepancias. - Discutir discrepancias con el 	Reportar discrepancias según documento presentado por el docente y estándares.
3	Criterios de Salida	<ul style="list-style-type: none"> - Documentación revisada. 	Documentos de acuerdo a Estándar, y sin discrepancias.

Tabla 86 Seguimiento de la documentación

2.1.2.4.7.2.4.3 Seguimiento de la adherencia a los Estándares

ID	Propósito	Sub Actividad	Detalle / Indicación
1	Criterios de Entrada	<ul style="list-style-type: none"> - Ninguna. 	Documentos, plan de desarrollo de Software.
2	Documentación	<ul style="list-style-type: none"> - Monitorear adherencias de los documentos a los estándares. 	Chequear documento desarrollado basado en la metodología RUP.
3	Diseño	<ul style="list-style-type: none"> - Monitorear adherencias del diseño a los estándares. 	Chequear documento UML del sistema.
4	Codificación	<ul style="list-style-type: none"> - Monitorear adherencias de la codificación a los estándares. 	Revisar de acuerdo a Patrones de Diseño MVC.
5	Métricas	<ul style="list-style-type: none"> - Revisar la métrica definida. 	Revisar de acuerdo al estándar Puntos de Función y otros.
6	Criterios de Salida	<ul style="list-style-type: none"> - Proceso de Documentación revisado. - Proceso de Diseño revisado. 	Discrepancias reportadas y solucionadas. Documentos de acuerdo a estándares.

Tabla 87 Seguimiento de la adherencia a los Estándares

2.1.2.4.7.3 Gestión de Riesgos

2.1.2.4.7.3.1 Relacionados con el tamaño del producto

Es posible no poder llegar a terminar todos los componentes del proyecto debido a que estos son muchos o grandes.

2.1.2.4.7.3.2 Relacionados con el impacto en la organización

No entregar el proyecto en el tiempo estimado.

El posible no uso del software debido a los requerimientos de equipos.

Poco uso del software.

Gran cantidad de documentación a entregar al Cliente.

Los límites legales y gubernamentales en cuanto al desarrollo y funciones del proyecto.

Posibles errores en el producto y costos asociados.

2.1.2.4.7.3.3 Relacionados con el tipo de Cliente

Si se tiene la disponibilidad de tiempo para la especificación formal de requerimientos.

Si están dispuestos a participar de las pruebas o revisiones.

Si se relacionará de forma ágil con el grupo de desarrollo.

2.1.2.4.7.3.4 Relacionados con la definición del proceso de producción

El software a ser utilizado en el control del desarrollo del proyecto.

Las herramientas a ser utilizadas en el análisis y diseño.

2.1.2.4.7.3.5 Relacionados con el entorno de desarrollo

Si hay herramientas de gestión de proyectos.

Hay herramientas de prueba apropiadas.

Generadores de código para la aplicación.

2.1.2.4.7.3.6 Relacionados con la tecnología

Es una nueva tecnología.

El hardware con el que debe interactuar es nuevo o cumple las expectativas.

La base de datos a ser utilizada ha sido probada y tiene la funcionalidad y rendimiento.

Las interfaces son especializadas.

La necesidad de nuevos componentes.

2.1.2.4.7.3.7 Relacionados con la experiencia y tamaño del equipo

Es el mejor personal disponible.

Los miembros tienen las técnicas apropiadas.

Hay suficiente gente disponible.

El personal está comprometido a lo largo de desarrollo del proyecto.

Tiene el personal las expectativas correctas del trabajo.

2.1.2.4.7.4 Gestión de Configuración

Se llevara a cabo una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. También se incluirían la gestión de las solicitudes de cambio y de las modificaciones que estas produzcan, informando y publicando dichos cambios a todos los participantes del proyecto.

.

2.2 Componente 2: Socialización del sistema automatizado y las TIC ejecutado e implementado.

2.2.1 Introducción

De acuerdo al modo de vida actual del mundo, el término Capacitación y Sistemas de Información están modificando la forma de trabajo de las empresas, los sistemas de Información ayudan a acelerar los procesos, por tanto, las organizaciones que los implantan logran ventajas competitivas al adoptarlos en sus funciones. Pero si el personal no está debidamente capacitado la producción de las empresa será deficiente y en mucho casos incompleto.

La capacitación se refiere a las metodologías que se usan para proporcionar a las personas dentro de la empresa, las habilidades necesarias que necesitan para realizar su trabajo de una manera más eficiente, esto contempla desde pequeños cursos que le permitan al usuario entender el funcionamiento básico del sistema nuevo, hasta capacitación más profunda y avanzadas a bases de prácticas y material didáctico como libros (Tutoriales) y otros.

La capacitación es un proceso que lleva a la mejora continua de la producción y con esto implantar nuevas formas de trabajo, como en este caso el manejo adecuado del Sistema Informático “SURD” que ayuda a la Gestión de documentos de la Brigada parlamentaria de Tarija.

2.2.2 Objetivos

2.2.2.1 Objetivo General

Lograr la participación y aportación por parte de las personas, haciendo uso de las nuevas tecnologías de Información y Comunicación

2.2.2.2 Objetivo Específico

- Fomentar la utilización del Sitio Web desarrollado.

- Preparar a los colaboradores para la ejecución de las diversas tareas y responsabilidades de la organización.

2.2.3 Aspectos Pedagógicos de la Socialización

2.2.3.1 Introducción

Recordando que la socialización es un proceso mediante el cual el individuo adopta los elementos socioculturales de su medio ambiente y los integra a su personalidad para adaptarse a la sociedad, por este motivo se pretende socializar el Proyecto con el objeto de generar flujos de comunicación y participación de la sociedad en nuestra cultura.

De esta forma el proyecto denominado “Gestión Administrativa Mejorada” se fortalece y se garantiza un tiempo de vida prolongado.

2.2.3.2 Objetivos

Hacer conocer a la Población sobre la existencia del proyecto

Fomentar la utilización y participación de la sociedad en el Sitio Web diseñado.

2.2.3.3 Metodologías y Estrategias

La socialización se la realizara mediante Propaganda y Publicidad.

Las metodologías utilizadas están relacionadas a la situación económica y al impacto que tendrán en la sociedad, por que tomo la decisión de utilizar las siguientes:

Publicidad Directa

Es la que se pone en contacto (sin agentes intermediarios) con el consumidor.

Publicidad Indirecta

En cambio esta no se dirige personalmente a cada consumidor, sino que utiliza los medios de comunicación masiva que hacen llegar su mensaje a todo un sector de la comunidad.

2.2.4 Aspectos Pedagógicos de la Capacitación

2.2.4.1 Definición de Capacitación

La capacitación es un proceso educacional de carácter estratégico aplicado de manera organizada y sistémica, mediante el cual los colaboradores adquieren o desarrollan conocimientos y habilidades específicas relativas al trabajo, y modifica sus actitudes frente a los quehaceres de la organización, el puesto o el ambiente laboral.

2.2.4.2 Importancia de la Capacitación

Permite que:

- Consolidación en la integración de los miembros de la organización.
- Mayor identificación con la cultura organizacional.
- Disposición desinteresada por el logro de la misión empresarial.
- Entrega total de esfuerzo por llegar a cumplir con las tareas y actividades.
- Mayor retorno de la inversión.
- Alta productividad.
- Promueve la creatividad, innovación y disposición para el trabajo.
- Mejora el desempeño de los colaboradores.
- Desarrollo de una mejor comunicación entre los miembros de una organización.
- Reducción de costos.
- Aumento de la armonía, el trabajo en equipo y por ende de la cooperación y coordinación.
- Obtener información de fuente confiable, como son los colaboradores.

2.2.4.3 Beneficios de la Capacitación

Como beneficia la capacitación a las organizaciones:

- Conduce a rentabilidad más alta y a actitudes más positivas.
- Mejora el conocimiento del puesto a todos los niveles.

- Eleva la moral de la fuerza de trabajo.
- Ayuda al personal a identificarse con los objetivos de la organización.
- Crea mejor imagen.
- Mejora la relación jefes-subordinados.
- Es un auxiliar para la comprensión y adopción de políticas.
- Se agiliza la toma de decisiones y la solución de problemas.
- Promueve al desarrollo con vistas a la promoción.
- Contribuye a la formación de líderes y dirigentes.
- Incrementa la productividad y calidad del trabajo.
- Ayuda a mantener bajos los costos.
- Elimina los costos de recurrir a consultores externos

2.2.4.4 Medios a utilizar

2.2.4.4.1 Aspectos Técnicos

- 1 Computador con el Sistema Desarrollado.
- Diapositivas de Exposición hechas en PowerPoint.

2.2.4.4.2 Aspectos Logísticos

- Material digital informativo (cds,dvds)

2.2.4.5 Conclusiones

Una vez realizado la capacitación y según la ejecución del contenido de la capacitación, se pudo diferenciar tipos de cambio en el personal, como por ejemplo el desarrollo de destreza y conocimiento de manera directa respecto al tema de Internet; además se desarrolló un cambio de actitud negativa por actitudes más favorables y positivas entre el personal de la Brigada Parlamentaria en relación de la difusión de la información por medio de Internet

2.2.4.6 Medios de Verificación

- Carta de conformidad por medio de la presidenta de la Brigada Parlamentaria de Tarija Jacqueline Mendieta.
- Fotografías tomadas durante las capacitaciones.

III. Capítulo III: Conclusiones y Recomendaciones

3.1 Conclusiones

De acuerdo al trabajo realizado y a medida que se desarrolló este proyecto, permite llegar a las siguientes conclusiones:

- Para Abordar el Proyecto el Marco Lógico fue una herramienta indispensable, desde el diseño e identificación cuál es el problema, la definición qué debemos hacer, cómo debemos hacerlo, la ejecución y supervisión lo estamos haciendo bien, hasta la evaluación lo hemos logrado.

Los principales beneficios de usar el Marco Lógico en la definición y diseño del proyecto incluyen: Mantener la línea de trabajo, mejor Comunicación, Objetivos claros, mejor Desempeño y Administración del proyecto.

- La Metodología RUP aplicada en el desarrollo del proyecto juega un papel preponderante es una de las metodologías sin duda esencial en un proyecto y en el paso inicial, que debe encajar en el equipo, guiar y organizar actividades que conlleven a las metas trazadas en el proyecto.

Entre Algunas de las ventajas que podemos mencionar son:

- Evaluación en cada fase que permite cambios de objetivos.
- Funciona bien en proyectos de innovación.
- Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- Seguimiento detallado en cada una de las fases.

Y entre las desventajas:

- La evaluación de riesgos es compleja
- Excesiva flexibilidad para algunos proyectos
- Estamos poniendo a nuestro cliente en una situación que puede ser muy incómoda para él.
- Nuestro cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él.

- Para la conclusión del proyecto se tuvo que realizar varios estudios para el manejo correcto de registro y control de documentación (Anexos).
- Para la implementación del sistema Web se utilizará el servicio de Hosting local o por defecto un hosting gratuito por lo tanto para poder tener acceso a la base de datos y dar mantenimiento a la misma por medio del Sistema de escritorio, éste deberá conectarse al servidor Host elegido.

3.2 Recomendaciones

Para un mejor desempeño del proyecto realizado sería aconsejable implementar un componente de socialización y difusión sobre la introducción de las TIC en las diferentes instituciones, con el fin de que el presente proyecto no quede solo en una iniciativa, y sea un modelo para futuros proyectos.

Se recomienda implementar el Sistema Propuesto, para ello se considera necesario buscar al personal mejor calificado para que ejerza un alto grado de control y análisis para que este Sistema desarrolle sus funciones en forma efectiva y segura. Es fundamental que se les brinde una completa y constante capacitación.

Se debe tener en cuenta un plan o programa de mantenimiento tanto para el equipo de cómputo como para el software. Ya que, los requerimientos como las necesidades de la institución van cambiando, el software, también, debe hacerlo. Y si un equipo no se encuentra en óptimas condiciones repercutirá en el desempeño del software.

Para los usuarios finales es recomendable que antes de empezar a usar el sistema, lean detenidamente el manual de usuario, pues, en este se explica el funcionamiento del mismo y no vayan a “probar” como comúnmente se dice.

GLOSARIO

Calidad

Conjunto de características que debe cumplir un servicio o producto en base a normativa preestablecida y que responda a las expectativas y necesidades de los usuario del servicio.

Parlamentario

Un miembro del parlamento, es el representante de los votantes en un parlamento. En congresos bicamerales, esta categoría incluye específicamente a los miembros de la cámara baja.

Ciudadano

Un ciudadano es una persona considerada como miembro de un estado titular de derechos civiles y políticos, sometido a sus leyes.

Documento

Es cualquier documento formulado por cualquier medio expresivo, como medio específico para probar, describir, comunicar y/o informar algo dentro de la estructura, la función, la actividad, y los tramites de cualquier entidad pública en cualquier parte y en cualquier época.

Documento Interno

Son aquellos que se originan y manejan al interior de la Institución.

Documento Externo

Son aquellos que provienen de diferentes organismos nacionales e internacionales, instituciones públicas, privadas y/o personas naturales.

Registro

Medio por el cual se escribe la Información de la correspondencia.

Derivación

Proceso de asignación por parte de un funcionario activo al responsable de dar respuesta a un determinado asunto.

Documento en Proceso

Conjunto de asuntos y/o expedientes ya atendidos, cuya atención se encuentra aún en curso.

Documento Finalizado

Conjunto de asuntos y/o expedientes ya atendidos, concluidos.