

I. Capítulo I

1.1 Presentación del Proyecto

1.1.1 Título del proyecto

“Mejoramiento del control del historial clínico odontológico de los pacientes de la clínica dental IRACEMA, a través de las TIC.”

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)

Clínica dental “IRACEMA”.

1.2 Personal vinculado al proyecto

1.2.1 Director del proyecto

Aguirre	Gudiño	Eduardo Luis	7143945 Tja.
Apellido Paterno	Apellido Materno	Nombre	CI
Estudiante Univ.	Ing. Informática	Ciencias y Tecnología	
Profesión	Carrera o Unidad	Facultad	
70217902	edu_94lui@hotmail.com		
Celular	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	Eduardo Luis Aguirre Gudiño	Estudiante en Ing. Informática	7143945 Tja.	
Tutor	Ing. Lizeth Camacho Adautt	Ing. Informático	5032287	

Tabla 2 Participantes del equipo de trabajo

1.2.3 Equipo de trabajo de:

Empresas/Instituciones/Organizaciones

participantes/cooperantes

Nombre: Clínica Dental "IRACEMA"			
Dirección: C. Félix Soto N° 1531 casi esq. Circunvalación		Teléf. Oficina: 66-37102 - 72995645	
Nombre y Apellidos	Cargo	C.I.	Firma
Patricia Iñiguez A.	Gerente y odontóloga – Mat. Prof. I-93	5023192	

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

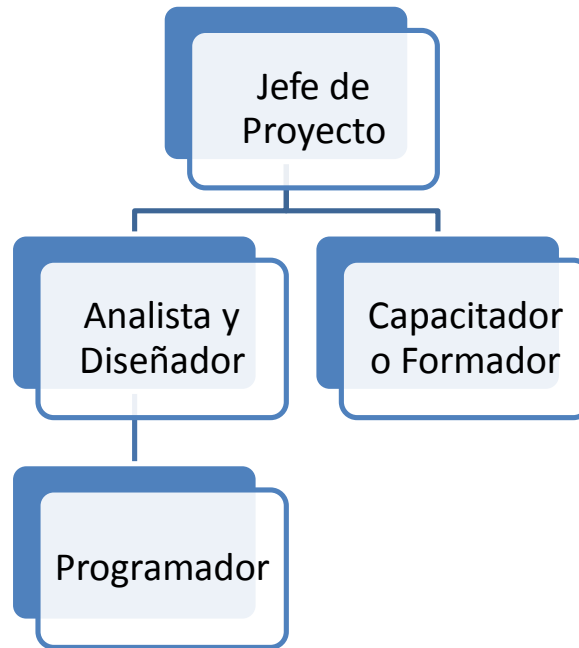


Figura 1 Organigrama del Equipo de Proyecto

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, el campo de la salud 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 sus pacientes, viendo que cada día el avance tecnológico es constante especialmente por el área de salud y sus diferentes especialidades las cuales facilitan el tratamiento de sus pacientes, el siguiente proyecto presenta la propuesta del desarrollo de un sistema web para un mejor control de la atención de los pacientes, en el cual se encargue de la administración del historial clínico de dicho paciente teniendo una información segura y confiable que beneficiará a todos los pacientes que pertenezcan a la clínica dental “IRCEMA” de la ciudad de Tarija.

El objetivo principal del presente proyecto es Desarrollar e Implementar un Sistema Automatizado para el control del historial clínico odontológico de los pacientes de la Clínica Dental “IRACEMA” de la Gerente y Odontóloga Dra. Patricia Iñiguez A. 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

En la actualidad el sistema de administración del historial clínico odontológico de los pacientes de la clínica dental IRACEMA es de forma manual y carece de un sistema que facilite de manera eficiente y eficaz el manejo de dicho historial clínico.

La gerente reconoce que existe la morosidad en el momento de atención a los pacientes por causa de la búsqueda de historiales clínicos, la pérdida de información en el momento del manejo de las mismas, la ocupación de espacio en las oficinas teniendo en cuenta que se la maneja en archivadores, la organización de los mismos es incompleto a causa de mucha información conservada.

El proyecto tiene como propósito mejorar la gestión de información que se produce en la clínica dental. 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 pacientes de la clínica dental IRACEMA aporta al manejo de datos de pacientes de forma oportuna, ordenada y disponible para realizar reportes.

Las TIC son medios que aportar 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 clínica, sin dejar atrás las normas y formalidades de trabajar con ellas.

1.3.2.1 Análisis de causas de problemas

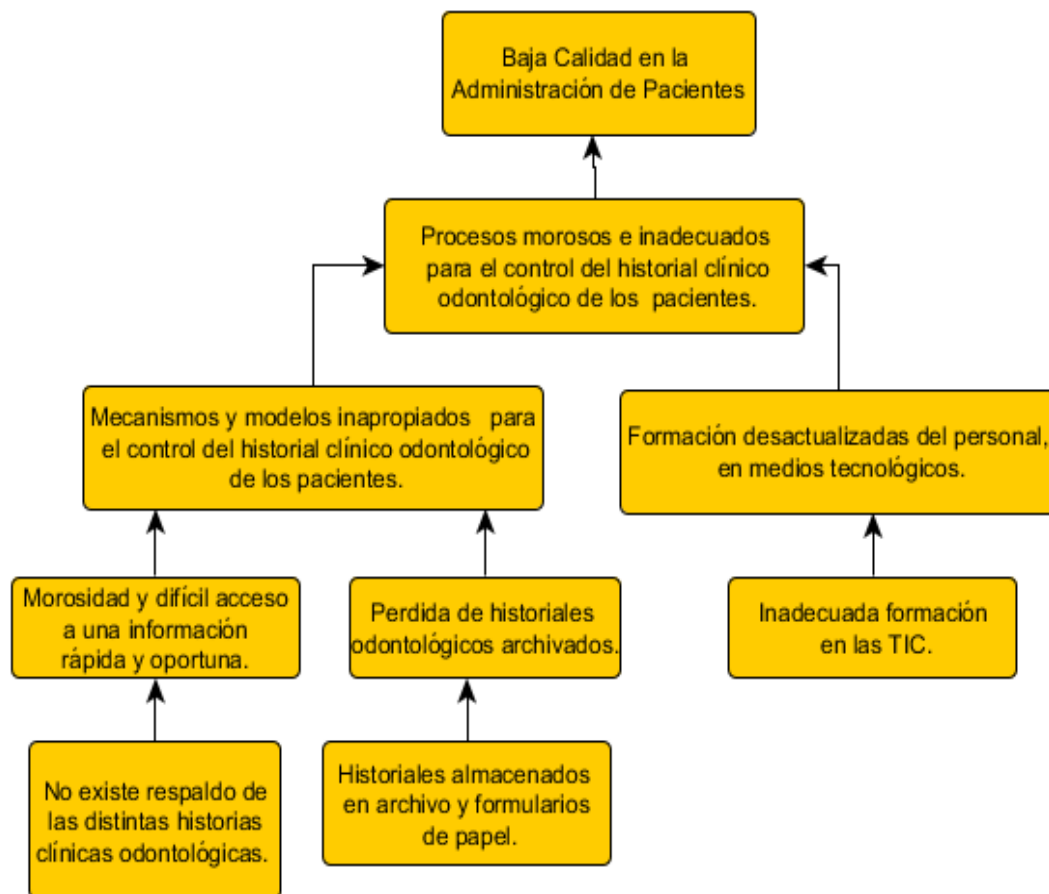


Figura 2 Árbol de problema

1.3.2.2 Análisis de Objetivos

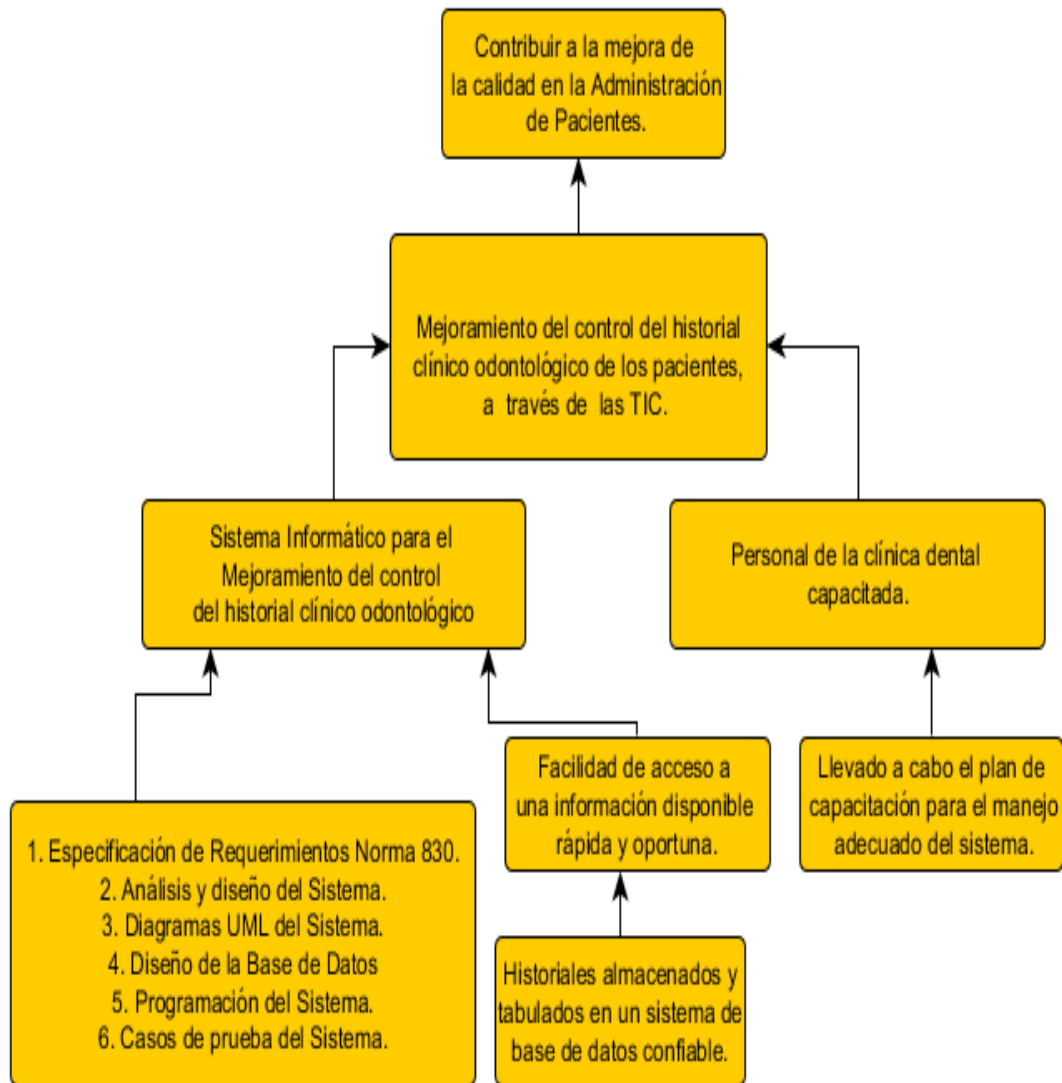


Figura 3 Árbol de Objetivo

1.3.3 Objetivos

1.3.3.1 Objetivo General

Mejorar el control de la historia clínica odontológica de los pacientes de la clínica dental “IRACEMA”, 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 del historial clínico odontológico de pacientes de la clínica dental IRACEMA.
- 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 del historial clínico odontológico Pacientes 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 él lenguaje UML (Unified Modeling Language).

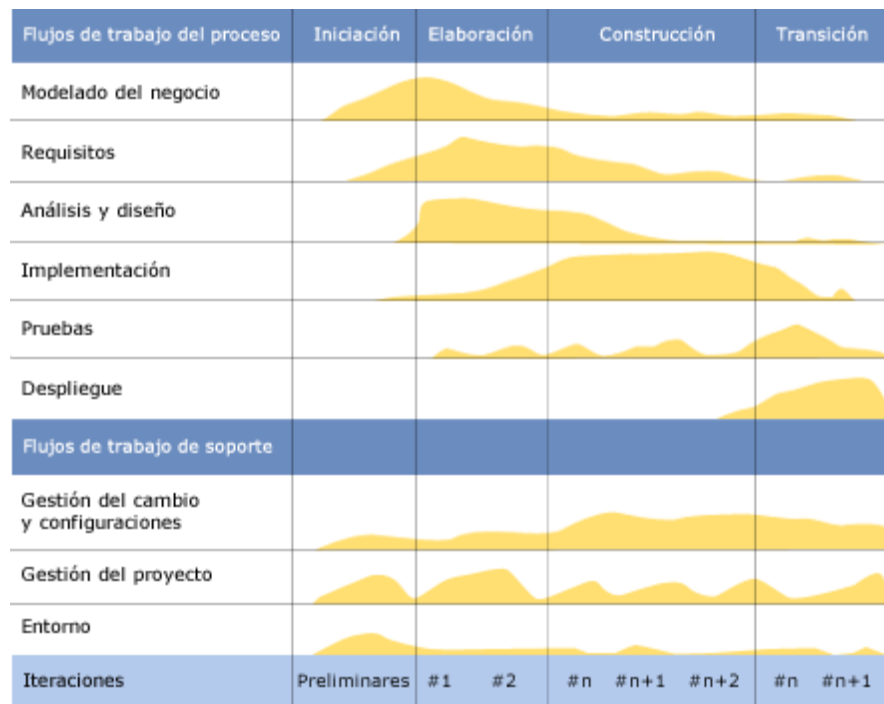


Figura 4 Ciclo de Vida de RUP

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.

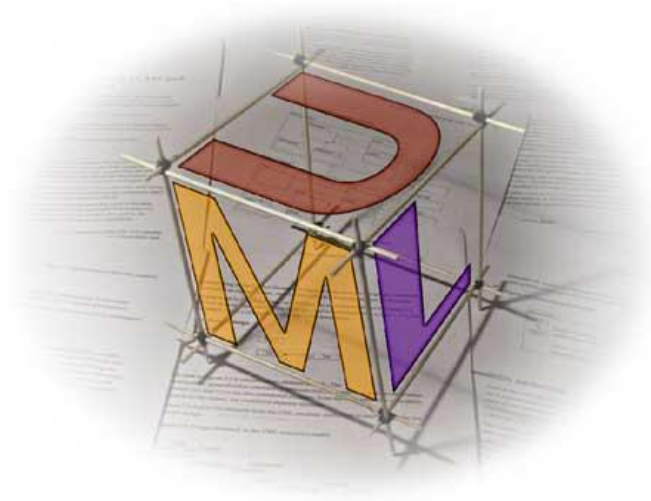


Figura 5 Logotipo de UML

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.

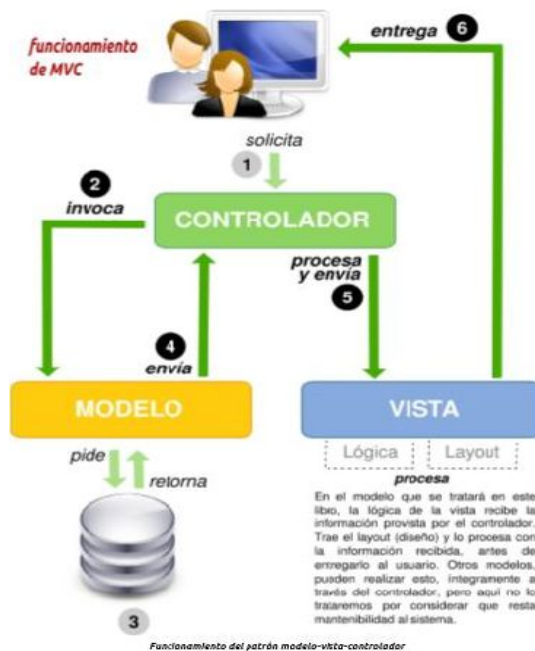


Figura 6 Patrón Arquitectónico MVC

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 (aun que 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 Fusion 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 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.
- Laravel: 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, **Laravel** fue diseñado siguiendo la filosofía que prima la convención sobre la configuración.

Laravel 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. Laravel 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 pacientes.

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:

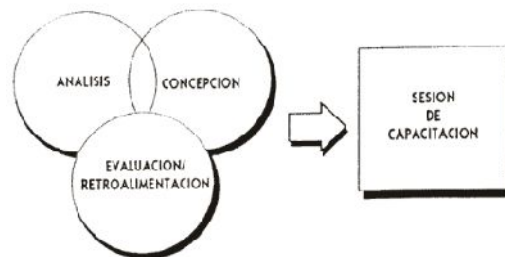


Figura 7 Modelo ACER

- **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
<p>Fin</p> <p>Contribuir a la mejora de la calidad en la Administración de Pacientes.</p>	<p>Al final del 1º año de ejecución del proyecto, el 51% de los pacientes atendidos por el servicio de la clínica dental manifiesta un alto grado de satisfacción por los servicios de salud provistos.</p>	<p>Resultados de la encuesta anual realizada por la Gerente de la Clínica dental; Teniendo de 10 encuestas, si de 5 + 1 son positivas se tiene el 51%</p>	<p>La clínica dental cuenta con medios tecnológicos suficientes para el desarrollo de sus funciones.</p>
<p>Objetivo General (Propósito)</p> <p>Mejoramiento del control del historial clínico odontológico de los pacientes, a través de las TIC.</p>	<p>Al concluir con el proyecto el 60% de los procesos asociados a la clínica dental se han automatizado, a partir del 2016 con relación al año base 2015.(NroPacientesporHora/NroPacientesporDia)X100</p>	<p>Cuadro comparativo de la cantidad de procesos del control del historial clínico dental vs. Cantidad de procesos automatizados</p>	<p>Aceptación de la implementación del sistema automatizado por parte de la gerente y odontóloga que es la</p>

		avalado por la Gerente General.	máxima autoridad de la clínica dental.
Objetivos Específicos (Componentes)			
<p>1. Sistema Informático para el Mejoramiento del control del historial clínico odontológico, desarrollado.</p>	<p>Al finalizar el proyecto se ha desarrollado el sistema certificado por los estándares en desarrollo de aplicaciones web y que cumplan con la norma IEE830.</p>	<p>* Análisis de requerimientos</p> <p>* Diseño Documentado del sistema automatizado.</p> <p>* Pruebas realizadas al sistema automatizado.</p>	<p>Disponibilidad de brindar información oportuna por parte de la clínica dental.</p> <p>Coordinación entre el equipo de trabajo del proyecto y el usuario final.</p>
<p>2. Socialización del sistema automatizado y las TIC ejecutado e implementado</p>	<p>Al terminar el proyecto el personal involucrado se encuentra capacitado un 75% para el manejo del sistema web.</p>	<p>Resultados de encuestas; Teniendo de cada encuesta, cuenta con 20 preguntas si de 15 se tienen afirmativas se tendrá el 75%</p>	<p>Los involucrados captan los conceptos de las nuevas tecnologías y las ponen en práctica.</p>
Actividades	Resumen presupuesto		
<p>1.0. Estudio de modelos y Metodologías.</p> <p>- Relevamiento de datos.</p>		<p>Informes del presupuesto de gastos.</p>	<p>* Se dispone de hardware de última generación.</p>

<ul style="list-style-type: none"> - Desarrollo del modelo. - Verificación del modelo. 1.1. Modelo del negocio 1.2. Requerimientos 1.3. Análisis y Diseño 1.4. Implementación 1.5. Pruebas 2.1. Análisis 2.2. Conceptuación 2.3. Evaluación – Retroalimentación - Análisis - Concepción - Evaluación – - Retroalimentación 	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">20000 Servicios No personales</td> <td style="width: 40%; text-align: right;">13120.-</td> </tr> <tr> <td>30000 Materiales y suministros</td> <td style="text-align: right;">310.-</td> </tr> <tr> <td>Total:</td> <td style="text-align: right;">Bs. 13430,0.-</td> </tr> </table>	20000 Servicios No personales	13120.-	30000 Materiales y suministros	310.-	Total:	Bs. 13430,0.-	<p>Cronogramas e informes de la capacitación.</p> <p>Manuales de usuario y de sistema.</p> <p>Testimonios de los involucrados en el desarrollo delo sistema.</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>
20000 Servicios No personales	13120.-								
30000 Materiales y suministros	310.-								
Total:	Bs. 13430,0.-								

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

Aguirre Apellido Paterno	Gudiño Apellido Materno	Eduardo Luis Nombre	7143945 Tja C.I.
27/02/1989 Fecha de Nacimiento	M...X... F..... Sexo	B/ Narciso Campero C/ Felix Soto Nro. 1503 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	2009
Ing. Informática - 3º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2010
Ing. Informática - 4º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2010
Ing. Informática - 5º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2011
Ing. Informática - 6º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2011
Técnico Medio de Programador de Aplicaciones	“INSTITUTO COMERCIAL SUPERIOR TARIJA INCOS”	Bolivia	2012
Ing. Informática - 7º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2012
Ing. Informática - 8º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2012

Ing. Informática – 9º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2013
Ing. Informática – 10º Semestre	“Universidad Autónoma Juan Misael Saracho”.	Bolivia	2013
Técnico Superior en Sistemas Informáticos	“INSTITUTO COMERCIAL SUPERIOR TARIJA INCOS”	Bolivia	2014
Contabilidad General – 1º Año	“INSTITUTO COMERCIAL SUPERIOR TARIJA INCOS”	Bolivia	2015
Contabilidad General – 2º Año	“INSTITUTO COMERCIAL SUPERIOR TARIJA INCOS”	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	6° Semestre	2011
Ing. Informática	7° Semestre	2012
Ing. Informática	8° Semestre	2012
Ing. Informática	Anual	2013
Ing. Informática	Laboratorio	2014

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
IT Training Course (Scrath)	Universidad Autónoma Juan Misael Saracho.	20	2011
XIX Congreso Nacional de ciencias de la computación – Conferencias – Tutoriales - Ponencias.	Universidad Autónoma Gabriel Rene Moreno.	60	2012

Introducción al desarrollo web (iDESWEB)	Universidad de Alicante	36	2013
Formación en Valores Éticos y Morales.	Universidad Autónoma Juan Misael Saracho.	60	2014
Diseño, Organización y Evaluación de videojuegos y gamificación	Universidad Europea de Madrid	30	2014
XXI Congreso Nacional de ciencias de la computación – Conferencias – Tutoriales.	Universidad Autónoma Juan Misael Saracho	60	2014
Robots y Videojuegos en las aulas: Scratch y Arduino para profesores	Universitat Pompeu Fabra de Barcelona	25	2015

Tabla 11 Cursos Realizados

II. Capítulo II

2.1 Componente 1: Sistema Informático para el Mejoramiento del control del historial clínico odontológico

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 Clínica Dental “IRACEMA”

2.1.1.2.1.1 Misión

Brindar servicios odontológicos integrales contribuyendo a resolver eficazmente los problemas de salud Oral de nuestros pacientes, aplicando para ello conceptos de excelencia en el servicio, eficiencia en la gestión y calidez personalizada en la atención.

2.1.1.2.1.2 Objetivo General

El principal objetivo es dar un servicio impecable tanto en el aspecto técnico como en el aspecto humano, intentando que la estancia del paciente en nuestra clínica sea lo más agradable posible. Para conseguir estas metas ponemos a disposición de nuestros pacientes un equipo humano con gran experiencia y altamente cualificado.

2.1.1.2.1.3 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

2.1.1.2.1.4 Diagnóstico Estratégico

El análisis realizado permite detectar que la clínica dental IRACEMA se encuentra compitiendo en un mercado complejo ya que la necesidad que satisface la clínica es susceptible a diversos competidores, convirtiéndose esto en una falencia de la clínica.

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

Es importante destacar que la preocupación de la salud bucal de las personas favorece a este tipo de clínica por la atención a amable y eficiente.

Es imprescindible que la empresa cuente con talento humano calificado que le permita desarrollar sus actividades de la mejor manera logrando así la satisfacción plena de quienes confían en el servicio ofrece.

Además que el Know How de la empresa para la atención de los tratamientos le permite crear una ventaja competitiva que se transforma en la percepción aceptable que las personas tiene de la empresa.

Debido al crecimiento del departamento en los últimos años Tarija se ha visto beneficiada con la cantidad de personas en el departamento aumenta significativamente, por tanto la clínica dental IRACEMA tiene la oportunidad de llegar a este publico debido a la estética, amabilidad y calidad (higiene y normas) de sus servicios.

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.

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

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 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).

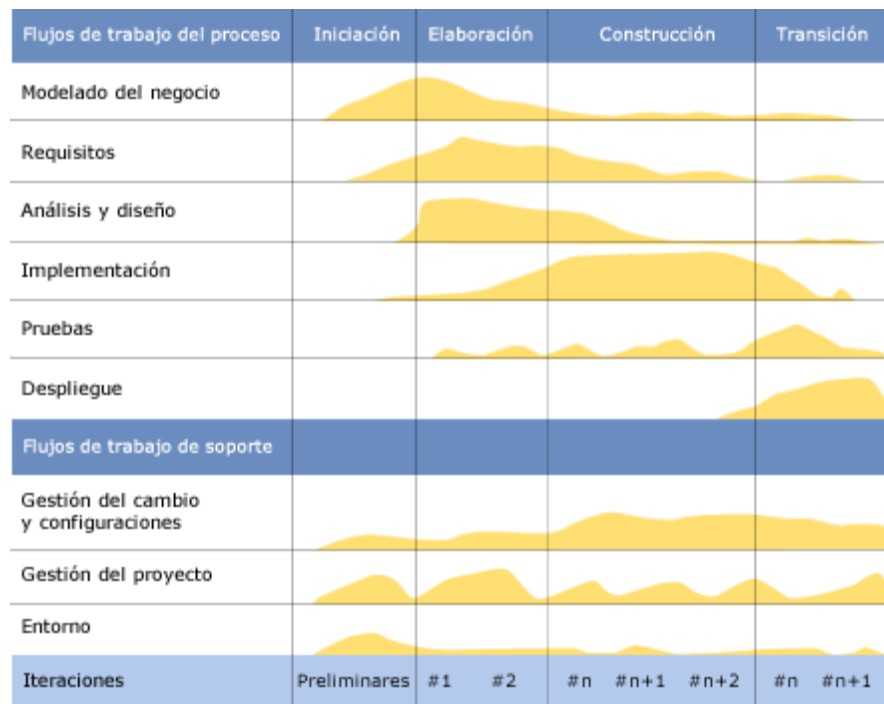


Figura 9 Fases en el Ciclo de Vida del RUP

- 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-costos, 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 mas 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

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.

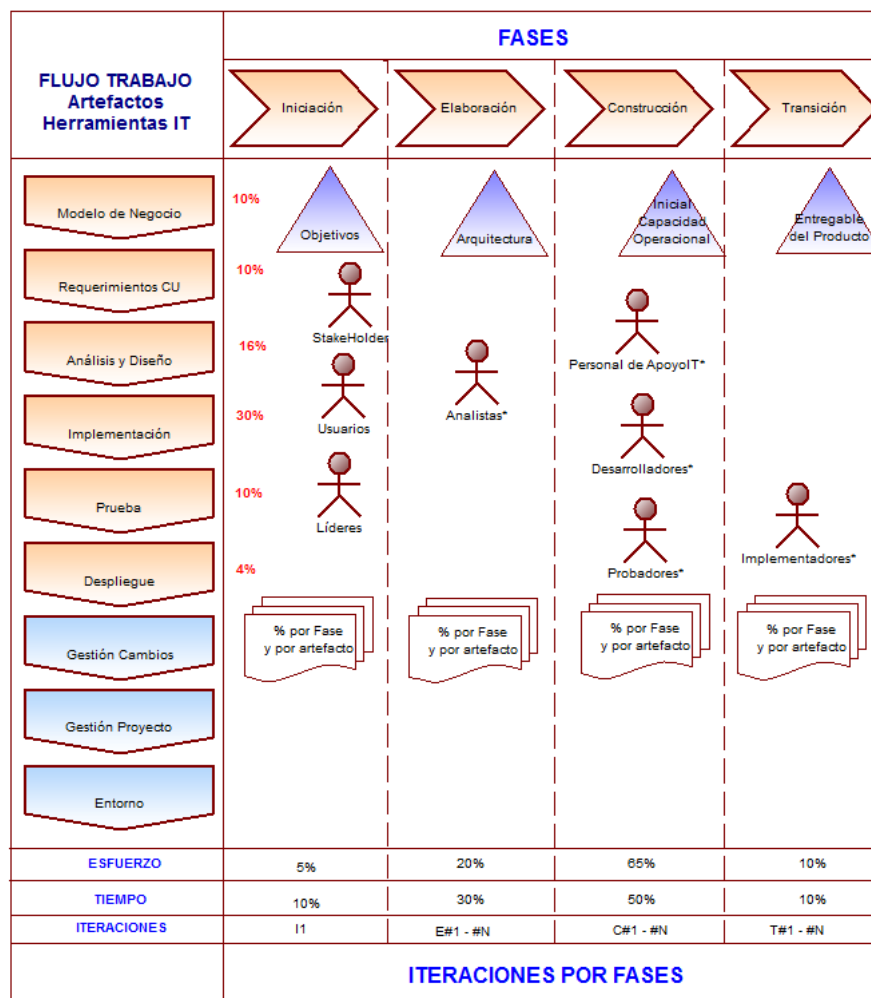


Figura 10 Flujo de Trabajo del RUP

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 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.

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

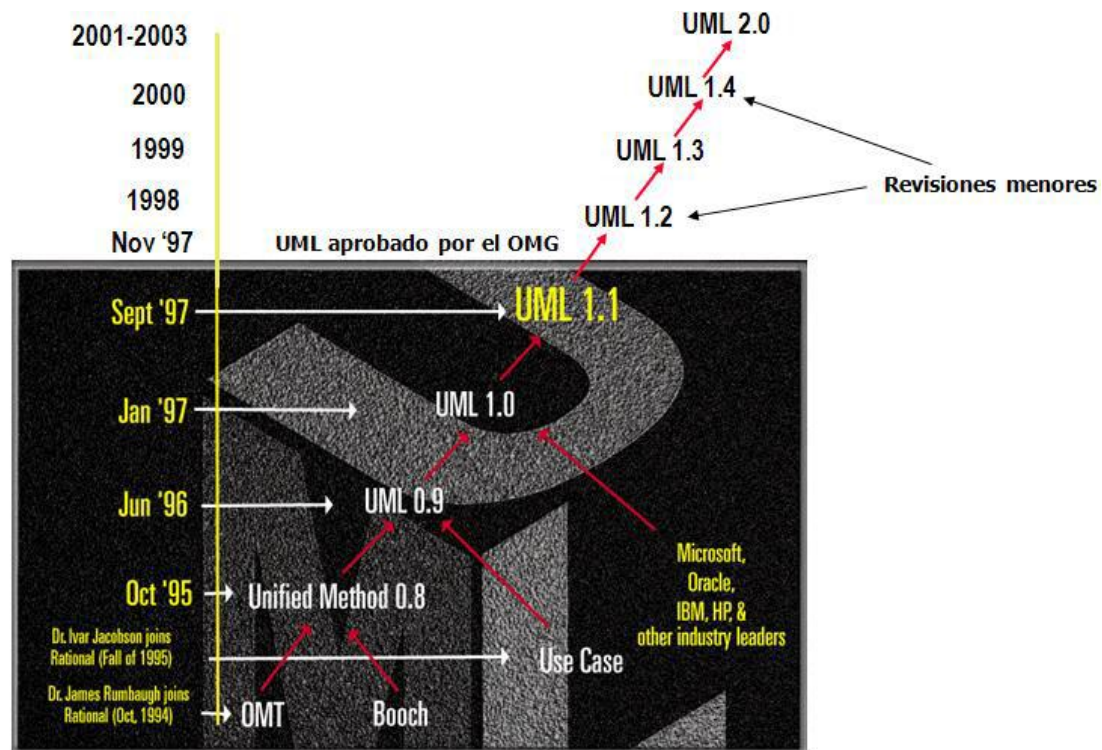


Figura 11 Fases históricas del UML

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 representadas.

- 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.

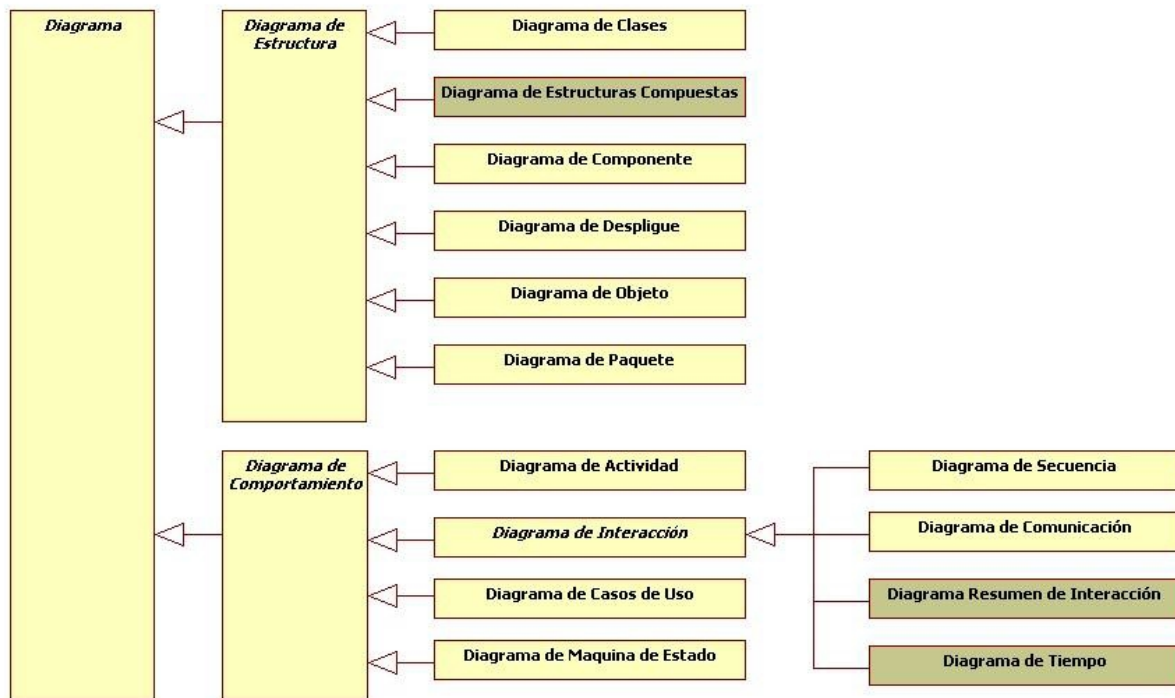


Figura 13 Diagramas en UML

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.

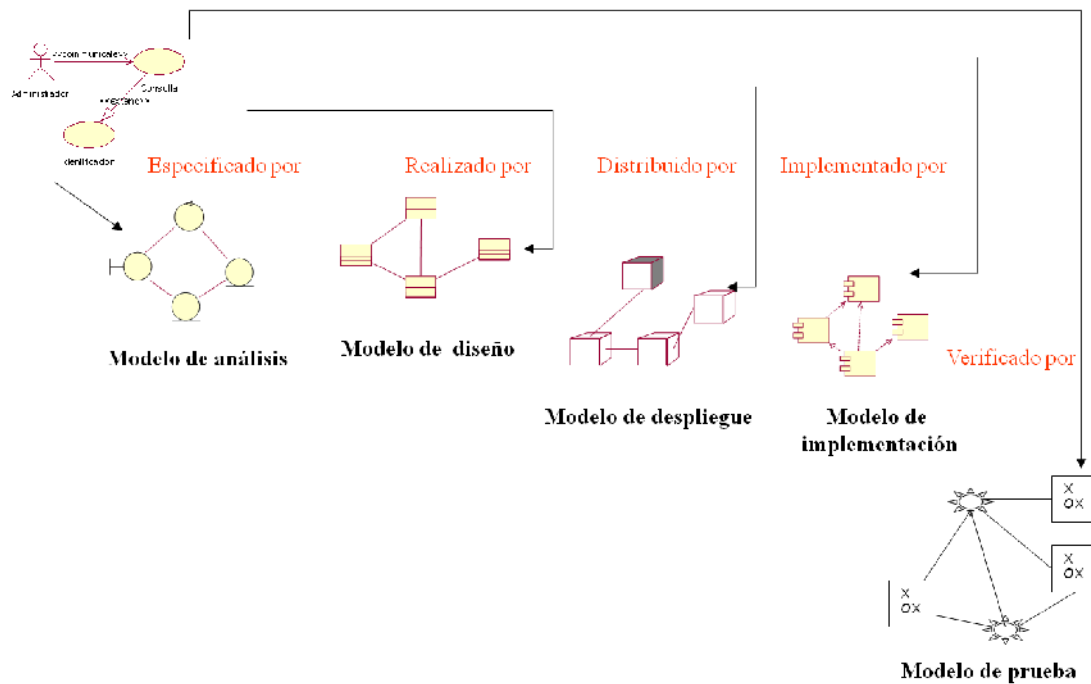


Figura 14 Dependencia de los demás modelos de los casos de uso

- 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.

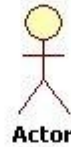


Figura 15 Diagrama de Caso de Uso: Artefacto - ACTOR

- **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 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 delimitada 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.

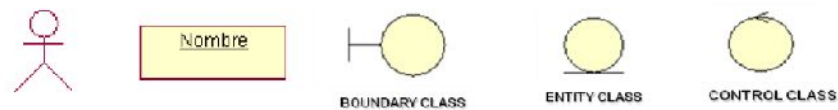


Figura 18 Diagrama de Secuencia: Artefacto – OBJETO

- **LÍNEA 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.

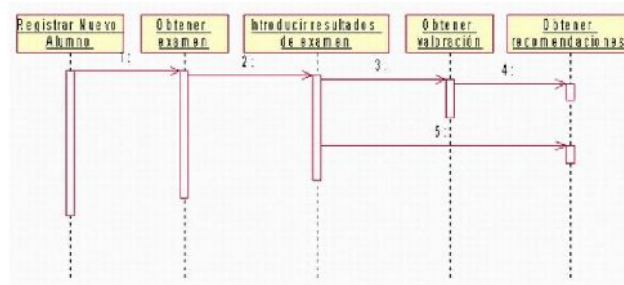


Figura 19 Diagrama de Secuencia: Artefacto - LÍNEA DE VIDA

- **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.

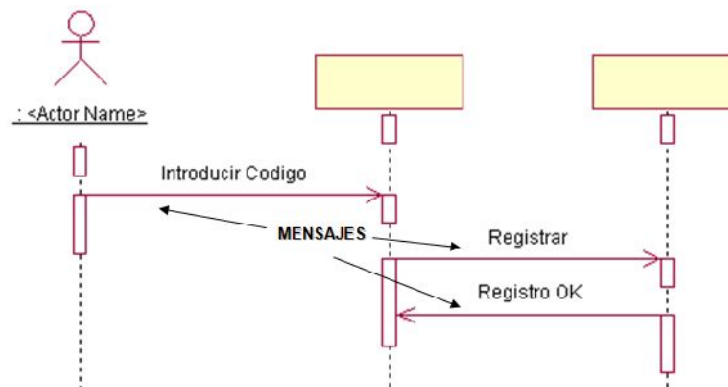


Figura 20 Diagrama de Secuencia: Artefacto - MENSAJE

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.

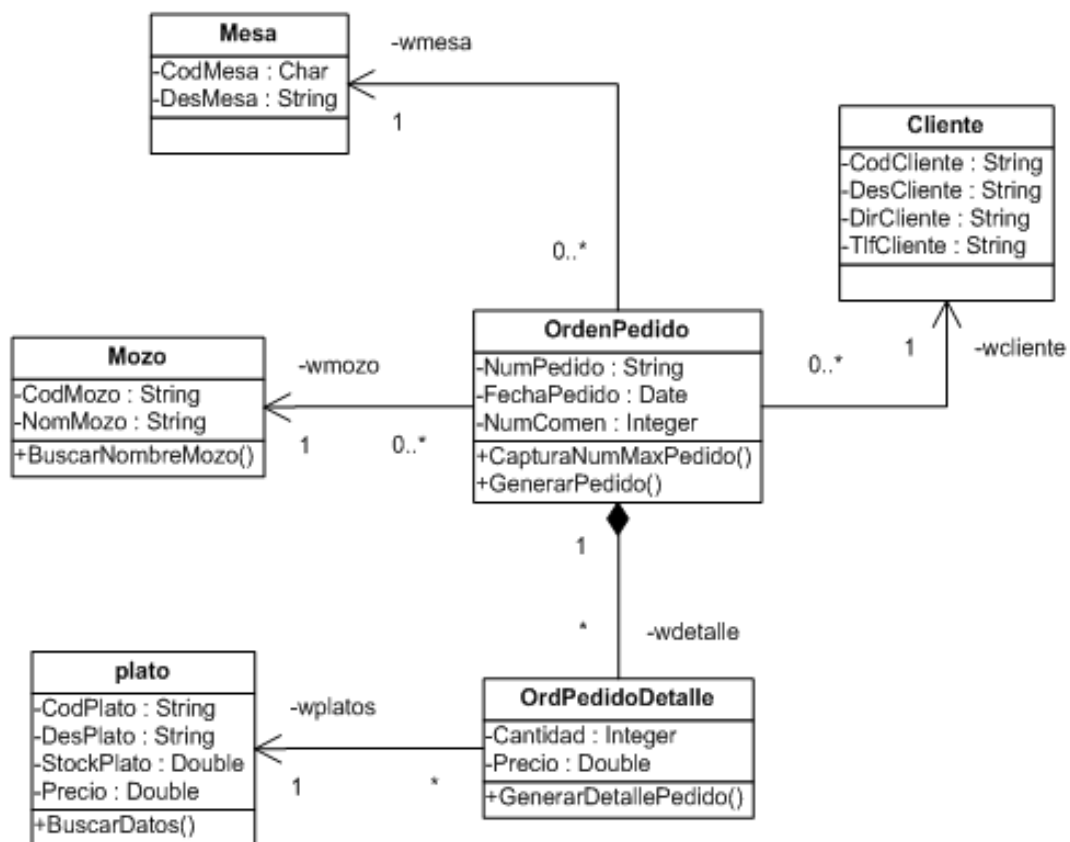


Figura 21 Diagrama de Clases: Modelo Lógico

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.

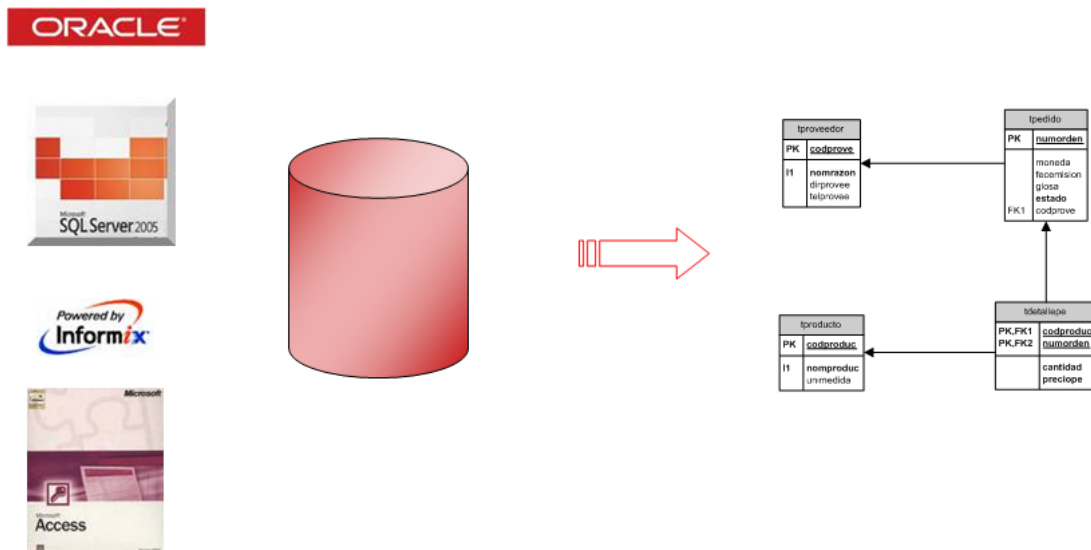


Figura 22 Diagrama de Clases: Modelo de Datos

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.

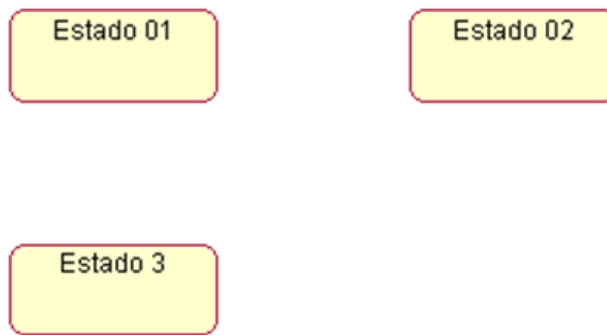


Figura 23 Diagrama de Actividad: Artefacto - ESTADO

- **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.

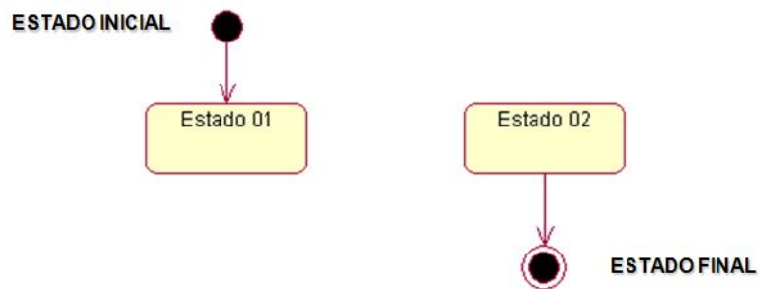


Figura 24 Diagrama de Actividad : Artefacto - ESTADO INICIAL Y FINAL

- **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.

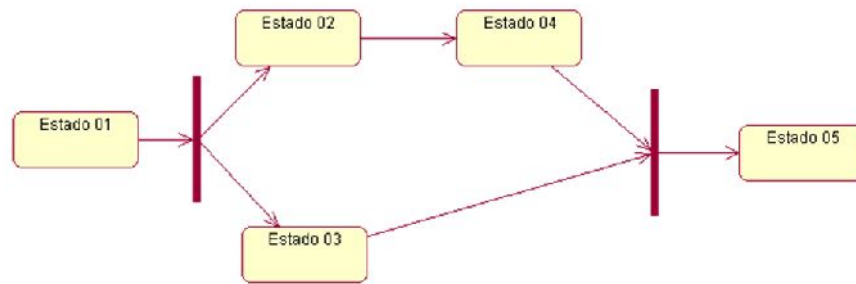
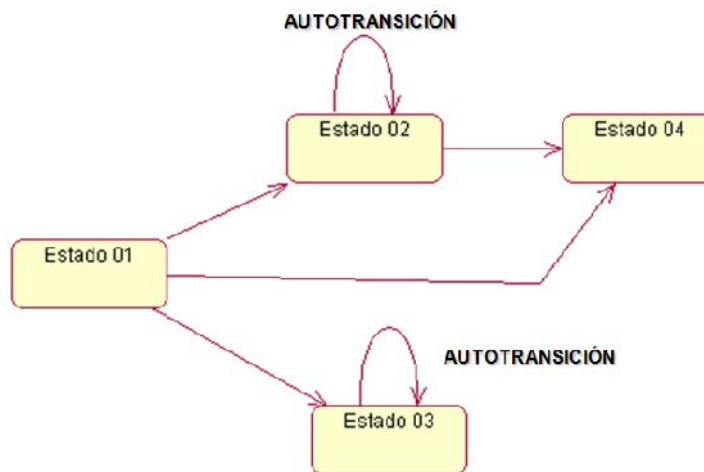


Figura 27 Diagrama de Actividad : Artefacto – TRANSICION COMPLEJA

- **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 sus .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.

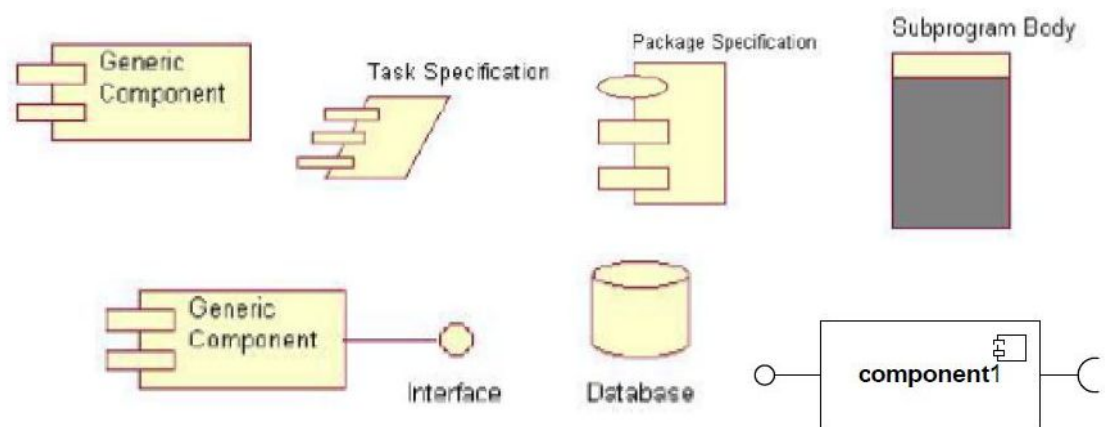


Figura 28 Diagrama de Componentes: Artefacto - COMPONENTE

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:

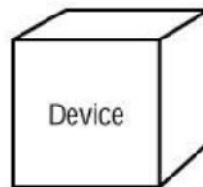


Figura 29 Diagrama de Despliegue: Artefacto – NODO

- **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 2

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

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

componentes de XAMPP. Pero XAMPP no da la opción para configurar la seguridad de MySQL y XAMPP.



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.

⁵ www.enterprisearchitect.org/docs/



Figura 32 Herramientas de Construcción de Software: Enterprise Architect

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.

⁶ dev.mysql.com/



Figura 33 MySQL

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 Laravel⁷

Laravel es un framework de desarrollo web MVC escrito en PHP de última generación, que reduce el tiempo y costo del desarrollo inicial y también su mantenimiento. Aporta una sintaxis expresiva muy clara en la forma de escritura y un completísimo set de funcionalidad que salvan hora y horas de trabajo.

Como otros muchos framework MVC, Laravel fue diseñado siguiendo la filosofía que prima la conversión sobre la configuración.



Figura 34 Framework Laravel

¿Qué significa?

Laravel tiene una forma rígida como se debe escribir, como llamar a los modelos, vista y controladores.

Busca un punto de encuentro entre minimalismo y funcionalidad que descansa en la propia filosofía ZEN a la que aspiran los desarrolladores que ven la programación como algo mucho más que escribir líneas y líneas de código, sencillo de entender con una base de código muy pequeño, ofrece soluciones de modo limpio, simple y elegante, los desarrolladores que ofrecen mucho tiempo trabajado en PHP.

Y por último el Motor de Plantillas BLADE que hace las vistas mucho más limpias y sencillas e incluye importantes mejoras en la presentación de la aplicación.

⁷ www.laravel.fr

2.1.1.5.3 ORM Eloquent⁸

Un ORM es un paquete realmente útil. Son las siglas para Mapeo objeto-relacional (Object Relational Mapper). La parte de mapeo, es porque se esta mapeando los objetos PHP o clases a tablas y filas de la base de datos. La parte Relacional quedara más clara en la sección de relaciones.

Eloquent viene con Laravel y puede ser usada sin modificar nada. La implementación activerecord de Laravel llamado Eloquent interactúa con la base de datos de un modo totalmente orientado a objetos, siguiendo los estándares mas modernos, con Eloquent podemos crear, recuperar, actualizar y borrar los registros de la base de datos sin necesidad de escribir ni una sola línea de código sql y además de esto Eloquent provee con un poderoso sistema de manejo de relaciones y puede incluso gestionar la paginación automáticamente por nosotros.

Con Eloquent podemos definir las relaciones ‘uno-a-uno’, ‘uno-a-muchos’ y ‘muchos-a-muchos’ simplemente añadiendo algunos métodos sencillos a los modelos de Eloquent.

- `has_one` (no a uno).
- `has_many` (uno a muchos).
- `has_many_and_belongs_to` (muchos a muchos).

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.

⁸ CODE HAPPY - Desarrollo de Aplicaciones con el framework de php Laravel – Capitulo 11

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 BD⁹

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.

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.

⁹ www.phpmyadmin.net/

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 Automatizado¹⁰

2.1.1.7.1 La Internet

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

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.

¹⁰ blogs.rimed.cu/gestion-informacion/2010/12/20/tic-y-gestion-de-la-informacion/

Hay 50 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: Eduardo Luis Aguirre Gudiño 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 clínica dental IRACEMA.

2.1.2.2.1.1 Propósito

Con el presente proyecto se pretende Mejorar “el control del historial clínico odontológico de la clínica dental IRACEMA”.

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:

- Modulo Administrador.
- Modulo Odontólogo.

- Modulo Paciente.

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, pero lo más importante mejorará el sistema actual.

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

Mejorar el “control del historial clínico odontológico de la clínica dental IRACEMA”.

2.1.2.2.1.3.2 Objetivo Específico

- Obtener un sistema para el uso en la gestión de los historiales de los pacientes de la clínica.
- Crear una base de datos de acuerdo a las necesidades de la clínica.
- Brindar información al usuario sobre los tratamientos y diagnósticos realizados a los pacientes.
- Implementar una interfaz para que el paciente pueda realizar un seguimiento de sus tratamientos y tener a la mano su historial clínico odontológico.

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 clínica dental IRACEMA, cuenta con:

- La clínica dispone equipos de computación, que incluye un computador, impresora en la recepción.
- Total disponibilidad de la gerente para proporcionar información.
- Personal con conocimientos básicos de computación.
- Presupuesto suficiente para la implementación del Sistema.
- Interés de la institución para utilizar el Sistema.
- Tener una línea telefónica habilitada para poder recibir y confirmar las asignaciones de citas.
- 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.

2.1.2.2.2.2 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 no abarca:
 - Control de Inventarios de material fungible.
 - Registros radiográficos.
 - Control del personal.
 - Control del área contable.
 - Gestión de tratamientos generales
 - 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 del Odontólogo

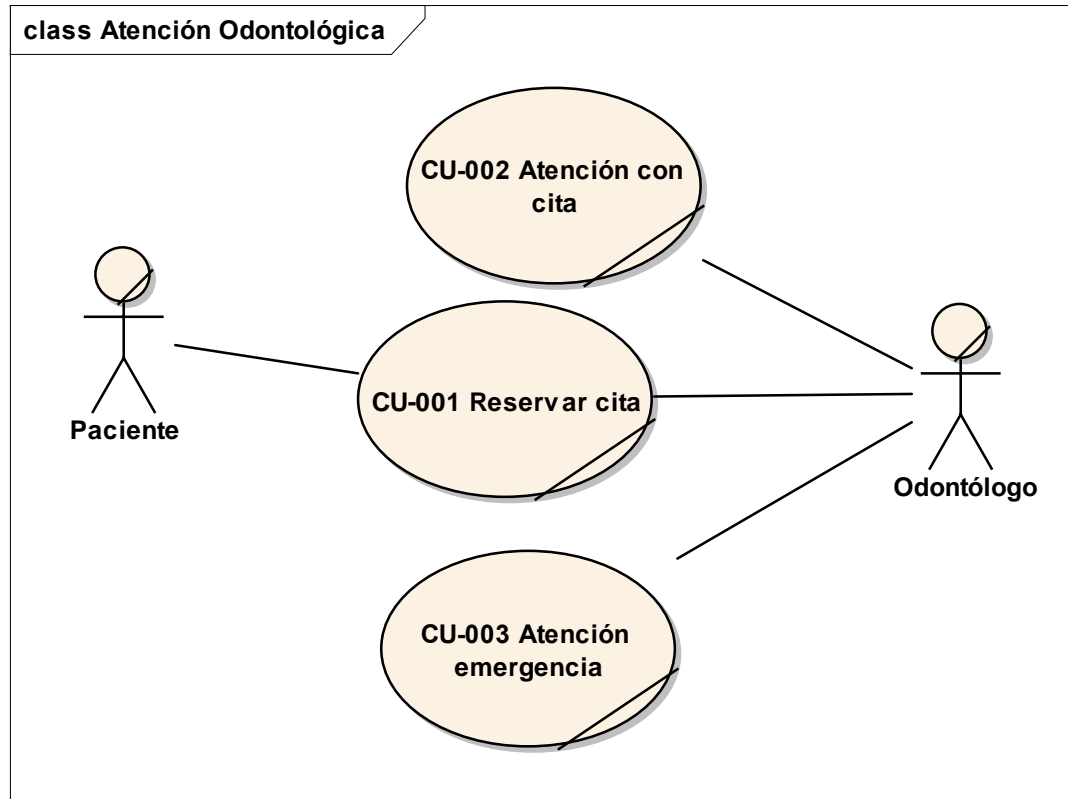


Figura 35 Modelo de Casos de Uso del Negocio del odontólogo

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:

Reservar cita

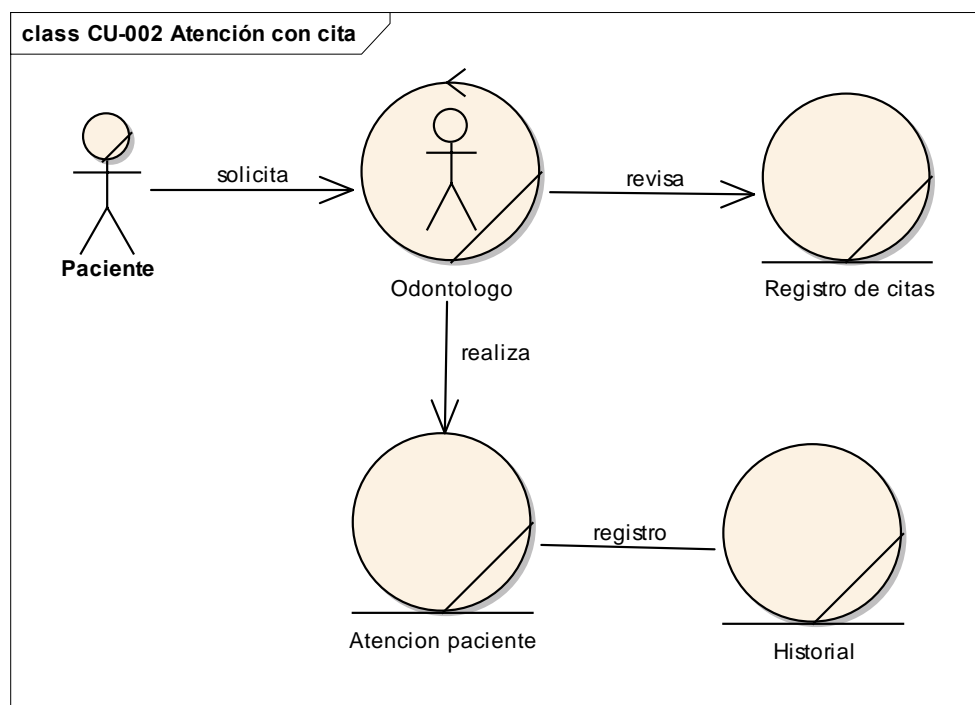


Figura 36 Modelo de Objeto del Negocio: Reservar cita

2.1.2.2.3.3.4.2 Modelo de Objeto del Negocio:

Atención con cita

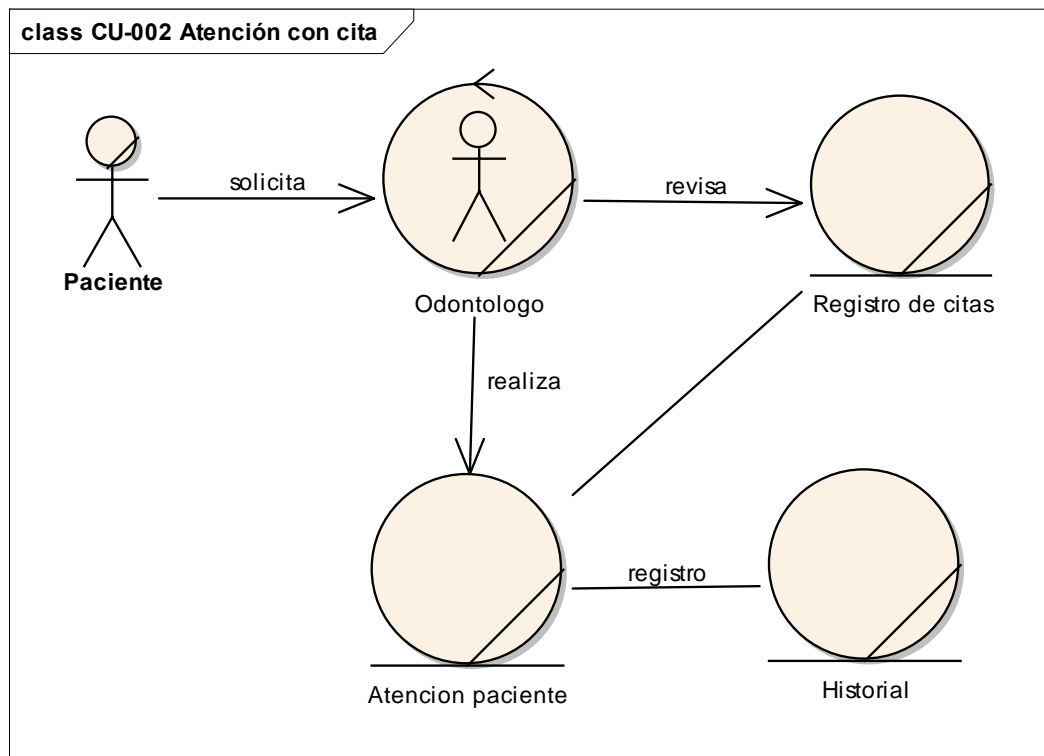


Figura 37 Modelo de Objeto del Negocio: Atención con cita

2.1.2.2.3.3.4.3 Modelo de Objeto del Negocio:

Atención emergencia

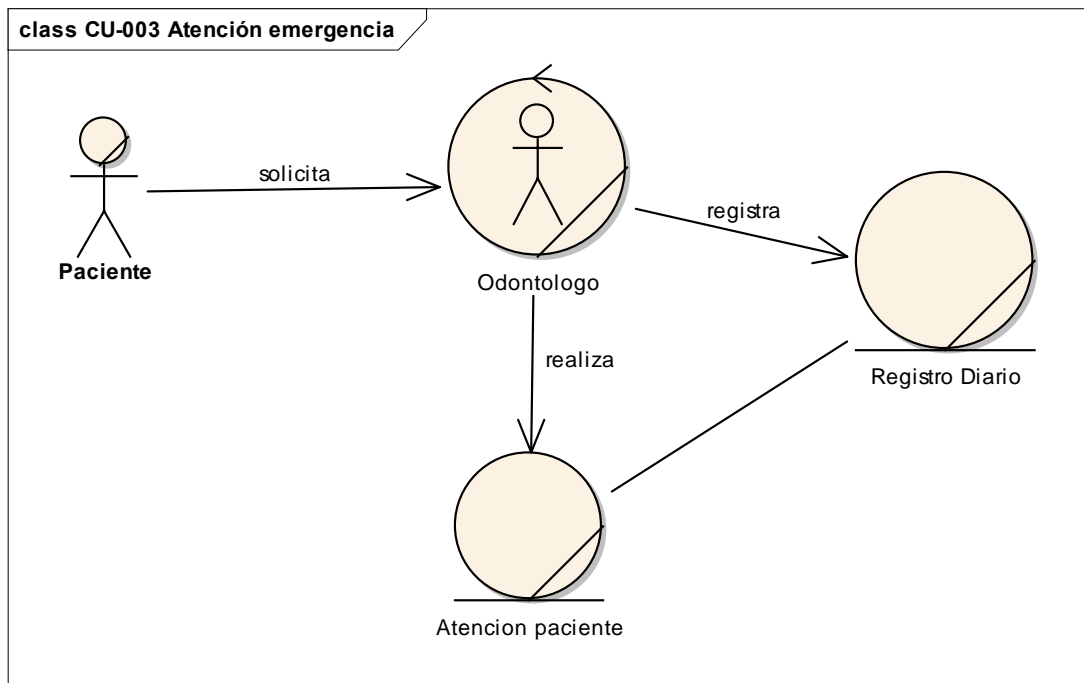


Figura 38 Modelo de Objeto del Negocio: Atención emergencia

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 de forma ascendente según ordenación alfabética tradicional.

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 usuario del servicio.

2.1.2.2.3.4.5.2 Cavidad Bucal

Boca; desde los labios, paladar duro y blando, carrillos, lengua, piso de boca dientes y encías.

2.1.2.2.3.4.5.3 Cirugía

Procedimiento quirúrgico bajo inflamación anestésica aplicada locamente para tener acceso y/o visibilidad a estructura u órganos afectados.

2.1.2.2.3.4.5.4 Diagnostico

Procedimiento preliminar para reconocer e identificar el problema o enfermedad a través de sus signos y síntomas.

2.1.2.2.3.4.5.5 Diente

Órgano de la cavidad bucal que en conjunto confirma una detención. El diente está dividido en corana y raíz, los dientes se agrupan en incisivos, canino, premolares y molares. Está constituido por esmalte, dentina, pulpa y periodonto

2.1.2.2.3.4.5.6 Endodoncia

La endodoncia es la rama de la odontología que se encarga del diagnóstico, tratamiento de las enfermedades pulpares.

2.1.2.2.3.4.5.7 Exodoncia

Procedimiento quirúrgico con el que se realiza la extracción de piezas dentales con pronósticos sin esperanza o con fines ortodónticos.

2.1.2.2.3.4.5.8 Odontólogo

Profesional del área de la salud que con un sentido integral se encarga de cuidar los dientes como un todo del cuerpo humano, manteniéndolos en un estado funcional. Dentista, estomatólogo.

2.1.2.2.3.4.5.9 Odontopediatria

Dentista que se especializa en el cuidado de la salud de los niños, desde bebés hasta adolescentes.

2.1.2.2.3.4.5.10 Radiografía

Examen de Diagnóstico que usa rayos invisibles de energía electromagnética para obtener imágenes de tejidos internos, huesos y órganos en una placa radiográfica.

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

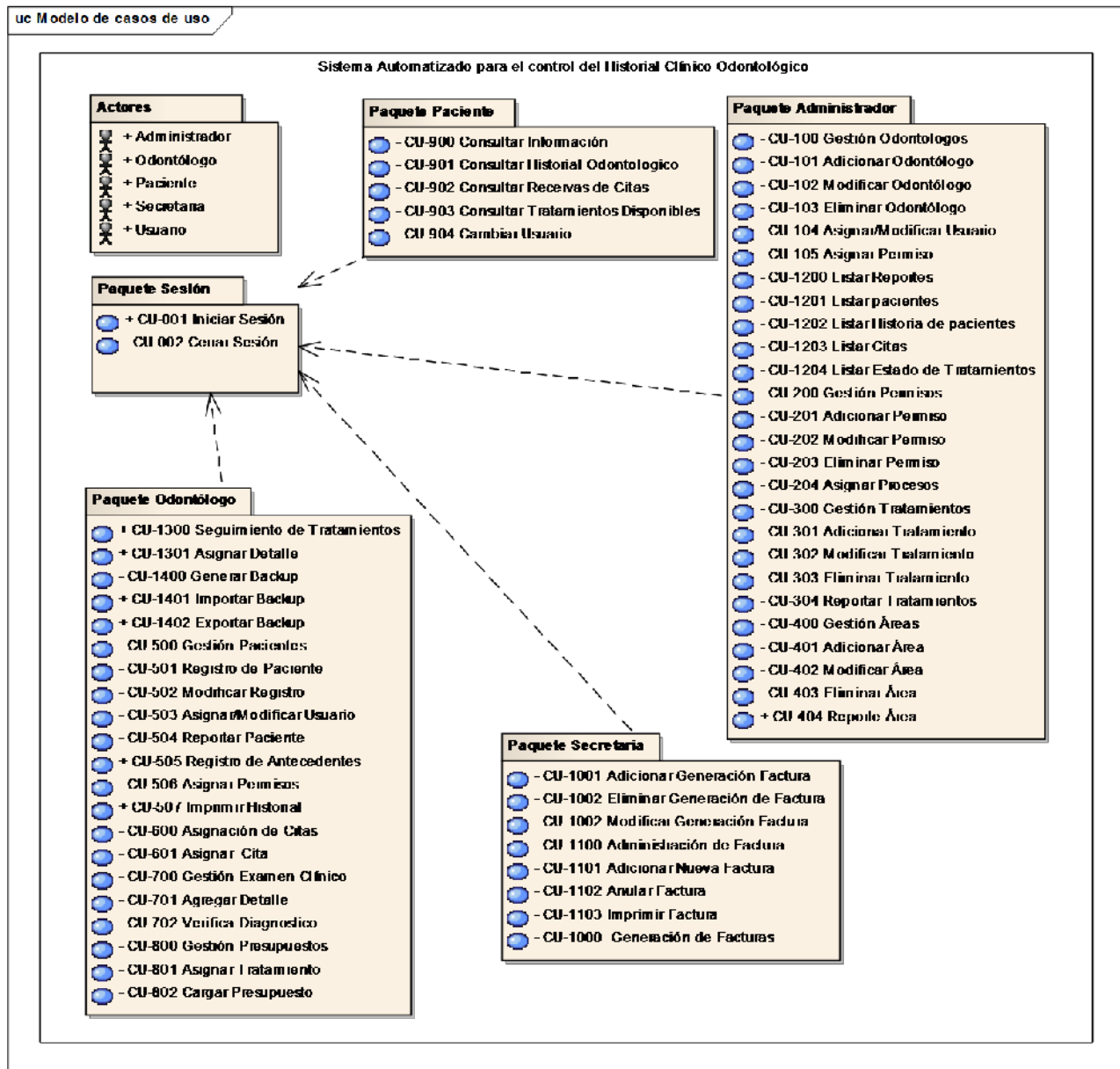


Figura 39 Casos de Usos del Sistema General

2.1.2.2.3.5.4.1.1 Modelo Casos de Uso del Módulo Administrador

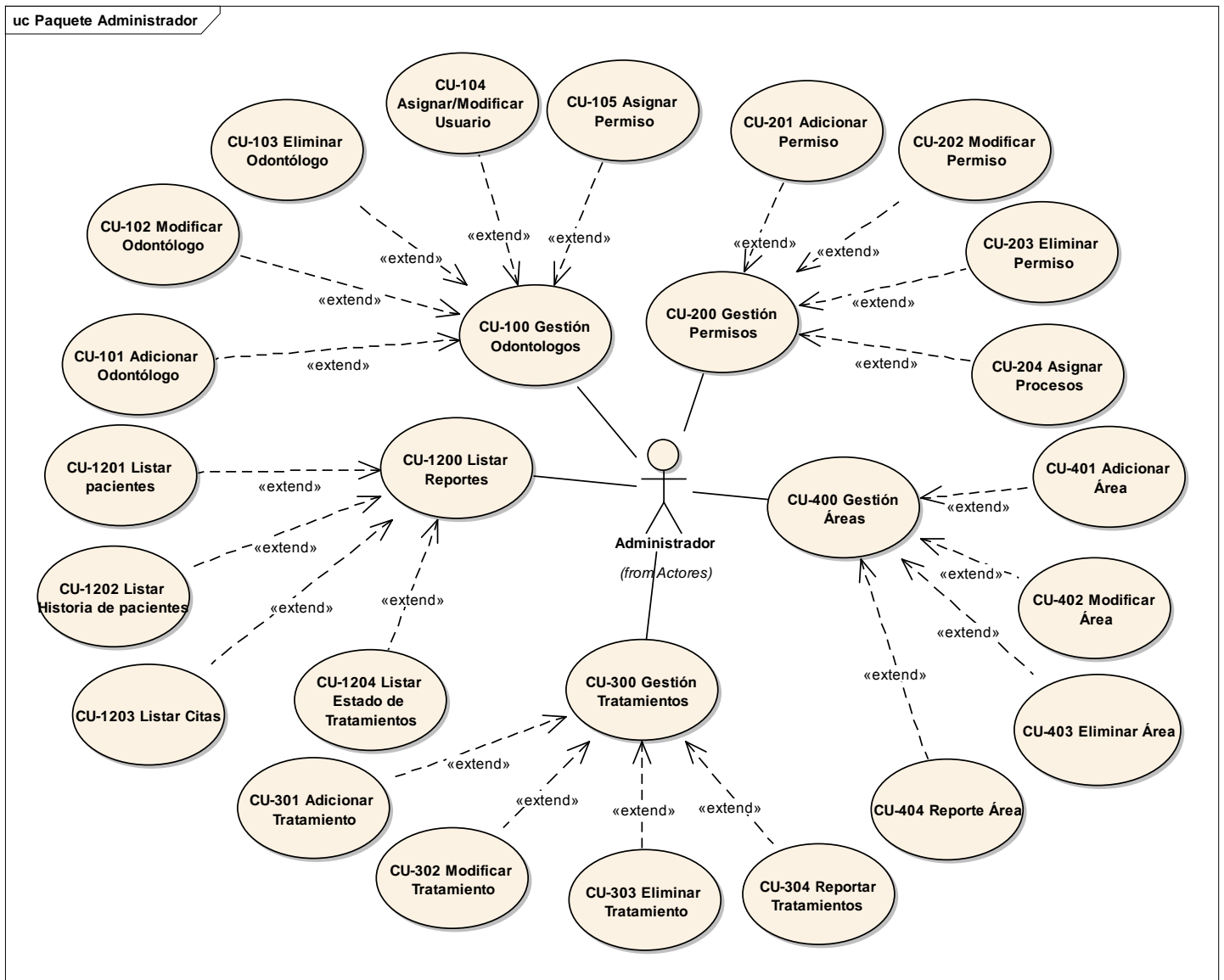


Figura 40 Modelo Casos de Uso del Módulo Administrador

2.1.2.2.3.5.4.1.2 Modelo Casos de Uso del Módulo Odontólogo

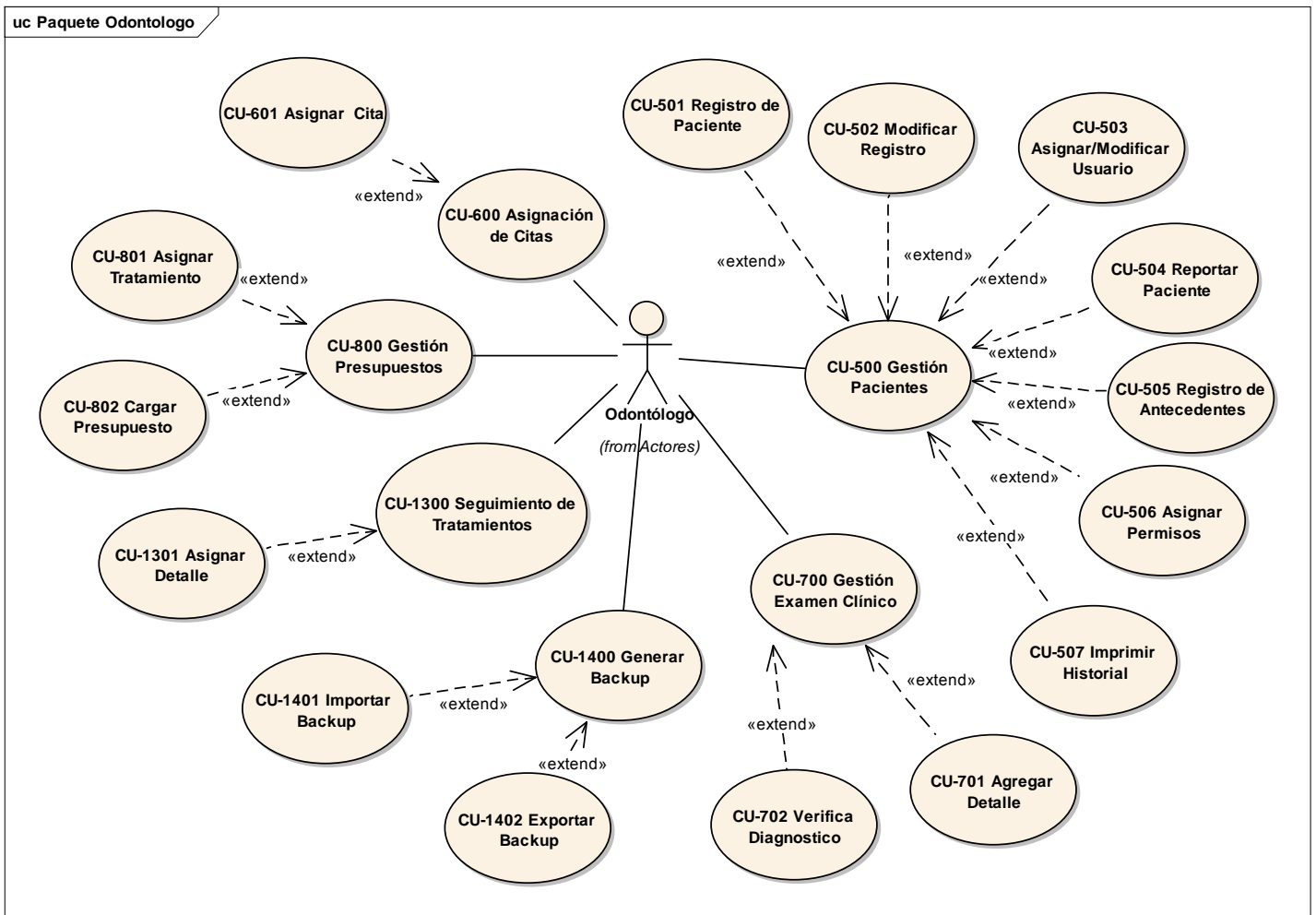


Figura 41 Modelo Casos de Uso del Módulo Odontólogo

2.1.2.2.3.5.4.1.3 Modelo Caso de Uso del Módulo Paciente

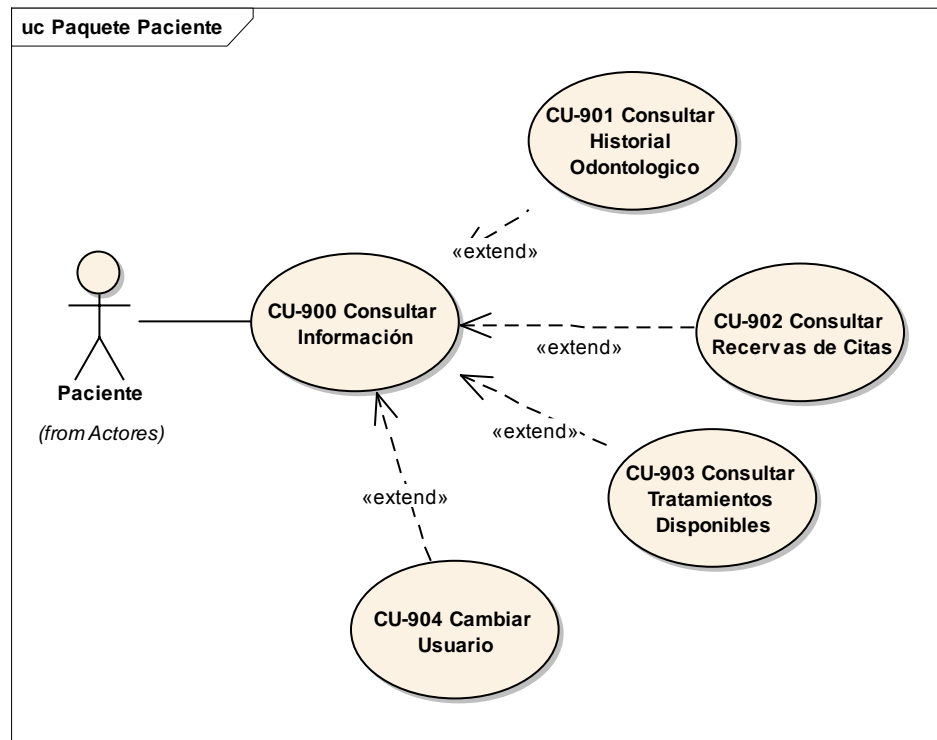


Figura 42 Modelo Caso de Uso del Módulo Paciente

2.1.2.2.3.5.4.1.4 Modelo Caso de Uso del Módulo Secretaria

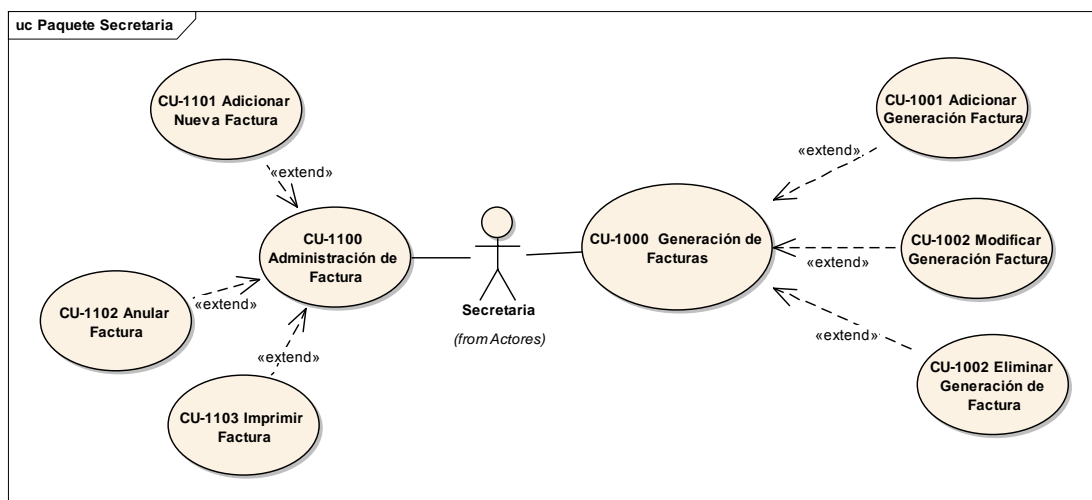


Figura 43 Modelo Caso de Uso del Módulo Secretaria

2.1.2.2.3.5.4.1.5 Modelo Caso de Uso del Módulo Sesión

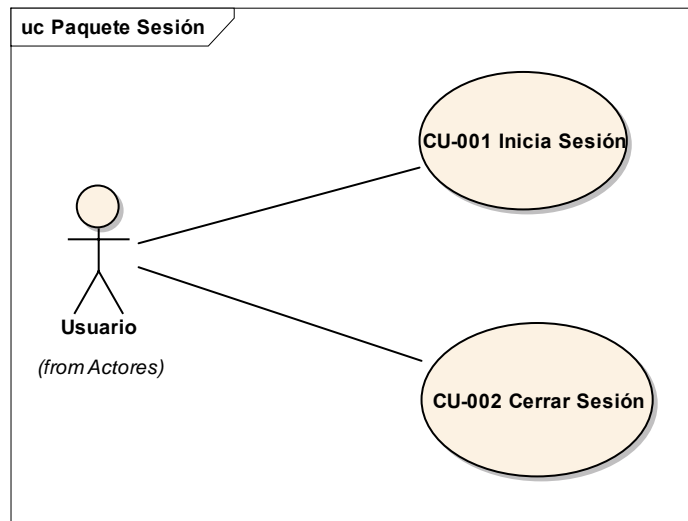


Figura 44 Modelo Caso de uso del Módulo Sesión

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 Odontólogo

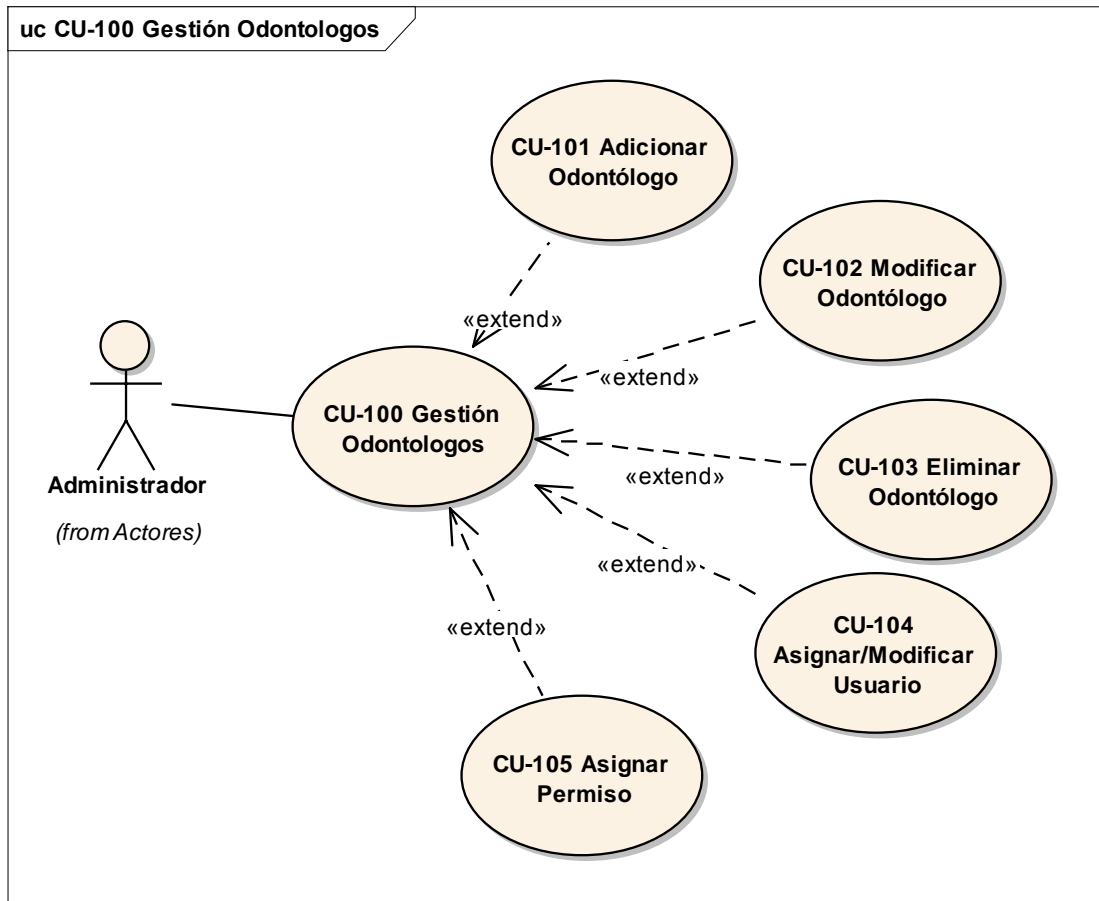


Figura 45 Modelo Caso de Uso Gestión Odontólogo

2.1.2.2.3.5.4.2.2 Modelo Caso de Uso Gestión Permisos

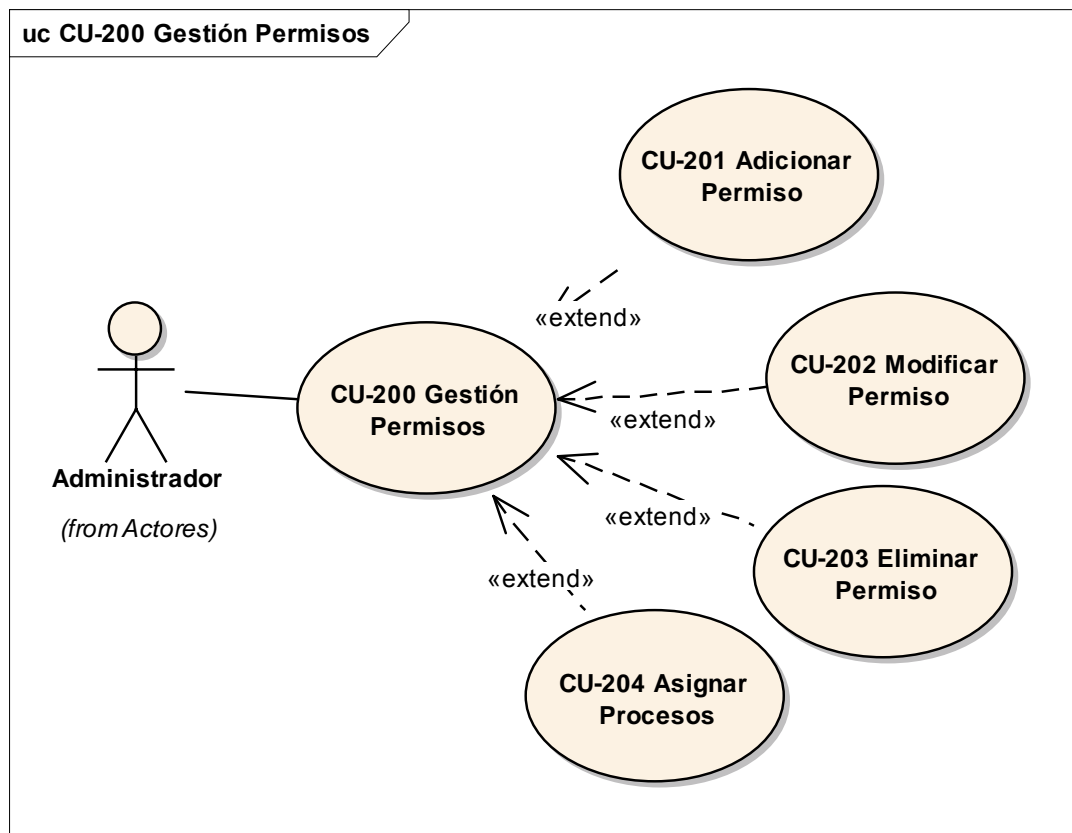


Figura 46 Modelo Caso de Uso Gestión Permisos

2.1.2.2.3.5.4.2.3 Modelo Caso de Uso Gestión Tratamientos

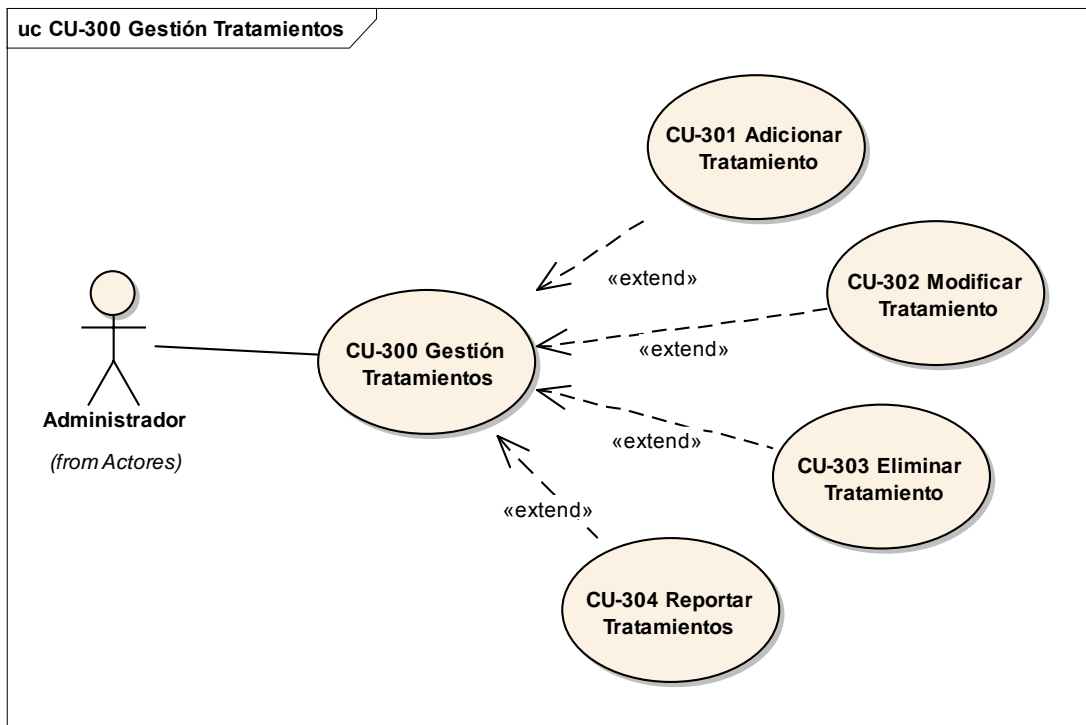


Figura 47 Modelo Caso de Uso Gestión Tratamientos

2.1.2.2.3.5.4.2.4 Modelo Caso de Uso Gestión Áreas

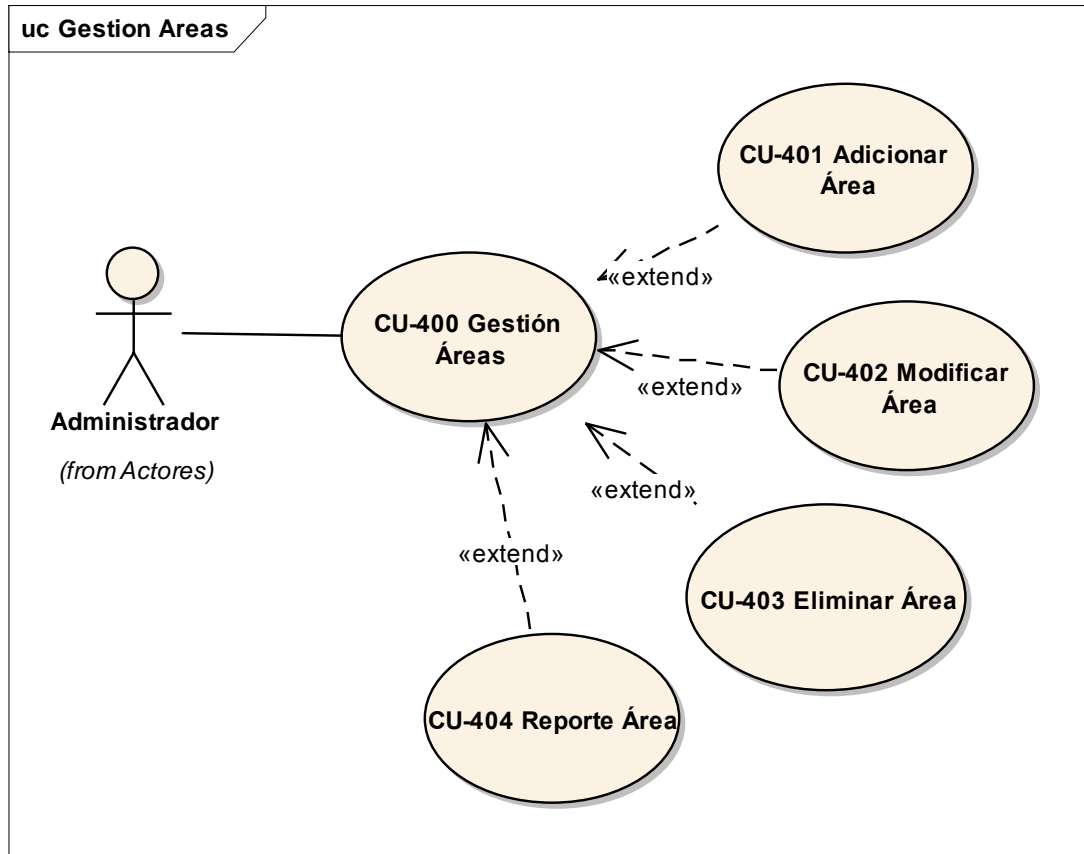


Figura 48 Modelo Caso de Uso Gestión Áreas

2.1.2.2.3.5.4.2.5 Modelo Caso de Uso Gestión Pacientes

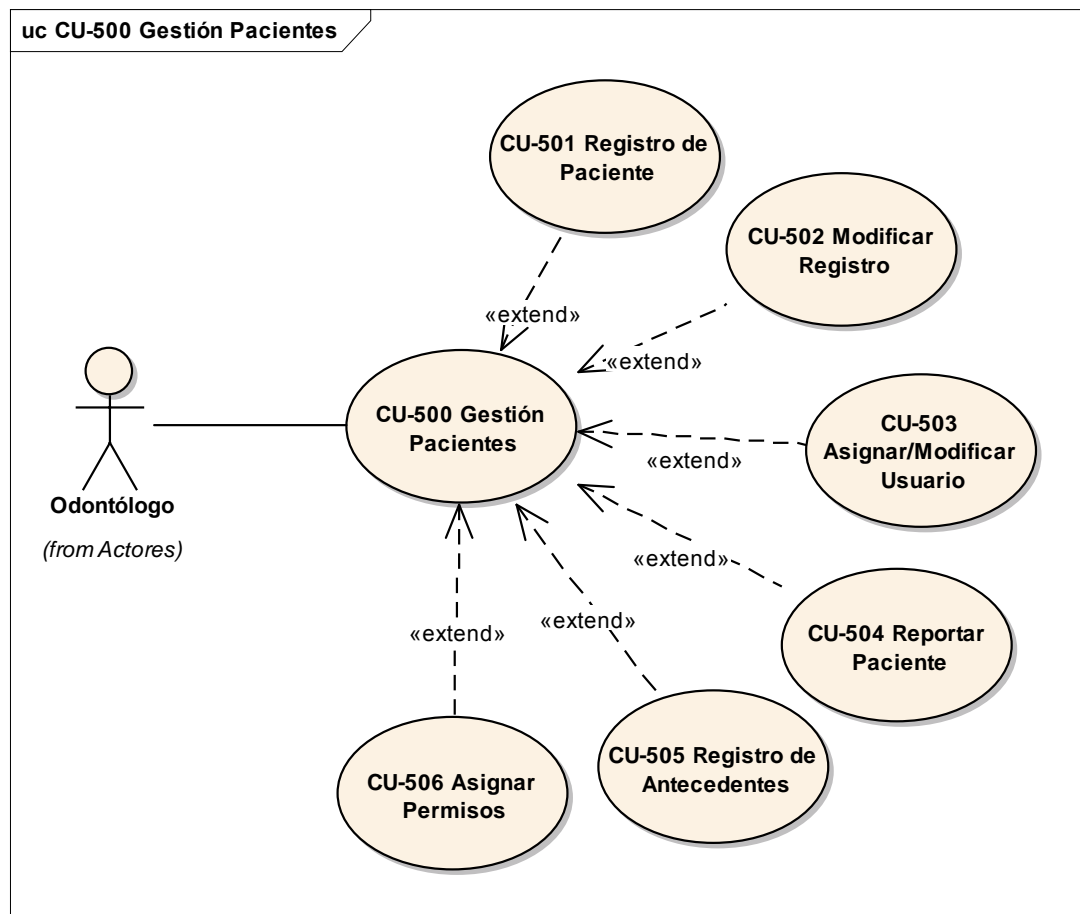


Figura 49 Modelo Caso de Uso Gestión Pacientes

2.1.2.2.3.5.4.2.6 Modelo Caso de Uso Asignación citas

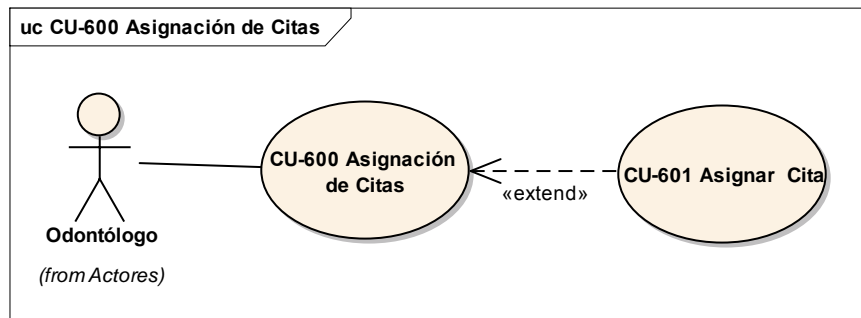


Figura 50 Modelo Caso de Uso Asignación citas

2.1.2.2.3.5.4.2.7 Modelo Caso de Uso Gestión Examen Clínico

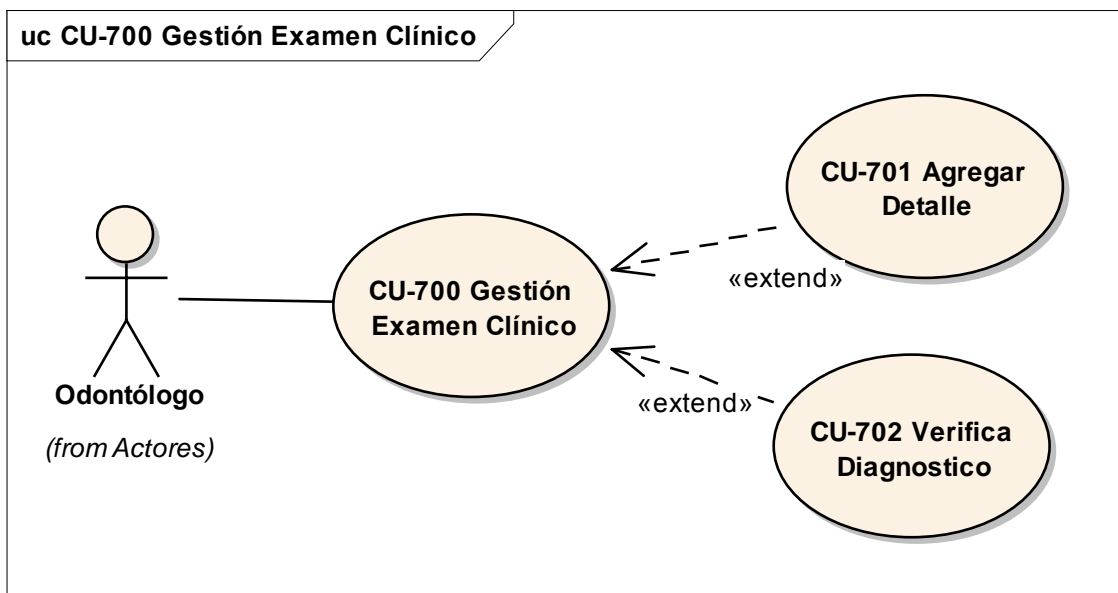


Figura 51 Modelo Caso de Uso Gestión Examen Clínico

2.1.2.2.3.5.4.2.8 Modelo Caso de Uso Gestión Presupuesto

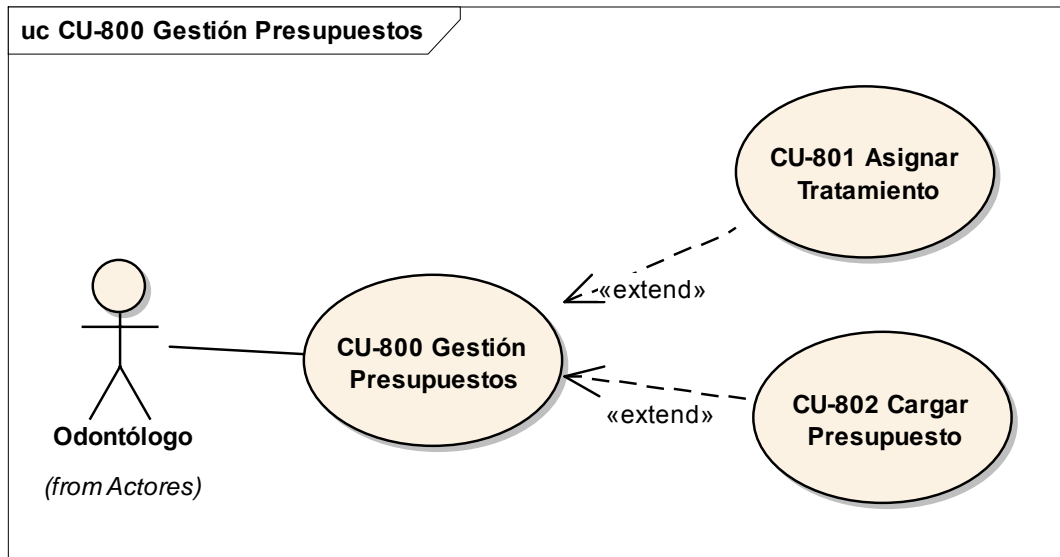


Figura 52 Modelo Caso de Uso Gestión Presupuestos

2.1.2.2.3.5.4.2.9 Modelo Caso de Uso Consultar Información

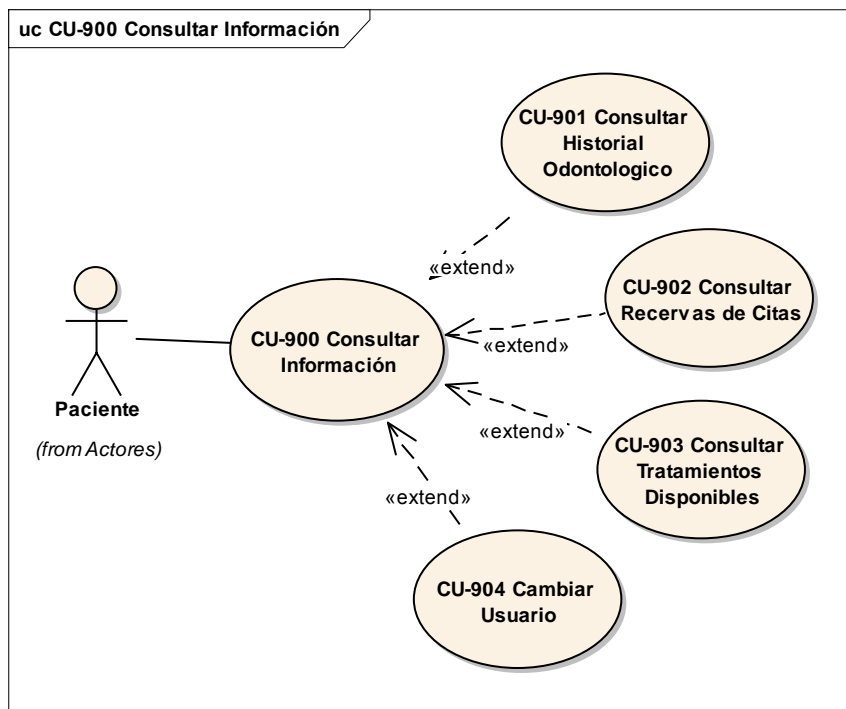


Figura 53 Modelo Caso de Uso Consultar Información

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 del paciente tomando en cuenta la seguridad.
- Contar con la información centralizada en una sola organización.
- Brindar una forma virtual de manejar el historial clínico odontológico.

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 clínica satisface la necesidad de mantener una salud bucal que sea para el bien en lo que respecta la salud y la apariencia (servicios), al brindar al paciente la oportunidad de satisfacer estas necesidades.

Por lo tanto las ventajas de implementar un sistema informático el cual también registre y lleve un control seguro de los historiales clínicos odontológicos 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 historiales.
afecta a	Clínica dental IRACEMA.
El impacto asociado es	Genera retardos y aglomeración de folder con historiales.
Una solución adecuada sería	Diseñar un Sistema informático destinado al Control de los historiales clínicos.

Tabla 12 Sentencia que define el problema

2.1.2.2.3.6.1.4.3 Sentencia que define la posición del producto

para	IRACEMA
quienes	Gerente General, empleados y población en general de Tarija
El nombre del producto	“IRACEMA Virtual”
que	Gestiona oportunamente y efectivamente la información de los historiales clínicos odontológicos.
Nuestro producto	Mejorar y automatizar el control de historiales clínicos odontológicos 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 Clínica Dental IRACEMA

Representante	Giovanna Patricia Iñiguez Aramayo
Descripción	Gerente General de IRACEMA
Tipo	Odontóloga
Responsabilidades	Administrador del Sistema IRACEMA VIRTUAL
Grado de participación	Velar por que se cumplan requisitos y funcionalidades, y que el proyecto sea ejecutado.

Tabla 15 Perfil Stakeholder: Clínica dental IRACEMA

2.1.2.2.3.6.1.5.4 Docente

Representante	Lic. Deysi Arancibia
----------------------	----------------------

Descripción	Ente regulador universitario.
Tipo	Guía de desarrollo.
Responsabilidades	Realiza un control paso a paso del desarrollo del proyecto. Aprueba los distintos puntos tratados en el proyecto. Analiza los distintos documentos presentados acerca del producto.
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 Odontólogo

Representante	Odontólogo
Descripción	Usuario con ciertos atributos- permisos de ejecución de la aplicación.
Tipo	Usuario principal.
Responsabilidades	Es el encargado de registrar los pacientes y sus historias clínicas odontológicas.
Grado de participación	Usar el Sistema, buscar debilidades del mismo.

Tabla 18 Perfil Stakeholder: Odontólogo

2.1.2.2.3.6.1.6.3 Paciente

Representante	Paciente
Descripción	Usuario que podrá realizar consultas de su información personal con la necesidad de identificarse
Tipo	Usuario regular
Responsabilidades	Es el usuario que interactúa directamente con la información personal
Grado de participación	Usar el Sistema, buscar debilidades del mismo.

Tabla 19 Perfil Stakeholder: Paciente

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 “IRACEMA Virtual”.

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:	<ol style="list-style-type: none"> 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 la Pantalla Principal de Administración. 	Flujo Alternativo: <ol style="list-style-type: none"> 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.
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

Odontólogos

Caso de uso:	Gestión Odontólogos
---------------------	---------------------

Descripción: Permite Listar, Adicionar, Modificar, Eliminar al Odontólogo, asignar usuario y permisos.	
Actores: Administrador.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El sistema muestra Listado de los Odontólogos registrados en el sistema. 2. El actor realiza la Búsqueda de los Odontólogos en el Listado. 3. El sistema muestra los Odontólogos 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 Odontólogos

2.1.2.2.3.11.2.2 Especificación CU-101 Adicionar Odontólogo

Caso de uso:	Adicionar Odontólogo
Descripción: Permite registrar un nuevo Odontólogo 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 botón Adicionar nuevo Odontólogo. 2. El sistema le muestra formulario. 3. El actor introduce datos del odontólogo 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 el odontólogo XXXX” 7. El actor presiona botón Cancelar y vuelve a la pantalla Gestión Odontólogos 	<ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Odontólogos. 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 Odontólogo

2.1.2.2.3.11.2.3 Especificación CU-102 Modificar Odontólogo

Caso de uso:	Modificar Odontólogo
Descripción: Permite modificar datos de un Odontólogo en el sistema.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión, Estar registrado en el sistema el Odontólogo	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Modificación. 2. El sistema le muestra formulario con datos del Odontólogo. 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Odontólogos. 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.

<ol style="list-style-type: none"> 3. El actor realiza los cambios de datos del odontólogo en el formulario. 4. Selecciona botón Modificar. 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 Odontólogos. 	
Pos condiciones: Se Modificó Satisfactoriamente	

Tabla 23 Especificación CU-102 Modificar Odontólogo

2.1.2.2.3.11.2.4 Especificación CU-103 Eliminar Odontólogo

Caso de uso:	Eliminar Odontólogo	
Descripción: Permite eliminar un Odontólogo registrada 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 Baja/Alta. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación lógica (Baja) o la alta si esta eliminado lógicamente el odontólogo 4. El sistema regresa a la pantalla de Gestión Odontólogo. 	Flujo Alternativo: <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 Odontólogo

2.1.2.2.3.11.2.5 Especificación CU-200 Gestión

Permisos

Caso de uso:	Gestión Permisos	
Descripción:	Permite Listar, Adicionar, Modificar y Eliminar los permisos.	
Actores:	Administrador.	
Precondiciones:	Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal:	Flujo Alternativo:	
<ol style="list-style-type: none"> 1. El sistema muestra Listado de los Permisos registrados en el sistema. 2. El actor realiza la Búsqueda de los Permisos en el Listado. 3. El sistema muestra los Permiso que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona la opción que requiere. 	<ol style="list-style-type: none"> 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación. 	
Pos condiciones:	Ninguno	

Tabla 25 Especificación CU-200 Gestión Permisos

2.1.2.2.3.11.2.6 Especificación CU-201 Adicionar

Permiso

Caso de uso:	Adicionar Permiso
Descripción:	Permite registrar un nuevo permiso en el sistema.

Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión	
Flujo Normal: 1. El actor presiona botón Adicionar nuevo Permiso. 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: “Se adiciono exitosamente” 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 Permisos. 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 Permiso

2.1.2.2.3.11.2.7 Especificación CU-202 Modificar Permiso

Caso de uso:	Modificar Permiso
Descripción: Permite modificar datos de un permiso en el sistema.	
Actores: Administrador.	
Precondiciones: Actor debe haber iniciado sesión, Estar registrado en el sistema el permiso	

<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Modificación. 2. El sistema le muestra formulario con datos del permiso. 3. El actor realiza los cambios de datos del permiso en el formulario. 4. Selecciona botón Modificar. 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 Permisos. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Permisos. 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.
<p>Pos condiciones: Se Modificó Satisfactoriamente</p>	

Tabla 27 Especificación CU-202 Modificar Permiso

2.1.2.2.3.11.2.8 Especificación CU-203 Eliminar Permiso

Caso de uso:	Eliminar Permiso
Descripción:	
Permite eliminar un Permiso registrada en el sistema.	
Actores:	
Administrador.	
Precondiciones:	
Actor debe haber iniciado sesión.	
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Baja/Alta. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación lógica (Baja) o la alta 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 1. Por no el sistema no realizará ninguna acción.

si esta eliminado lógicamente el permiso	
4. El sistema regresa a la pantalla de Gestión Permisos.	
Pos condiciones: Se eliminó Satisfactoriamente	

Tabla 28 Especificación CU-203 Eliminar Permiso

2.1.2.2.3.11.2.9 Especificación CU-300 Gestión Tratamientos

Caso de uso:	Gestión Tratamientos	
Descripción: Permite Listar, Adicionar, Modificar y Eliminar Tratamientos.		
Actores: Administrador.		
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.		
Flujo Normal:		Flujo Alternativo:
<ol style="list-style-type: none"> 1. El sistema muestra Listado de los Tratamientos registrados en el sistema. 2. El actor realiza la Búsqueda de los Tratamientos 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. 		<ol style="list-style-type: none"> 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno		

Tabla 29 Especificación CU-300 Gestión Tratamientos

2.1.2.2.3.11.2.10 Especificación CU-301 Adicionar

Tratamiento

Caso de uso:	Adicionar Tratamiento	
Descripción: Permite registrar un nuevo tratamiento 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 botón Adicionar nuevo Tratamiento. 2. El sistema le muestra formulario. 3. El actor introduce datos del tratamiento 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 Tratamientos 		<ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Tratamientos. 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 Tratamiento

2.1.2.2.3.11.2.11 Especificación CU-302 Modificar

Tratamiento

Caso de uso:	Modificar Tratamiento	
Descripción: Permite modificar datos de un tratamiento 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 Modificación. 2. El sistema le muestra formulario con datos del tratamiento. 3. El actor realiza los cambios de datos del tratamiento en el formulario. 4. Selecciona botón Modificar. 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 Tratamientos.	Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Tratamientos. 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 31 Especificación CU-302 Modificar Tratamiento

2.1.2.2.3.11.2.12 Especificación CU-303 Eliminar Tratamiento

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

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

Tabla 32 Especificación CU-303 Eliminar Tratamiento

2.1.2.2.3.11.2.13 Especificación CU-400 Gestión Áreas

Caso de uso:	Gestión Áreas
Descripción:	
Permite Listar, Adicionar, Modificar, Eliminar y realizar un reporte de las Áreas.	
Actores:	
Administrador.	
Precondiciones:	
Actor no tiene privilegios para realizar ninguna acción.	
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El sistema muestra Listado de las Áreas registrados en el sistema. 2. El actor realiza la Búsqueda de los Áreas en el Listado. 3. El sistema muestra las Áreas que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona la opción que requiere. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Pos condiciones: Ninguno	

Tabla 33 Especificación CU-400 Gestión Áreas

2.1.2.2.3.11.2.14 Especificación CU-401 Adicionar Área

Caso de uso:	Adicionar Área	
Descripción:	Permite registrar un nuevo Área 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 Área. 2. El sistema le muestra formulario. 3. El actor introduce datos del Área 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 Áreas 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Áreas. 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 Área

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

Precondiciones: Actor debe haber iniciado sesión, Estar registrado en el sistema el Área	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona en el menú la opción Modificación. 2. El sistema le muestra formulario con datos del Área. 3. El actor realiza los cambios de datos del Área en el formulario. 4. Selecciona botón Modificar. 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 Áreas. 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Áreas. 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 Área

2.1.2.2.3.11.2.16 Especificación CU-403 Eliminar Área

Caso de uso:	Eliminar Área
Descripción: Permite eliminar un Área registrada 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 Baja/Alta. 2. El sistema muestra el mensaje de confirmación. 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Por no el sistema no realizará ninguna acción.

<p>3. Por Si, El sistema realiza la eliminación lógica (Baja) o la alta si esta eliminado lógicamente el Área</p> <p>4. El sistema regresa a la pantalla de Gestión Áreas.</p>	
<p>Pos condiciones: Se eliminó Satisfactoriamente</p>	

Tabla 36 Especificación CU-403 Eliminar Área

2.1.2.2.3.11.3 Especificación de Caso de Uso del Odontólogo

2.1.2.2.3.11.3.1 Especificación CU-500 Gestión Pacientes

Caso de uso:	Gestión Pacientes	
<p>Descripción: Permite Listar, Registrar, Modificar, Asignar Antecedentes, Asignar Usuario y Aginar Permisos al Pacientes.</p>		
<p>Actores: Odontólogo.</p>		
<p>Precondiciones: Actor no tiene privilegios para realizar ninguna acción.</p>		
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El sistema muestra Listado de los Pacientes registrados en el sistema. 2. El actor realiza la Búsqueda de los Pacientes en el Listado. 3. El sistema muestra los Listados que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona la opción que requiere. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación. 	
<p>Pos condiciones: Ninguno</p>		

Tabla 37 Especificación CU-500 Gestión Pacientes

2.1.2.2.3.11.3.2 Especificación CU-501 Registro de Paciente

Caso de uso:	Registro de paciente	
Descripción:	Permite registrar un paciente con sus antecedentes personales y familiares, periodoncia, enfermedades actuales, examen bucal, aplicación de flúor y nomenclatura en el sistema.	
Actores:	Odontólogo.	
Precondiciones:	Actor debe haber iniciado sesión	
Flujo Normal:	Flujo Alternativo:	
<ol style="list-style-type: none"> 1. El actor presiona botón Adicionar nuevo Paciente. 2. El sistema le muestra formulario. 3. El actor introduce datos del Paciente y sus datos clínicos 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 el paciente” 7. El actor presiona botón Cancelar y vuelve a la pantalla Gestión Pacientes 	<ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Pacientes. 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 Registro de Pacientes

2.1.2.2.3.11.3.3 Especificación CU-502 Modificar

Registro

Caso de uso:	Modificar Registro
Descripción: Permite modificar datos personales y clínicos del paciente en el sistema.	
Actores: Odontólogo.	
Precondiciones: Actor debe haber iniciado sesión, Estar registrado en el sistema el paciente	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor busca el paciente. 2. El sistema le muestra formulario con datos del paciente. 3. El actor realiza los cambios de datos del paciente en el formulario. 4. Selecciona botón Modificar. 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 Pacientes. 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Pacientes. 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 Registro

2.1.2.2.3.11.3.4 Especificación CU-504 Reportar

Paciente

Caso de uso:	Reportar Paciente
Descripción: Permite realizar un reporte de los datos personales y clínicos de un paciente en el sistema.	

Actores: Odontólogo.	
Precondiciones: Actor debe haber iniciado sesión, Estar registrado en el sistema el paciente	
Flujo Normal: 1. El actor presiona en el menú la opción Reporte. 2. El sistema le muestra un pdf con los datos del paciente en otra pestaña.	Flujo Alternativo: 1. No selecciona nada y se mantiene en la pantalla Gestión pacientes.
Pos condiciones: Se Modificó Satisfactoriamente	

Tabla 40 Especificación CU-504 Reportar Paciente

2.1.2.2.3.11.3.5 Especificación CU-505 Detalle

Diagnostico

Caso de uso:	Detalle Diagnostico
Descripción: Permite asignar los diagnósticos encontrados durante el examen físico al paciente.	
Actores: Odontólogo.	
Precondiciones: Actor debe haber iniciado sesión. Estar registrado en el sistema el paciente.	
Flujo Normal: 1. El actor presiona botón Detalle Diagnostico. 2. El sistema le muestra formulario. 3. El actor introduce datos del detalle en el formulario. 4. Selecciona botón Guardar.	Flujo Alternativo: 1. Selecciona botón Cancelar, se regresa a la pantalla Gestión Pacientes. 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.

<p>5. Se valida datos del formulario que son obligatorios estén introducidos.</p> <p>6. El sistema guarda datos del formulario y muestra mensaje: “Se adicióno exitosamente”.</p> <p>7. El actor presiona botón Cancelar y vuelve a la pantalla Gestión Pacientes.</p>	
<p>Pos condiciones: Ninguno</p>	

Tabla 41 Especificación CU-505 Detalle Diagnostico

2.1.2.2.3.11.3.6 Especificación CU-600 Asignación de Citas

<p>Caso de uso:</p>	<p>Asignación de Citas</p>	
<p>Descripción:</p> <p>Permite mostrar las citas del Odontólogo seleccionado y permite dar la opción de Asignar un Cita.</p>		
<p>Actores:</p> <p>Odontólogo.</p>		
<p>Precondiciones:</p> <p>Actor no tiene privilegios para realizar ninguna acción.</p>		
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El sistema muestra Listado de las citas por doctor registrados en el sistema. 2. El actor realiza la Búsqueda de los citas en el Listado. 3. El sistema muestra las citas que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona la opción que requiere. 	<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 1. El actor decide no confirmar selección y el sistema no realiza ninguna operación. 	

Pos condiciones: Ninguno

Tabla 42 Especificación CU-600 Asignación de Citas

2.1.2.2.3.11.3.7 Especificación CU-601 Asignar Cita

Caso de uso:	Asignar Cita
Descripción: Permite Asignar una cita en el sistema.	
Actores: Odontólogo.	
Precondiciones: Actor debe haber iniciado sesión.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor presiona botón Asignar Cita. 2. El sistema le muestra formulario. 3. El actor introduce datos del cita a ser asignada en el formulario. 4. Selecciona botón Guardar Cita. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario y muestra mensaje: "Se asignó exitosamente la cita" 7. El actor presiona botón Retornar y vuelve a la pantalla Asignación de Cita 	Flujo Alternativo: <ol style="list-style-type: none"> 1. Selecciona botón Retornar, se regresa a la pantalla Asignación de Cita. 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 43 Especificación CU-601 Asignación Cita

2.1.2.2.3.11.3.8 Especificación CU-700 Gestión Examen Clínico

Caso de uso:	Gestión Examen Clínico
---------------------	------------------------

Descripción: Permite agregar detalles del odontograma con sus respectivos diagnóstico.	
Actores: Odontólogo.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El sistema muestra Listado de los Paciente registrados en el sistema. 2. El actor realiza la Búsqueda de los Pacientes en el Listado. 3. El sistema muestra los Pacientes 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 44 Especificación CU-700 Gestión Examen Clínico

2.1.2.2.3.11.3.9 Especificación CU-701 Verificar Diagnostico

Caso de uso:	Seguimiento del Tratamiento
Descripción: Permite verificar si existe un diagnostico ya registrado o registrarlo si es nuevo.	
Actores: Odontólogo.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal:	Flujo Alternativo:

<ol style="list-style-type: none"> 1. El sistema verifica si existe un diagnostico relacionado con el área seleccionada. 2. Si existe el diagnostico el sistema procede a realizar el llenado de los datos del detalle. 3. Si no existe se mantiene los datos del detalle. 	
Pos condiciones: Ninguno	

Tabla 45 Especificación CU-701 Seguimiento del Tratamiento

2.1.2.2.3.11.3.10 Especificación CU-702 Agregar Detalle

Caso de uso:	Agregar Detalle	
Descripción: Permite agregar los datos del detalle al presupuesto del paciente.		
Actores: Odontólogo.		
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.		
Flujo Normal: <ol style="list-style-type: none"> 1. Seleccionar Agregar Detalle 2. Llenar datos del detalle 3. Seleccionar cavidad dentaria para agregar el detalle. 		Flujo Alternativo: <ol style="list-style-type: none"> 1. Selecciona otro paciente y muestra un odontograma nuevo para el paciente.
Pos condiciones: Ninguno		

Tabla 46 Especificación CU-702 Mantenimiento del Diagnostico

2.1.2.2.3.11.3.11 Especificación CU-800 Gestión Presupuesto

Caso de uso:	Gestión Presupuestos
Descripción:	

Permite asignar los tratamientos a los diagnósticos asignados en el examen clínico	
Actores: Odontólogo.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El sistema muestra Listado de los pacientes registrados en el sistema. 2. El actor realiza la Búsqueda de los pacientes en el Listado. 3. El sistema muestra los pacientes 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 47 Especificación CU-800 Gestión Presupuesto

2.1.2.2.3.11.3.12 Especificación CU-801 Asignar Tratamiento

Caso de uso:	Asignar Tratamiento
Descripción: Permite Asignar Tratamiento al Diagnóstico de las cavidades dentarias examinadas en el examen clínico.	
Actores: Odontólogo.	
Precondiciones: Actor debe haber iniciado sesión	
Flujo Normal: 1. El actor selecciona una cavidad	Flujo Alternativo: 1. El actor selecciona cancelar

<p>dentaria.</p> <p>2. El sistema muestra los detalles asignados en el examen clínico con un formulario para seleccionar el tratamiento a ser realizado.</p> <p>3. El actor selecciona Registrar.</p> <p>4. El sistema procede a registrar la asignación.</p> <p>5.</p>	<p>2. El sistema vuelve al odontograma con los diagnósticos.</p>
<p>Pos condiciones: Ninguno</p>	

Tabla 48 Especificación CU-801 Asignación Presupuesto

2.1.2.2.3.11.3.13 Especificación CU-802 Cargar Presupuesto

Caso de uso:	Cargar Presupuesto
Descripción:	
Permite calcular el monto total del presupuesto.	
Actores:	
Odontólogo.	
Precondiciones:	
El sistema debió haber asignado un tratamiento a una cavidad dentaria.	
Flujo Normal:	Flujo Alternativo:
<p>1. El sistema asigna satisfactoriamente un tratamiento.</p> <p>2. El sistema calcular el monto total de los diagnósticos seleccionados .</p>	<p>1.</p>
Pos condiciones: Se Calculó el monto total del Presupuesto Satisfactoriamente	

Tabla 49 Especificación CU-802 Ampliación de Presupuesto

2.1.2.2.3.11.4 Especificación de Caso de Uso del Paciente

2.1.2.2.3.11.4.1 Especificación CU-900 Consultar

Información

Caso de uso:	Consultar Información
Descripción: Permite realizar consulta y reportes del historial odontológico, reservas de citas, tratamientos disponibles y cambiar usuario.	
Actores: Odontólogo.	
Precondiciones: Actor no tiene privilegios para realizar ninguna acción.	
Flujo Normal: 1. El sistema muestra Listado de las Consultas que puede realizar el paciente registrado en el sistema. 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 50 Especificación CU-900 Consultar Información

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

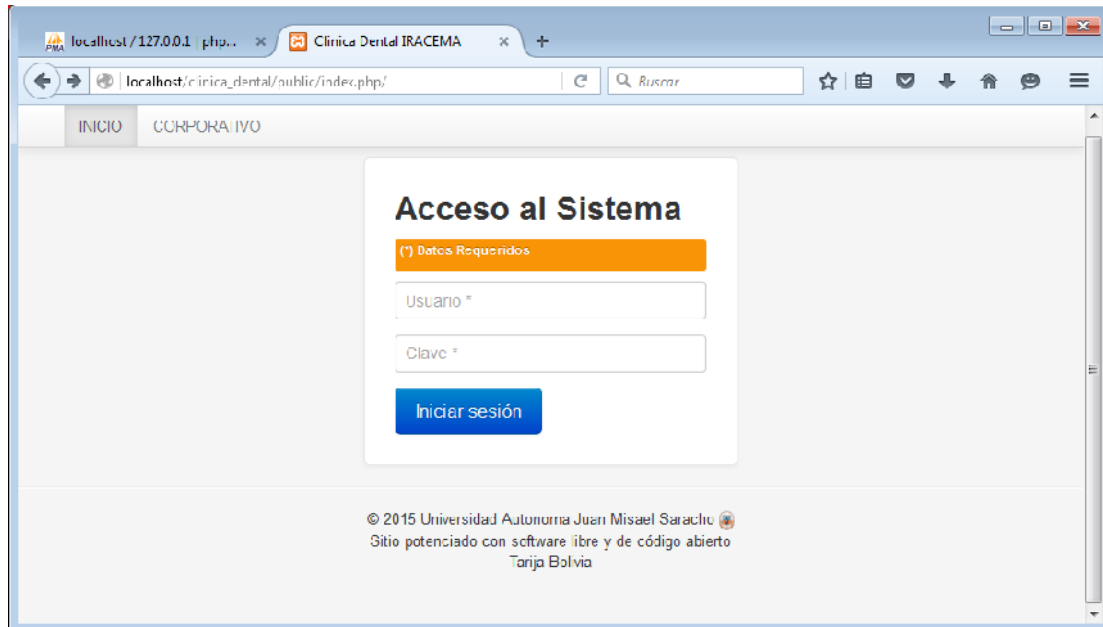


Figura 54 Pantalla Iniciar Sesión

2.1.2.2.3.13.4.1.1 Pantalla Principal Administrador

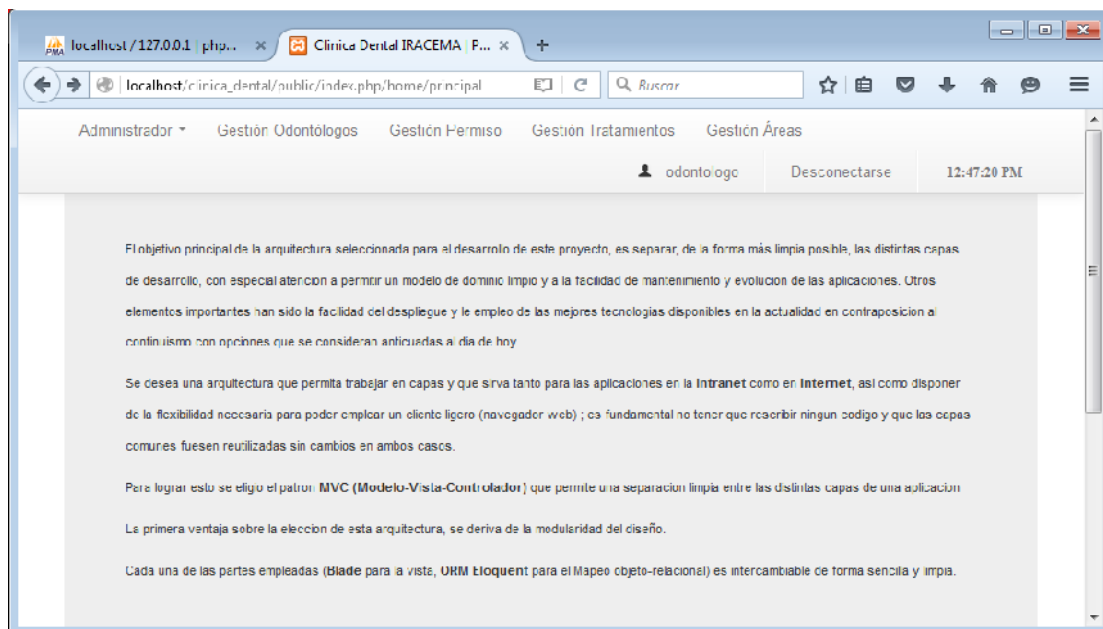


Figura 55 Pantalla Principal Administrador

2.1.2.2.3.13.4.1.2 Pantalla Gestión Odontólogos

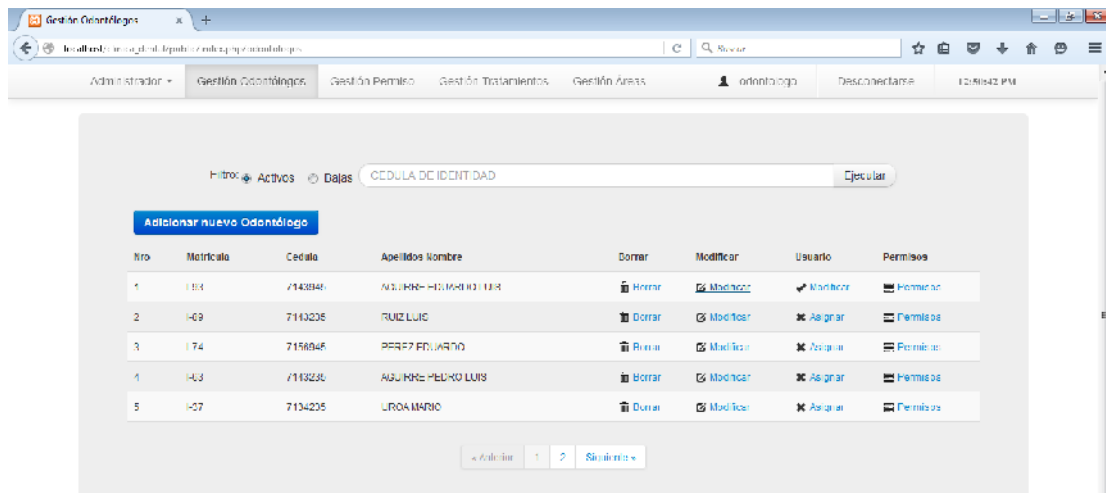


Figura 56 Pantalla Gestión Odontólogos

2.1.2.2.3.13.4.1.3 Pantalla Adicionar Odontólogo

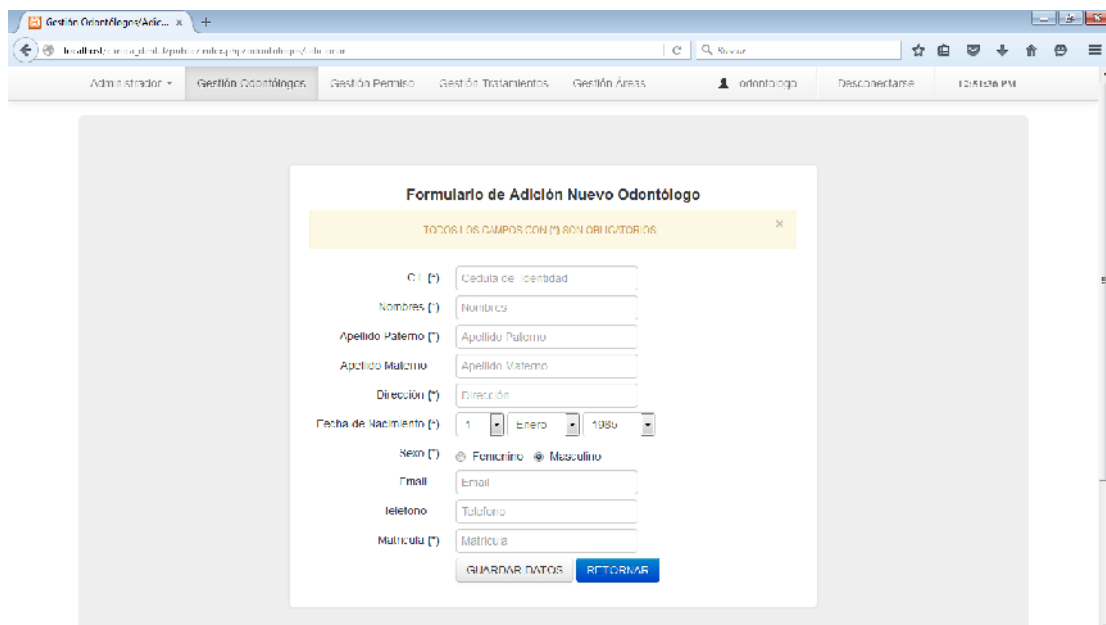


Figura 57 Pantalla Adicionar Odontólogo

2.1.2.2.3.13.4.1.4 Pantalla Modificar Odontólogo

Formulario de Modificar Odontólogo

⚠️ TODOS LOS CAMPOS CON (*) SON OBLIGATORIOS

C.I. (*)

Nombres (*)

Apellido Paterno (*)

Apellido Materno

Dirección (*)

Fecha de Nacimiento (*)

Sexo (*) Femenino Masculino

Email

Teléfono

Matrícula (*)

Figura 58 Pantalla Modificar Odontólogo

2.1.2.2.3.13.4.1.5 Pantalla Eliminar Odontólogo

Historial: Activos Datos: OEDULA DE IDENTIDAD

ID	Matrícula	Celular	Apellidos	Eliminar	Modificar	Usuario	Permisos
1	1-04	7143849	ACURRE	<input type="button" value="Eliminar"/>	<input type="button" value="Modificar"/>	<input checked="" type="checkbox"/> Mantener	<input type="button" value="Permisos"/>
2	-03	7143226	QUELLIS	<input type="button" value="Eliminar"/>	<input type="button" value="Modificar"/>	<input checked="" type="checkbox"/> Asignar	<input type="button" value="Permisos"/>
3	74	7198848	PEREZ EDUARDO	<input type="button" value="Eliminar"/>	<input type="button" value="Modificar"/>	<input checked="" type="checkbox"/> Asignar	<input type="button" value="Permisos"/>
4	-03	7143230	ACURRE PEDRO LUIS	<input type="button" value="Eliminar"/>	<input type="button" value="Modificar"/>	<input checked="" type="checkbox"/> Asignar	<input type="button" value="Permisos"/>
5	-07	7104226	URDARRIN	<input type="button" value="Eliminar"/>	<input type="button" value="Modificar"/>	<input checked="" type="checkbox"/> Asignar	<input type="button" value="Permisos"/>

© 2015 Universidad Autónoma Juan Misael Galacho

Figura 59 Pantalla Eliminar Odontólogo

2.1.2.2.3.13.4.1.6 Pantalla Asignar/Modificar Usuario

Formulario de Asignación de Usuario

TODOS LOS CAMPOS SON OBLIGATORIOS

Usuario (*) R/148256

Contraseña (*) Contraseña

Repite Contraseña (*) Repite Contraseña

GUARDAR DATOS RETORNAR

© 2015 Universidad Autónoma Juan Misael Saracho
 Sitio potenciado con software libre y de código abierto
 Tarija-Bolivia

Figura 60 Pantalla Asignar/Modificar Usuario

2.1.2.2.3.13.4.1.7 Pantalla Permisos

Asignación de Permisos a Usuario

USUARIO: admin@ucom

Permisos Asignados

Admin. strador

Odontólogo

Permisos no Asignados

Paciente

RETORNAR

© 2015 Universidad Autónoma Juan Misael Saracho
 Sitio potenciado con software libre y de código abierto
 Tarija-Bolivia

Figura 61 Pantalla Permisos

2.1.2.2.3.13.4.1.8 Pantalla Gestión Permisos

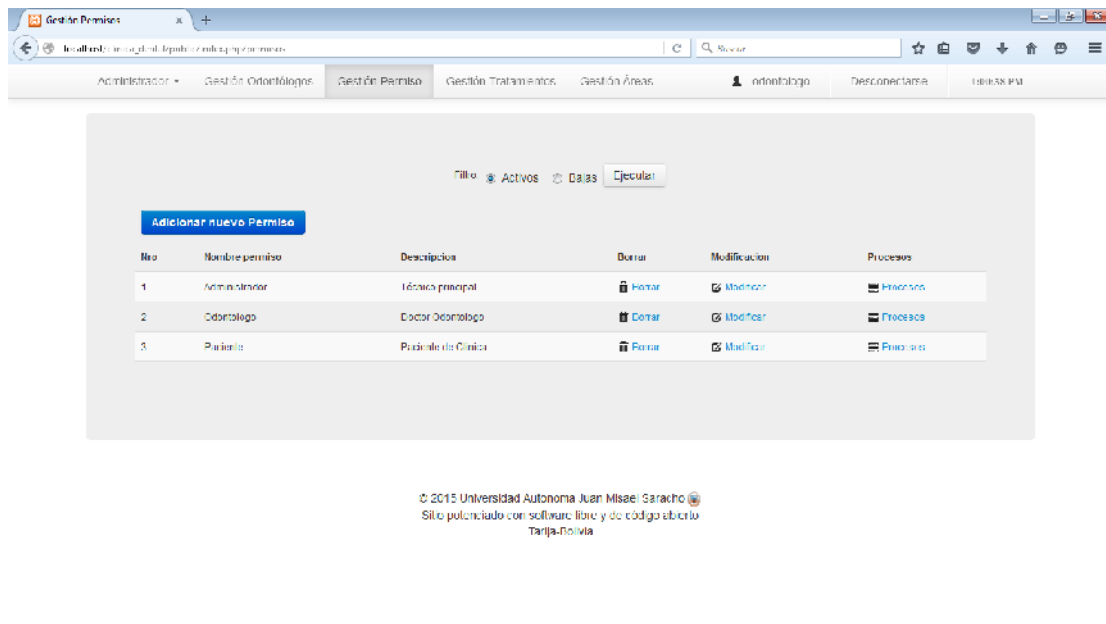


Figura 62 Pantalla Gestión Permisos

2.1.2.2.3.13.4.1.9 Pantalla Gestión Tratamientos

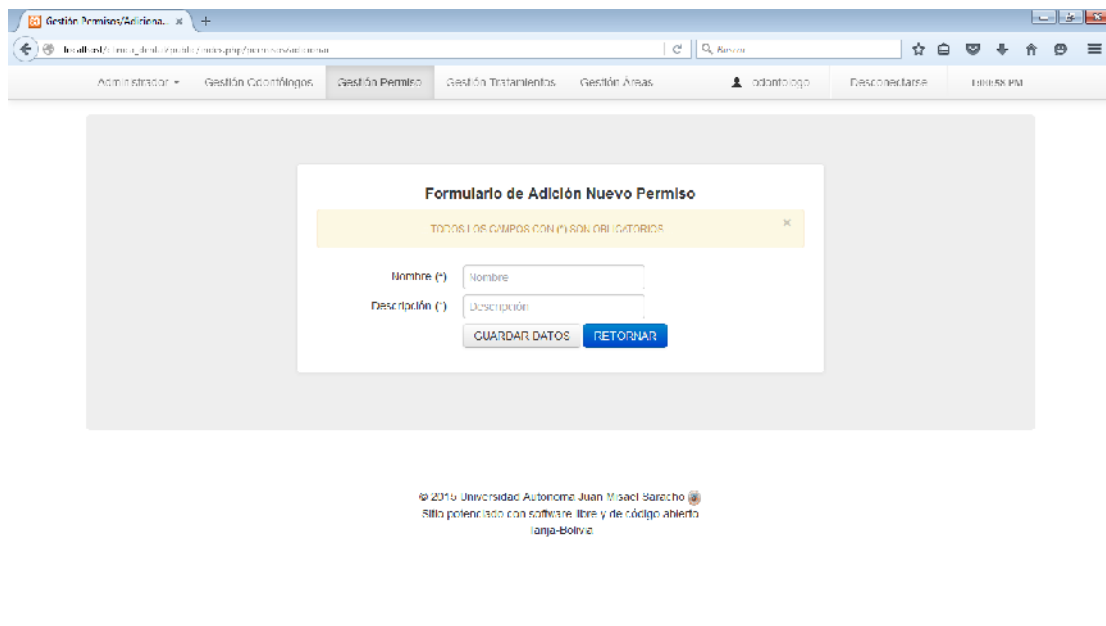


Figura 63 Pantalla Gestión Tratamientos

2.1.2.2.3.13.4.1.10 Pantalla Gestión Áreas

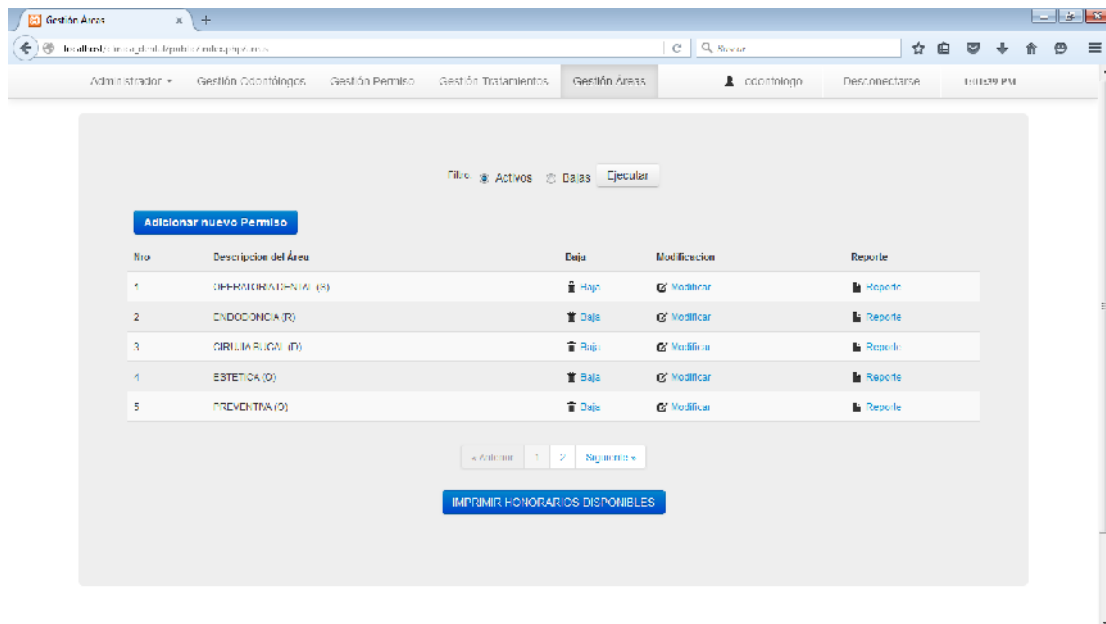


Figura 64 Pantalla Gestión Áreas

2.1.2.2.3.13.4.1.11 Pantalla Reporte

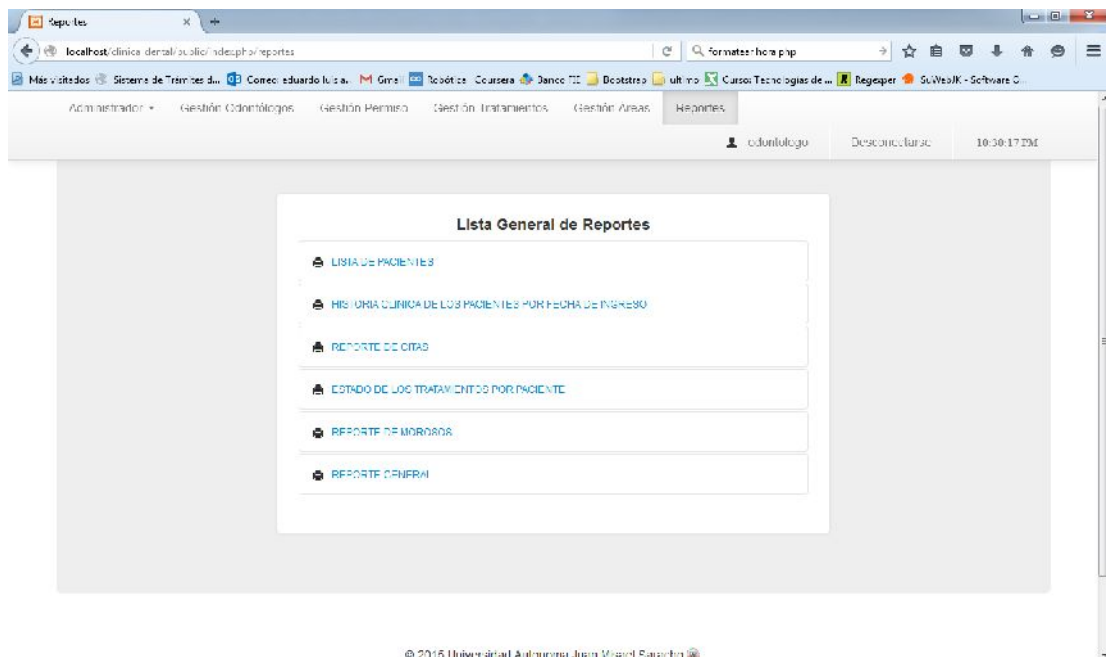


Figura 65 Pantalla Reportes

2.1.2.2.3.13.4.1.12 Reporte LISTA DE PACIENTES

CLINICA DENTAL "IRACEMA"
Patricia Trigueros A.

Lista: 06/06/2016
Pag: 13/137116
Hor: 02:27

NOMBRE PATROIA TRIGUERO	DIRECCION: CALLE FELIX RCTC N° 111
TELEFONO:	AUTORIZADO POR: EDITH RODRIGUEZ PINHEIRO ANDRÉ
CATEGORIA: TIENFERA	SEXO: FEMENINO
FECHA DE NACIMIENTO: 01/01/52	FECHA DE INGRESO: 14/06/2016
NOMBRE: MARIO BLACUTT	DIRECCION: CALLE FELIX RCTC N° 111
TELEFONO:	AUTORIZADO POR:
CATEGORIA: DOCTOR	SEXO: MASCULINO
FECHA DE NACIMIENTO: 01/01/52	FECHA DE INGRESO: 14/06/2016
NOMBRE: MARTA JACAYO REFORO	DIRECCION: SAN ANTONIO GAYOROSA 3100
TELEFONO: 7012331	AUTORIZADO POR:
CATEGORIA: ESTUDIANTE	SEXO: MASCULINO
FECHA DE NACIMIENTO: 01/01/16	FECHA DE INGRESO: 16/06/2016

Figura 66 Reporte Lista de Pacientes

2.1.2.2.3.13.4.1.13 Reporte HISTORIA CLINICA DE LOS PACIENTES POR FECHA DE INGRESO

CLINICA DENTAL "IRACEMA"
Patricia Trigueros A.

Lista: 06/06/2016
Pag: 13/137116
Hor: 02:30

HISTORIA CLINICAS DE LOS PACIENTES INGRESADOS DESDE 01/01/1985 HASTA 31/12/2016

NOMBRE PATROIA TRIGUERO	DIRECCION: CALLE FELIX RCTC N° 111
TELEFONO:	AUTORIZADO POR: EDITH RODRIGUEZ PINHEIRO ANDRÉ
CATEGORIA: TIENFERA	SEXO: FEMENINO
FECHA DE NACIMIENTO: 01/01/52	FECHA DE INGRESO: 14/06/2016
NOMBRE: MARIO BLACUTT	DIRECCION: CALLE FELIX RCTC N° 111
TELEFONO:	AUTORIZADO POR:
CATEGORIA: DOCTOR	SEXO: MASCULINO

Figura 67 Reporte Historia clínica de los pacientes por fecha de ingreso

2.1.2.2.3.13.4.1.14 Reporte de citas

The screenshot shows a web browser window with the following tabs: 'Reportes', 'REPORTE DE LAS CITAS DE...', 'HB ORA CLINICAS DE...', and 'LISTA DE PACIENTES AC...'. The address bar shows 'localhost/clinica_dental/suafco/adespho/reportes/reportes/0585-01-31/2016-12-31'. The browser's address bar contains 'formata=html.php'. The browser's address bar also shows 'Más visitados', 'Sistema de Trámites d...', 'Conoc: eduardo luis a.', 'Giro', 'Robótica', 'Coursara', 'Danco TI', 'Dcostrap', 'ultim', 'Curso: Tecnologías de...', 'Regeper', and 'SuWebK - Software C...'. The browser's address bar also shows 'Página: 1 de 1' and 'Tamaño automático'.

The report is titled 'CLINICA DENTAL "TRACEMA" Patricia Trigueros A.' and is dated 'Jueves, 12 de Agosto, Fecha: 12/08/2016 Hora: 09:41'. The report is titled 'REPORTE DE LAS CITAS DESDE 01/01/1985 HASTA 31/12/2016'.

The report is divided into two sections: 'DOCTOR EDUARDO LUIS ALVAREZ' and 'DOCTOR EDUARDO PEREZ'. Each section contains a table with columns: 'PACIENTE', 'ESTADO', 'I ESDE', 'TIEMPO', and 'CONSERVACIONES'.

DOCTOR EDUARDO LUIS ALVAREZ				
PACIENTE	ESTADO	I ESDE	TIEMPO	CONSERVACIONES
PATRICIA TRIGUEROS	ASISTIDO	11/11/11	31 min	De suma urgencia
PATRICIA TRIGUEROS	CONFIRMADO	35/10/11	20 min.	
CECILIA FERRAZO ALDANA DEARDO	CONFIRMADO	35/10/11	20 min.	
MARIC ELACUET	CONFIRMADO	37/11/11	31 min	
MARIC ELACUET	NO ASISTIDO	14/11/11	31 min	Se le acordó
MARIC ELACUET	PENALIZADO	16/11/11	31 min.	De suma urgencia
PATRICIA TRIGUEROS	NO ASISTIDO	14/11/11	31 min	Se le acordó
PATRICIA TRIGUEROS	PENALIZADO	15/11/11	31 min	De suma urgencia
PATRICIA TRIGUEROS	PENALIZADO	11/11/11	20 min.	Diagnóstico

DOCTOR EDUARDO PEREZ				
PACIENTE	ESTADO	I ESDE	TIEMPO	CONSERVACIONES
PATRICIA TRIGUEROS	CONFIRMADO	11/11/11	20 min.	Exámen
PATRICIA TRIGUEROS	PENALIZADO	35/10/11	20 min.	De suma urgencia
MARIC ELACUET	PENALIZADO	35/10/11	10 min.	De suma urgencia
MARIC ELACUET	CONFIRMADO	11/10/11	10 min.	Diagnóstico

Figura 68 Reporte de citas

2.1.2.2.3.13.4.1.15 Estado de los tratamientos por paciente

CLINICA DENTAL "IRACEMA"		HISTORIA CLINICA ODONTOLOGICA			
<i>Patricia Iniguez A.</i>					
DATOS PERSONALES					
Paciente: PATRICIA INIGUEZ			Tel.:		
Domicilio: CALLE FELIX SOTO N° 1111			Fecha de Ingreso: 14-06-2012		
Email:			Edad: 34 años		
ANTECEDENTES PATOLOGICOS FAMILIARES					
Antecedente familiar:					
Enfermedades de infancia:					
Forma de los dientes: NORMAL					
Color de los dientes: AMARILLENTO					
Tamaño de los dientes:					
Tipo de Sangre:		Tiempo de Coagulación: 1 min.		Tiempo de Sangría: 1 min.	
TRATAMIENTO PRESUPUESTO					
OPERATORIA DENTAL		100			
ENDODONCIA		450			
CIRUJA BUCAL		0			
ESTETICA		30			
PREVENTIVA		0			
GENERAL		0			
TOTAL:		780			
EVOLUCIÓN Y TRATAMIENTO					
Fecha	Pos. Dentaria	Diagnostico	Tratamiento	Estado	Observaciones
11-05-2016	15	CARIES 1°	Preparación	Ejecutado y Canalado	
11-05-2016	32	OBSTRUCCION	Obturación Compuesta de Amalg	Ejecutado totalmente	Refieno de plata
13-05-2016	22	CARIES 1°	Molero	Ejecutado y Canalado	Se puso protector
13-05-2016	42	LIMPIEZA GENERAL	LIMPIEZA	Ejecutado y Canalado	limpieza de surro
Autorización del Paciente: EDUARDO LUIS AGUIRRE GUDIÑO					
.....					
Firma Espacio					
.....					
Pagina 1/1					

Figura 69 ESTADO DE LOS TRATAMIENTOS POR PACIENTE

2.1.2.2.3.13.4.1.16 Pantalla Gestión Pacientes

Odontólogo - Gestión Pacientes - Asignación de Citas - Gestión Examen Clínico - Gestión Presupuestos - Odontólogo - Desconectarse - 11:24 AM

Filtros: Activos Datas CEDULA DE IDENTIDAD

[Agregar nuevo Paciente](#)

No.	Cedula	Apellidos Nombre	Donar	Modificar	Antecedentes	Usuario	Permisos	Reporte
1	7142117	IBÁÑEZ PAINEZA	<input type="button" value="Donar"/>	<input type="button" value="Modificar"/>	<input type="button" value="Antecedentes"/>	<input checked="" type="checkbox"/> Modificar	<input type="button" value="Permisos"/>	<input type="button" value="Reporte"/>
2	7142332	DLACUTT IBARRO	<input type="button" value="Donar"/>	<input type="button" value="Modificar"/>	<input type="button" value="Antecedentes"/>	<input checked="" type="checkbox"/> Asignar	<input type="button" value="Permisos"/>	<input type="button" value="Reporte"/>
3	7646448	ARROYO RINOSO MITAS	<input type="button" value="Donar"/>	<input type="button" value="Modificar"/>	<input type="button" value="Antecedentes"/>	<input checked="" type="checkbox"/> Asignar	<input type="button" value="Permisos"/>	<input type="button" value="Reporte"/>
4	7688316	ALDANA PIZARRO COCHAR FERNANDO	<input type="button" value="Donar"/>	<input type="button" value="Modificar"/>	<input type="button" value="Antecedentes"/>	<input checked="" type="checkbox"/> Asignar	<input type="button" value="Permisos"/>	<input type="button" value="Reporte"/>
5	7777747	AVILA GALLO JULIO OSCAR	<input type="button" value="Donar"/>	<input type="button" value="Modificar"/>	<input type="button" value="Antecedentes"/>	<input checked="" type="checkbox"/> Asignar	<input type="button" value="Permisos"/>	<input type="button" value="Reporte"/>

« Anterior 1 2 3 4 5 6 7 8 9 Siguiente »

© 2015 Universidad Autónoma Juan Misael Saracho

Figura 70 Pantalla Gestión Pacientes

2.1.2.2.3.13.4.1.17 Pantalla Registro de Paciente

Odontólogo - Gestión Pacientes - Asignación de Citas - Gestión Examen Clínico - Gestión Presupuestos - Odontólogo - Desconectarse - 11:24 AM

Formulario de Registro de Paciente

¡¡ TODOS LOS CAMPOS CON (*) SON OBLIGATORIOS !!

C.I. (*)

Nombres (*)

Apellido Paterno (*)

Apellido Materno

Dirección (*)

Fecha de Nacimiento (*) 1 Enero 1980

Sexo (*) Femenino Masculino

Email

Telefono

Autoriza

Ocupación

Figura 71 Pantalla Registro de Paciente

2.1.2.2.3.13.4.1.18 Pantalla Modificar Registro

Formulario de Modificar Datos del Paciente

TODOS LOS CAMPOS CON (*) SON OBLIGATORIOS

C.I. /143912

Nombres (*) IBAÑUELA

Apellido Paterno (*) IBÁÑUEZ

Apellido Materno Apellido Materno

Dirección (*) CALLE FERRER SOTO Nº 1111

Fecha de Nacimiento (*) 1 Enero 1980

Sexo (*) Femenino Masculino

Email Email

Teléfono Teléfono

Autómat EDUARDO LUIS AGUIRRE GUDIÑ

Ocupación INGENIERA

GUARDAR DATOS RETORNAR

Figura 72 Pantalla Modificar Registro

2.1.2.2.3.13.4.1.19 Pantalla Antecedentes

Formulario de Registro de Antecedentes del Paciente

Fecha de Ingreso 2016-03-01

Ant. familiares Antecedentes familiares

Enfer. de infancia Enfermedades de infancia

Forma Forma de los dientes

Color Color de los dientes

Tamaño Tamaño de los dientes

Tipo de sangre Tipo de Sangre

Tiempo de coagulación Tiempo de coagulación min.

Tiempo de sangría Tiempo de sangría min.

GUARDAR DATOS RETORNAR

Figura 73 Pantalla Antecedentes

2.1.2.2.3.13.4.1.20 Pantalla Reporte

CLINICA DENTAL "IRACEMA"
Patricia Iniguez A.
 Reporte generado el 01-03-2016 17:06
HISTORIA CLINICA ODONTOLOGICA

DAFOS PERSONALES
 Paciente: PATRICIA INIGUEZ Telf.:
 Domicilio: CALLE FELIX SOTO N° 1111 Fecha de Ingreso: 30-11-0001
 Email: Edad: 34 años

ANTECEDENTES PATOLOGICOS FAMILIARES
 Antecedente familiar:
 Enfermedades de infancia:
 Forma de los dientes:
 Color de los dientes: AMARILLENTOS
 Tamaño de los dientes: NORMAL
 Tipo de Sangre: OR Tiempo de Coagulación: 5 min. Tiempo de Sangría: 3 min.

Autorización del Paciente: EDUARDO LUIS AGUIRRE GUIDINO

 Firma Espacio

Figura 74 Pantalla Reporte

2.1.2.2.3.13.4.1.21 Pantalla Asignación de Citas

Asignación de Citas

Calendario Gestión Pacientes **Asignación de Citas** Gestión Examen Clínica Gestión Postoperatorio Perfil de Usuario Documentos 1:05:10 PM

AGENDA DE FECHA 01/03/2016
 Citas del Doctor: EDUARDO LUIS AGUIRRE +Asignar cita

Marzo 2016						
Lu	ma	mi	Jue	vi	dom	dom
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Citas de Marzo:

- 20160310 09:30
 Paciente: JHITHI CS-SARAWIA ALU 110
 Descripción: Tiempo:30 min. Estado:Confirmado
- 20160310 09:00
 Paciente: MARIAS ANKITO BERNIZO
 Descripción: Tiempo:50 min. Estado:Confirmado
- 20160316 09:30
 Paciente: MARIO OLACUTT
 Descripción: Tiempo:50 min. Estado:Confirmado

Figura 75 Pantalla Asignación de Citas

2.1.2.2.3.13.4.1.22 Pantalla Gestión Examen Clínico

Inicio | **Gestión Examen Clínico** | + | Localhost:clínica_dermatología/index.php/examenclínico_pacientes=ab | Buscar | 1:10:48PM

Colembolojo | Gestión Pacientes | Asignado De Clases | **Gestión Examen Clínico** | Gestión Presupuestos

Orolograma del Paciente: Felisa Guadalupe

Agregar Diagnóstico
 Marcar ausente
 Marcar utilizada

Área

Selección Área

Descripción

Imagenes

Color

<p> <input checked="" type="radio"/> 15 <input type="radio"/> 16 <input type="radio"/> 17 <input type="radio"/> 18 <input type="radio"/> 19 <input type="radio"/> 20 <input type="radio"/> 21 <input type="radio"/> 22 <input type="radio"/> 23 <input type="radio"/> 24 <input type="radio"/> 25 <input type="radio"/> 26 <input type="radio"/> 27 <input type="radio"/> 28 <input type="radio"/> 29 <input type="radio"/> 30 </p>	<p> <input type="radio"/> 31 <input type="radio"/> 32 <input type="radio"/> 33 <input type="radio"/> 34 <input type="radio"/> 35 <input type="radio"/> 36 <input type="radio"/> 37 <input type="radio"/> 38 <input type="radio"/> 39 <input type="radio"/> 40 <input type="radio"/> 41 <input type="radio"/> 42 <input type="radio"/> 43 <input type="radio"/> 44 <input type="radio"/> 45 <input type="radio"/> 46 <input type="radio"/> 47 <input type="radio"/> 48 <input type="radio"/> 49 <input type="radio"/> 50 </p>	<p> <input type="radio"/> 51 <input type="radio"/> 52 <input type="radio"/> 53 <input type="radio"/> 54 <input type="radio"/> 55 <input type="radio"/> 56 <input type="radio"/> 57 <input type="radio"/> 58 <input type="radio"/> 59 <input type="radio"/> 60 <input type="radio"/> 61 <input type="radio"/> 62 <input type="radio"/> 63 <input type="radio"/> 64 <input type="radio"/> 65 <input type="radio"/> 66 <input type="radio"/> 67 <input type="radio"/> 68 <input type="radio"/> 69 <input type="radio"/> 70 <input type="radio"/> 71 <input type="radio"/> 72 <input type="radio"/> 73 <input type="radio"/> 74 <input type="radio"/> 75 <input type="radio"/> 76 <input type="radio"/> 77 <input type="radio"/> 78 <input type="radio"/> 79 <input type="radio"/> 80 </p>	<p> <input type="radio"/> 81 <input type="radio"/> 82 <input type="radio"/> 83 <input type="radio"/> 84 <input type="radio"/> 85 <input type="radio"/> 86 <input type="radio"/> 87 <input type="radio"/> 88 <input type="radio"/> 89 <input type="radio"/> 90 <input type="radio"/> 91 <input type="radio"/> 92 <input type="radio"/> 93 <input type="radio"/> 94 <input type="radio"/> 95 <input type="radio"/> 96 <input type="radio"/> 97 <input type="radio"/> 98 <input type="radio"/> 99 <input type="radio"/> 100 </p>
---	--	--	---

Atención!
 Recuerde que para así ingresar al Orolograma de pacientes por primera vez el programa no le dejará volver a ingresar otro Orolograma a un paciente que ya tiene registro, para eso es necesario ir a la pestaña GESTIÓN PACIENTES.

© 2013 Universidad Autónoma Juan Misael Solís
 Sitio desarrollado con esfuerzos libres y de código abierto

Figura 76 Pantalla Gestión Examen Clínico

2.1.2.2.3.13.4.1.23 Pantalla Presupuesto

Gección Presupuesto

Inicio / clínic / clinica_dental / pa_bic / index.php / presupuesto / paciente = 1

Odontólogo | Casión Pacientes | Asignació De Citas | Casión Camien Clínico | Casión Presupuestos

Buscar | Desconectarse | 11:26 PM

Odontólogo

Presupuesto del Paciente: PATRICKA MUELZ

--	--

Presupuesto

Queriado	400
Aceptado	400

© 2016 Universidad Autónoma Juan Misao Borja
Sito desarrollado con software libre y de código abierto
Tariá-Borja

Figura 77 Pantalla Presupuesto

2.1.2.2.3.13.4.1.24 Pantalla Seguimiento de Tratamientos

Seguimiento del paciente: PATRICIA NIQUIZ

DET.	CAVIDAD	CARA	DIAGNOSTICO	TRATAMIENTO	COSTO
1	SEGUNDO BICUSPIDE O PREMOLAR-15	centro	CARIES 1*	Premolares	300
2	LATERAL-22	centro	CARIES 1*	Molares	350
4	LATERAL-42	centro	LIMPIEZA GENERAL	LIMPIEZA	30
5	LATERAL-32	centro	OBTURACION	Obturación Compuesta de Amalga	100

Fecha: 2016-03-14

Estado:

- Sin ejecutar
- Ejecutado parcialmente
- Ejecutado totalmente
- Ejecutado y Cancelado

Ejecutado por: AGUIRRE EDUARDO LUIS

Observaciones:

Figura 78 Seguimiento de tratamientos

2.1.2.2.3.13.4.1.25 Pantalla Backup

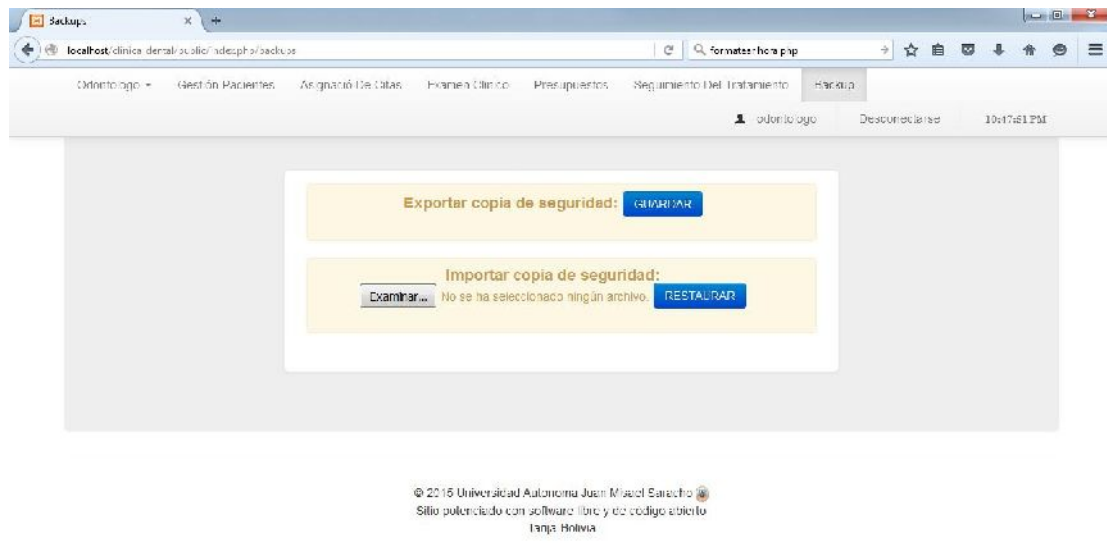


Figura 79 Pantalla Backup

2.1.2.2.3.13.4.1.26 Pantalla Gestión de Generación de Facturas

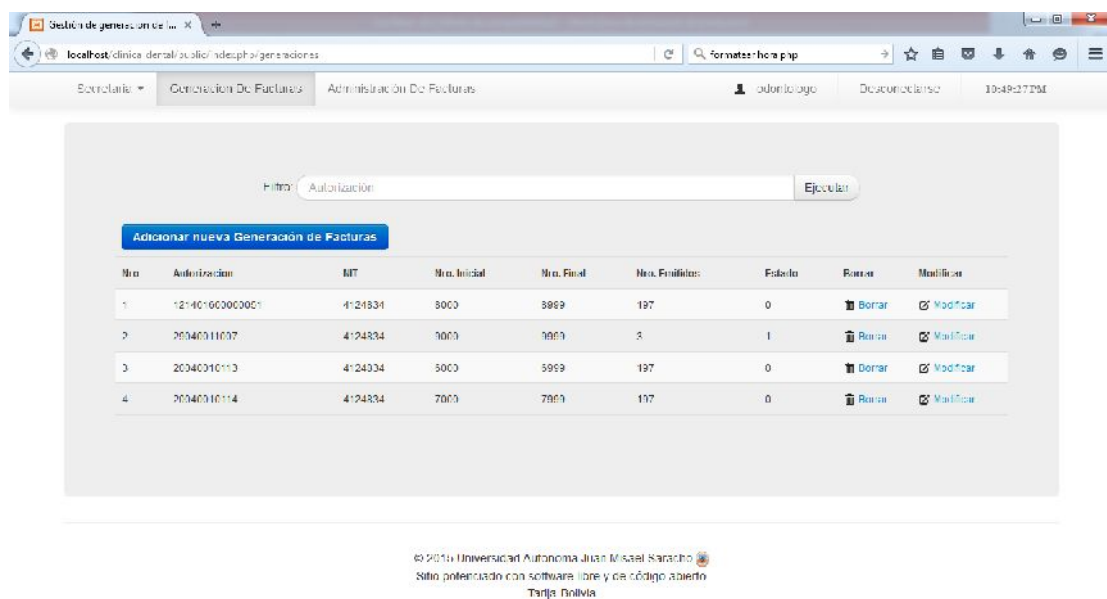


Figura 80 Pantalla Gestión de Facturas

2.1.2.2.3.13.4.1.27 Pantalla Adicionar Generación de Factura

Formulario de Adicionar Facturas

ⓘ (TODOS LOS CAMPOS CON (*) SON OBLIGATORIOS)

AUTORIZACION (*)

NIT (*)

N° INICIO FACT (*)

N° FIN FACTURA (*)

FECHA DE EMISION (*) 1 Enero 2016

LLAVE DISTRIB (*)

© 2015 Universidad Autónoma Juan Misael Saracho
Sitio potenciado con software libre y de código abierto

Figura 81 Pantalla Adicionar Generación de Factura

2.1.2.2.3.13.4.1.28 Pantalla Administración de Factura

Administración de Facturas

Filtrar Activos Bajos

Nro	CODIGO	AUTORIZACION	NITICI	FECHA	NOMBRE	Revisar	Modificar	Factura
1	9301	26940311037	7143812	2015-03-13	PATRICIA INIGUEZ	<input type="button" value="Revisar"/>	<input type="button" value="Modificar"/>	<input type="button" value="Factura"/>
2	9302	26940311037	7143830	2015-03-13	MARIO RIACULT	<input type="button" value="Revisar"/>	<input type="button" value="Modificar"/>	<input type="button" value="Factura"/>
3	9303	26940311037	7143812	2015-03-13	PATRICIA INIGUEZ	<input type="button" value="Revisar"/>	<input type="button" value="Modificar"/>	<input type="button" value="Factura"/>

© 2015 Universidad Autónoma Juan Misael Saracho
Sitio potenciado con software libre y de código abierto
Tariju-Buliviú

Figura 82 Pantalla Administración de Factura

2.1.2.2.3.13.4.1.29 Pantalla Administración de Factura

Adicionar Facturas

AUTORIZACION: 29040011007

NIT: 4124304

Nro de Factura: 0004

Carpet Identidad (*): 71884417

Nombre (*): Natalia Suarez Garduño

NIT CLIENTE: NIT 000

Fecha: 2016-05-14

CODIGO DE CONTROL: 99-94-A2-4A

TRATAMIENTOS

Tratamiento	Monto	Precio Real	Descrp.

TRATAMIENTOS - ACTUALES

Tratamiento	Monto	Precio Real	Descrp.
Obturación Compuesta	100	100	Obturación Compuesta

Totales: 100.00

Figura 83 Pantalla Adición de Factura

2.1.2.2.3.13.4.1.30 Pantalla Imprimir Factura

CLINICA DENTAL "IRACEMA" <i>Patricia Ñíguez A.</i> <small>CASA MATEIZ BARRIO MARCISO CAMPEÑO CALLE FELIX SOTO N° 1521 TELÉF. 6637182 CEL. 72395645 TARIJA-BOLIVIA</small>		NIT: 4124834 No. Factura: 9001 No. Autorización: 29040011007 ORIGINAL Servicios odontológicos integrales
FACTURA		
Lugar y fecha: TARIJA, 15/03/2016		
Señor(es): PATRICIA ÑIGUEZ		
NIT/CI: 7143912		
CODIGO	DESCRIPCION	SUBTOTAL
7	Premolares	150.00
Son: Ciento Cincuenta 00/100 Bolivianos.		TOTAL Bs.: 150.00
Codigo de Control: 06-1A-F2-78-0B		
Fecha Limite de Emision: 07/04/2016		
ESTA FACTURA CONTRIBUYE AL DESARROLLO DEL PAÍS. EL USO ILÍCITO DE ÉSTA SERÁ SANCIONADO DE ACUERDO A LEY.		
Ley N° 453: El prestador de servicio médico debe brindar atención de calidad al paciente.		

Figura 84 Pantalla Imprimir Factura

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

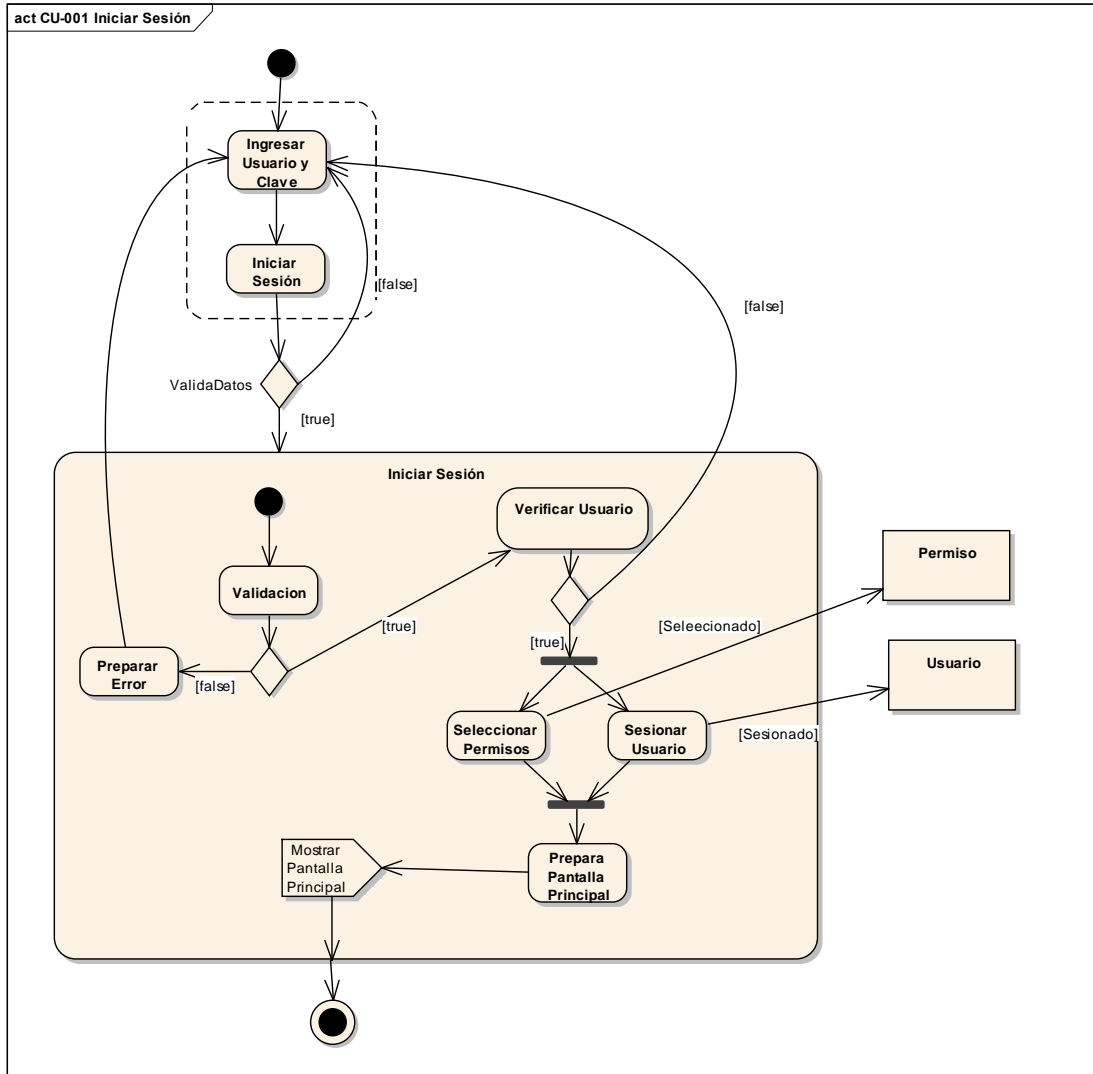


Figura 85 Diagrama de Actividad: CU-001 Iniciar Sesión

2.1.2.2.3.14.1.1.2 Diagrama de Actividad: CU-100 Gestión

Odontólogos

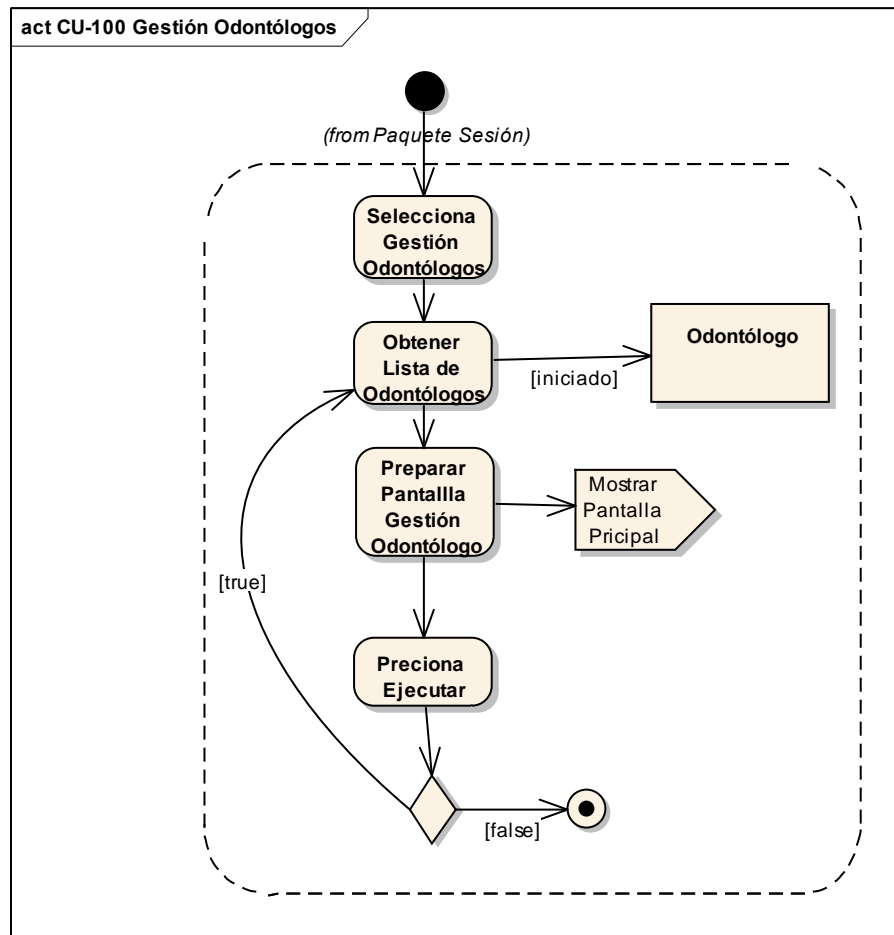


Figura 86 Diagrama de Actividad: CU-100 Gestión Odontólogos

2.1.2.2.3.14.1.1.3 Diagrama de Actividad: CU-101 Adicionar

Odontólogo

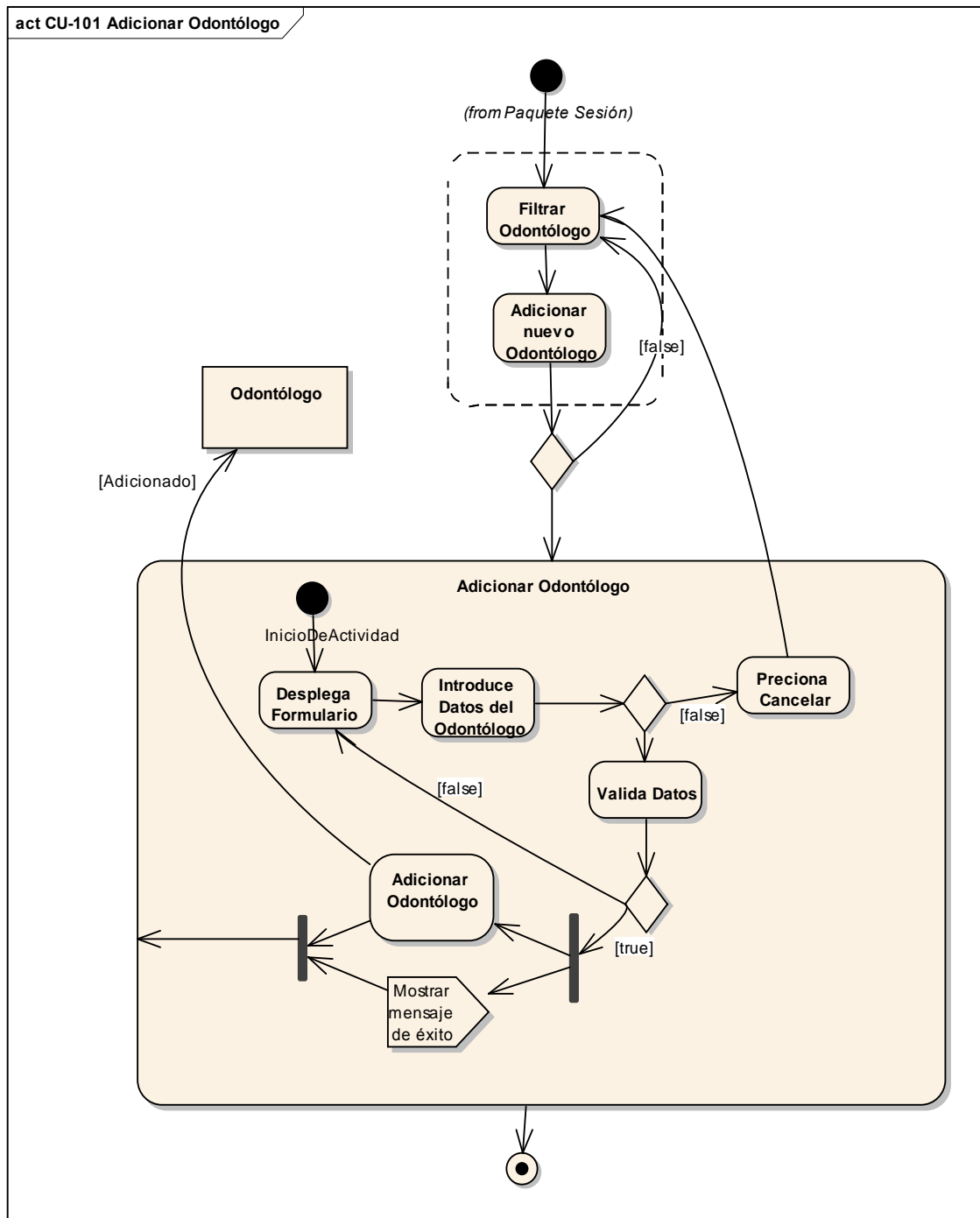


Figura 87 Diagrama de Actividad: CU-101 Adicionar Odontólogo

2.1.2.2.3.14.1.1.4 Diagrama de Actividad: CU-102 Modificar

Odontólogo

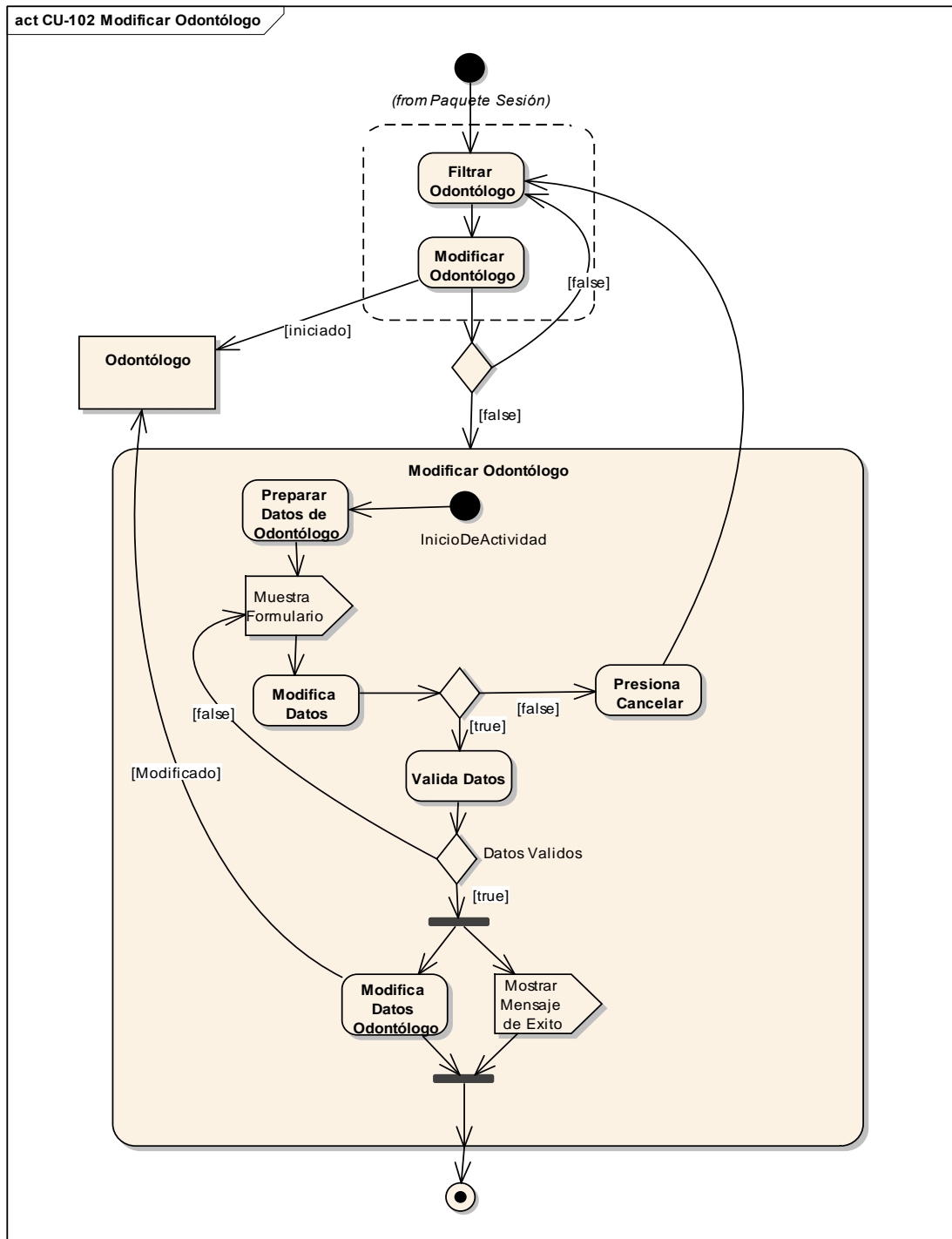


Figura 88 Diagrama de Actividad: CU-102 Modificar Odontólogo

2.1.2.2.3.14.1.1.5 Diagrama de Actividad: CU-103 Eliminar

Odontólogo

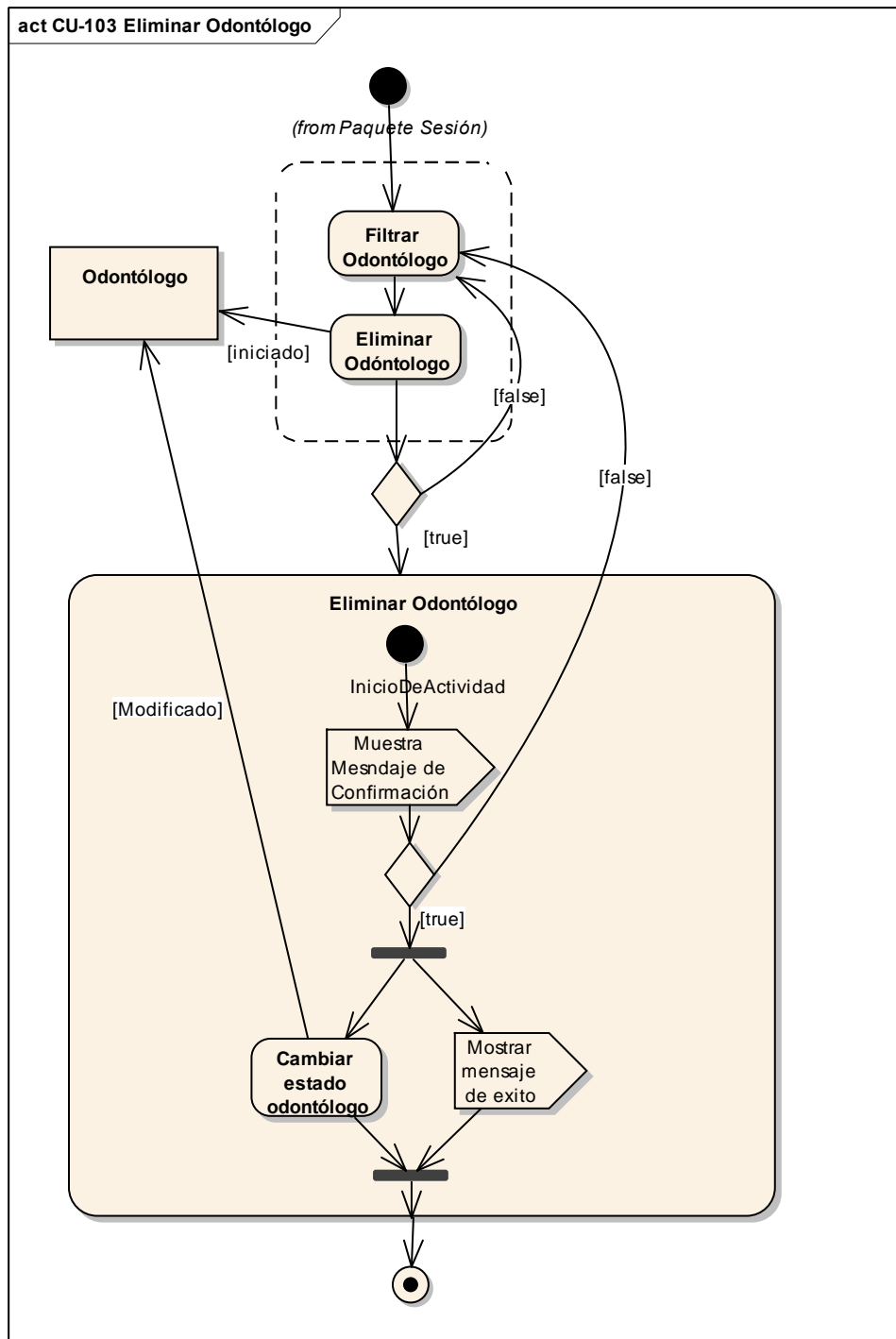


Figura 89 Diagrama de Actividad: CU-103 Eliminar Odontólogo

2.1.2.2.3.14.1.1.6 Diagrama de Actividad: CU-200 Gestión Permisos

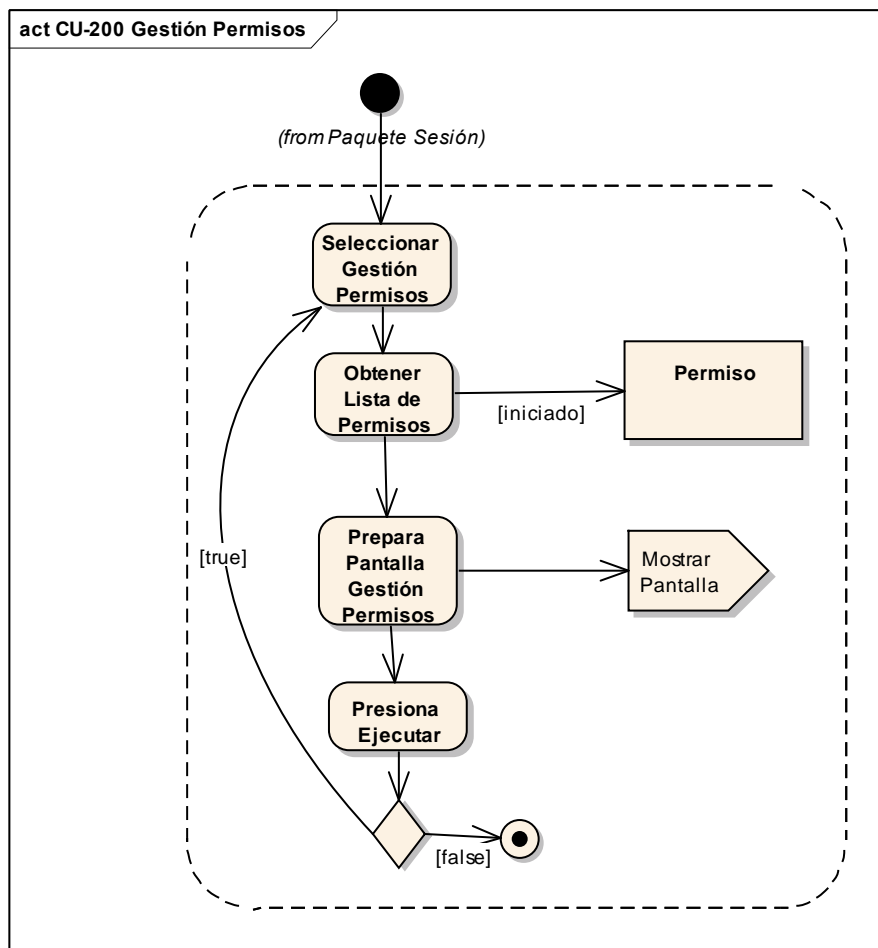


Figura 90 Diagrama de Actividad: CU-200 Gestión Permisos

2.1.2.2.3.14.1.1.7 Diagrama de Actividad: CU-300 Gestión

Tratamientos

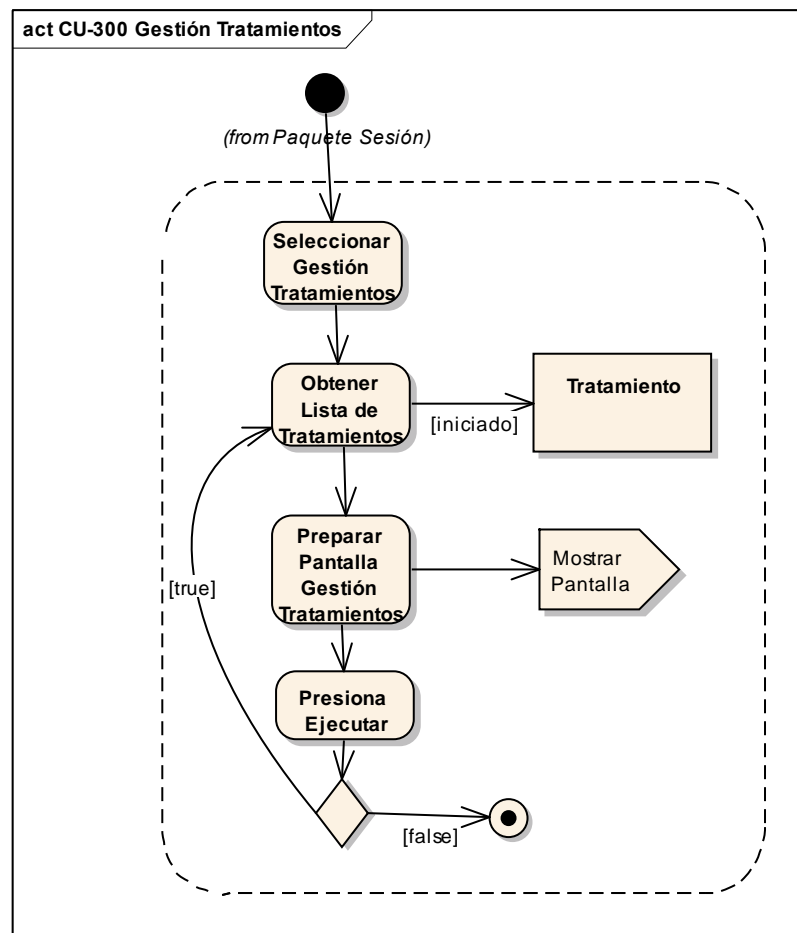


Figura 91 Diagrama de Actividad: CU-300 Gestión Tratamientos

2.1.2.2.3.14.1.1.8 Diagrama de Actividad: CU-400 Gestión Áreas

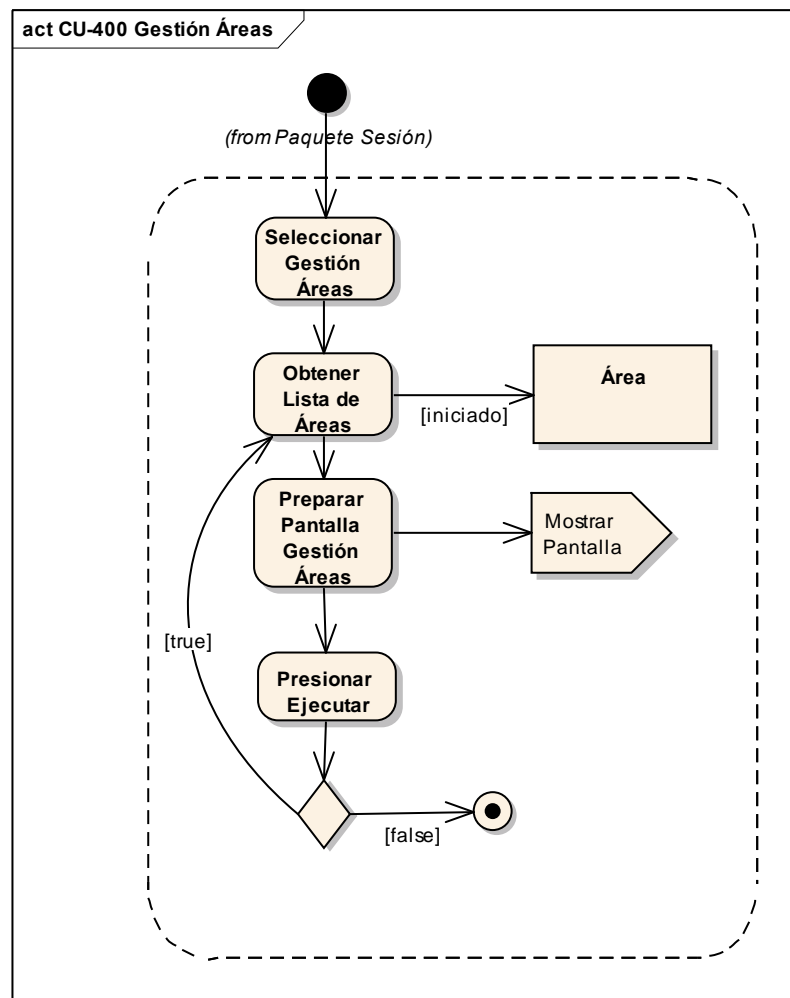


Figura 92 Diagrama de Actividad: CU-400 Gestión Áreas

2.1.2.2.3.14.1.1.9 Diagrama de Actividad: CU-500 Gestión Paciente

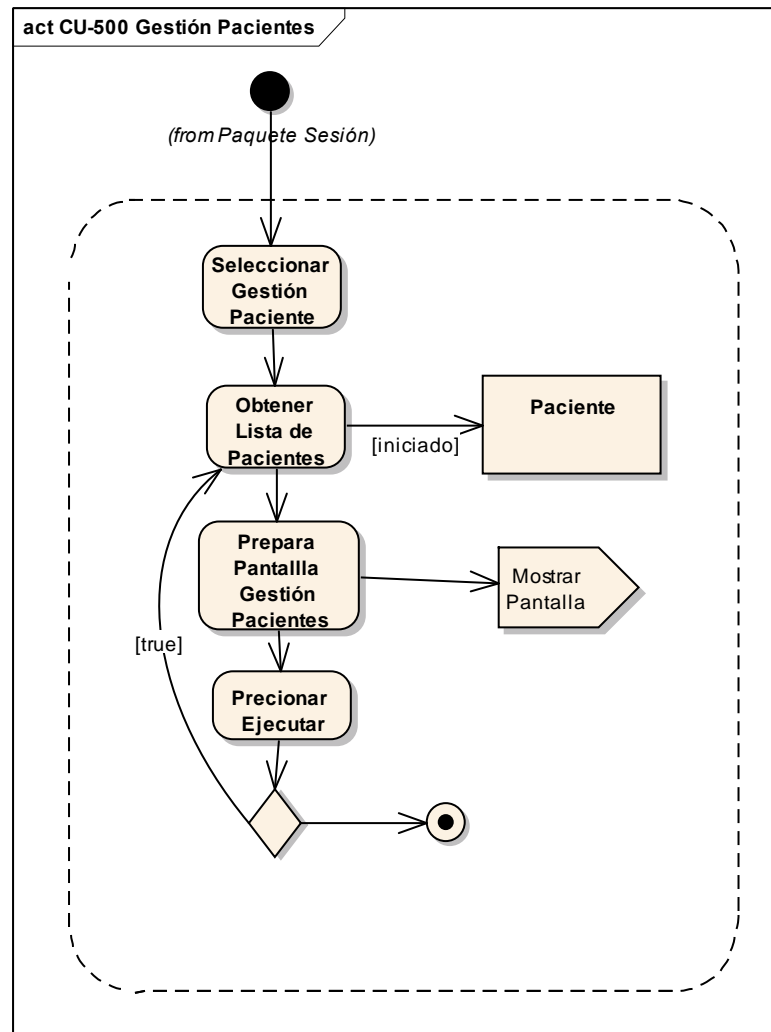


Figura 93 Diagrama de Actividad: CU-500 Gestión Paciente

2.1.2.2.3.14.1.1.10 Diagrama de Actividad: CU-501 Registro de

Paciente

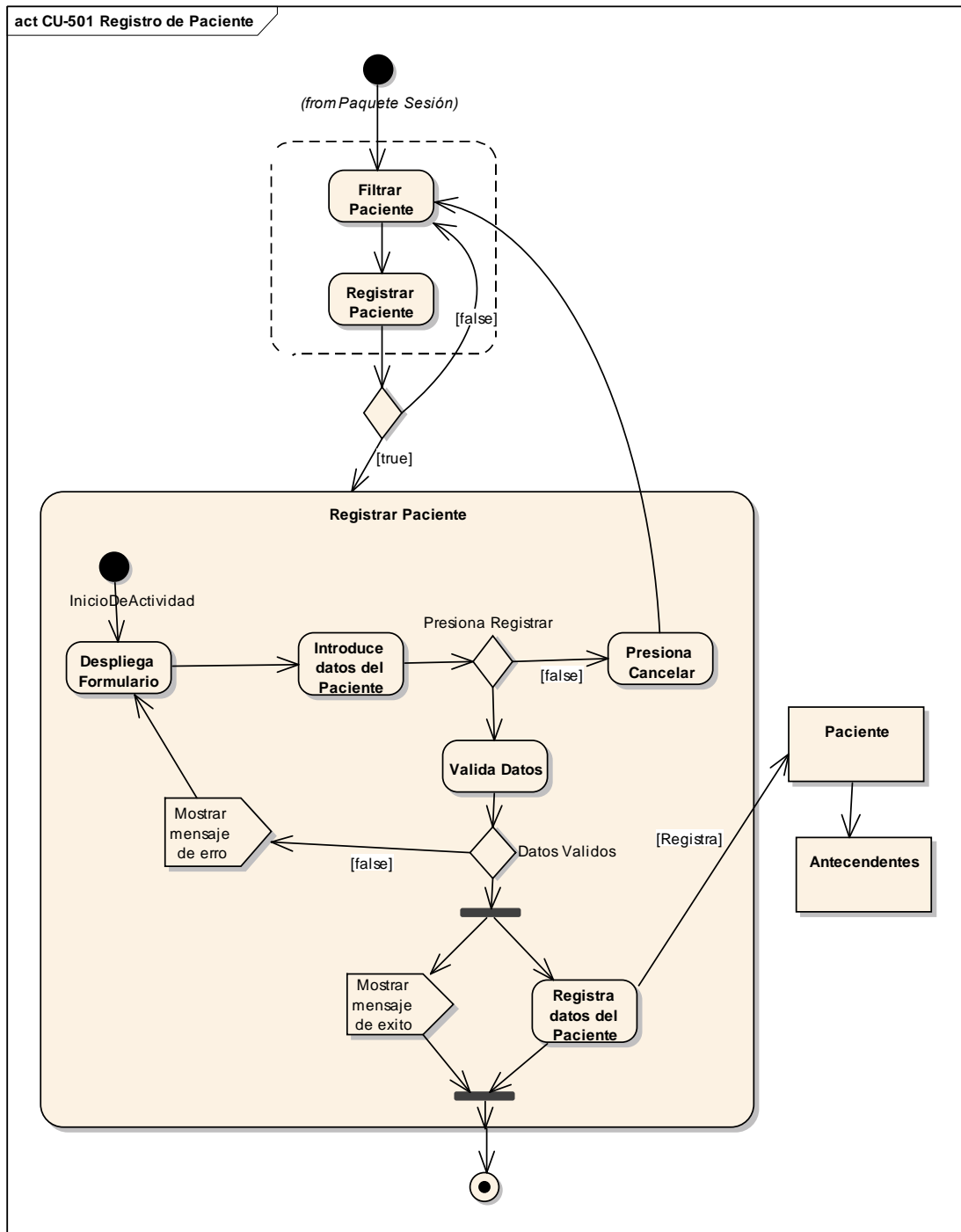


Figura 94 Diagrama de Actividad: CU-501 Registro de Paciente

2.1.2.2.3.14.1.1.11 Diagrama de Actividad: CU-502 Modificar

Registro

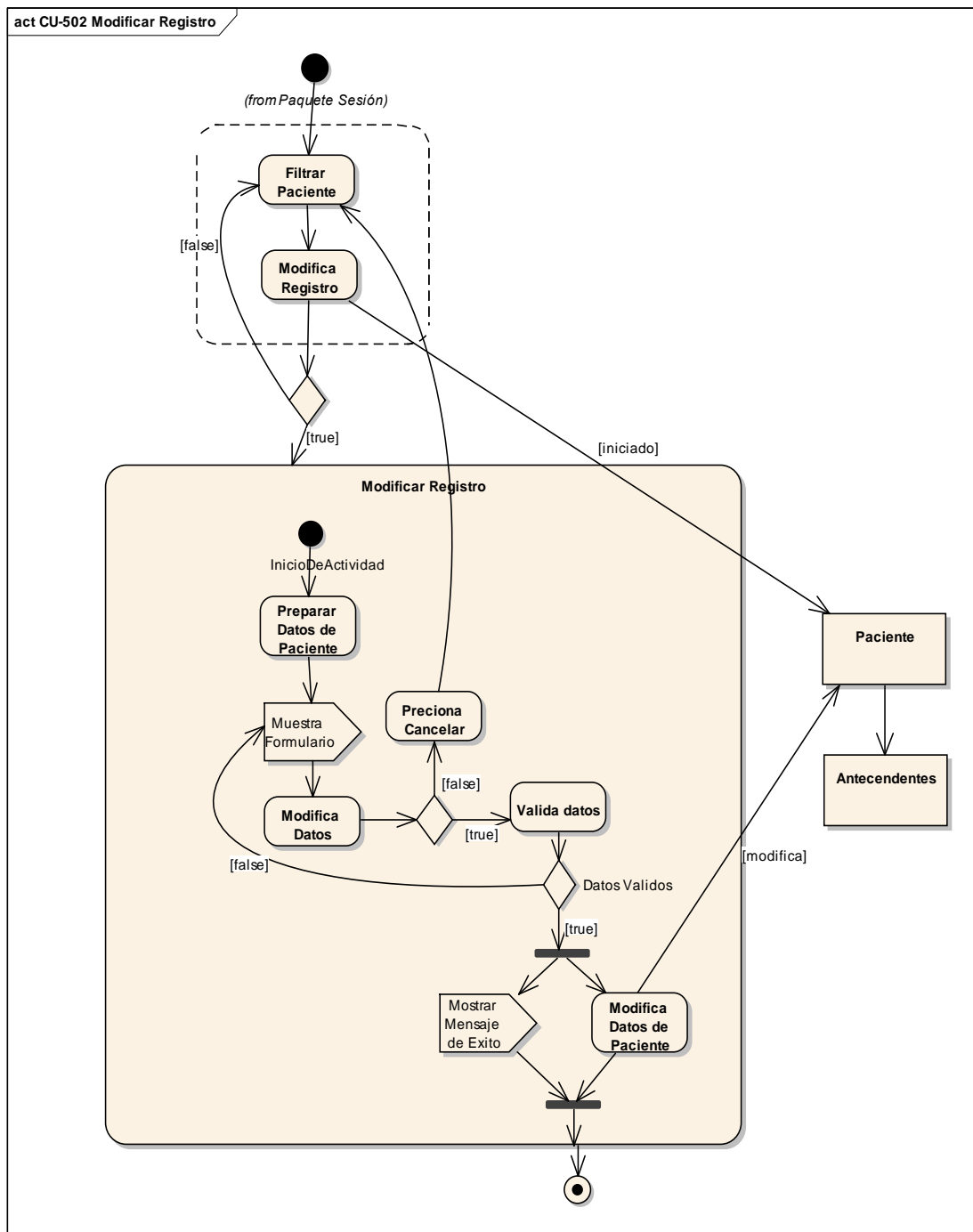


Figura 95 Diagrama de Actividad: CU-502 Modificar Registro

2.1.2.2.3.14.1.1.12

Diagrama de Actividad: CU-503

Asignar/Modificar Usuario

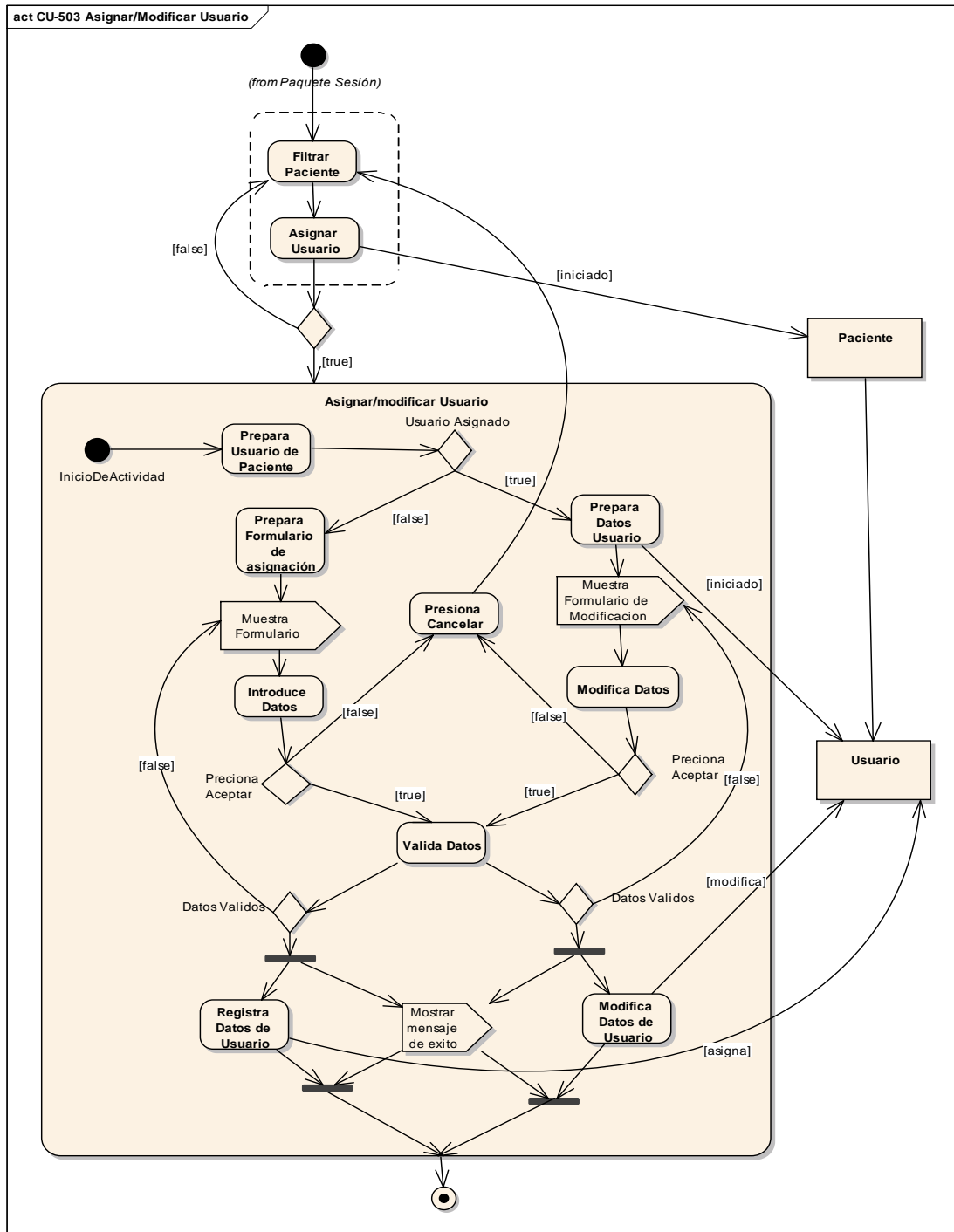


Figura 96 Diagrama de Actividad: CU-503 Asignar/Modificar Usuario

2.1.2.2.3.14.1.13

Diagrama de Actividad: CU-600 Asignación

de Citas

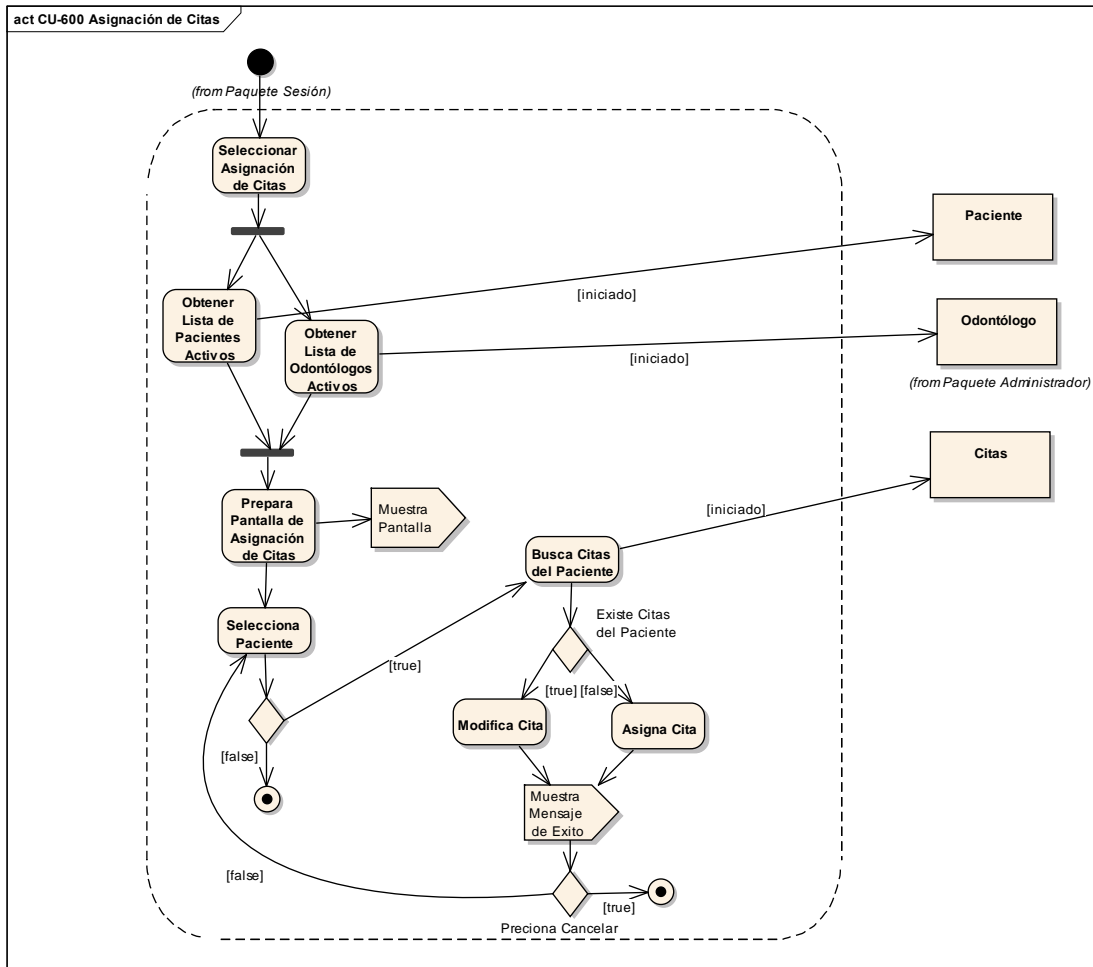


Figura 97 Diagrama de Actividad: CU-600 Asignación de Citas

2.1.2.2.3.14.1.14

Diagrama de Actividad: CU-700 Gestión

Examen Clínico

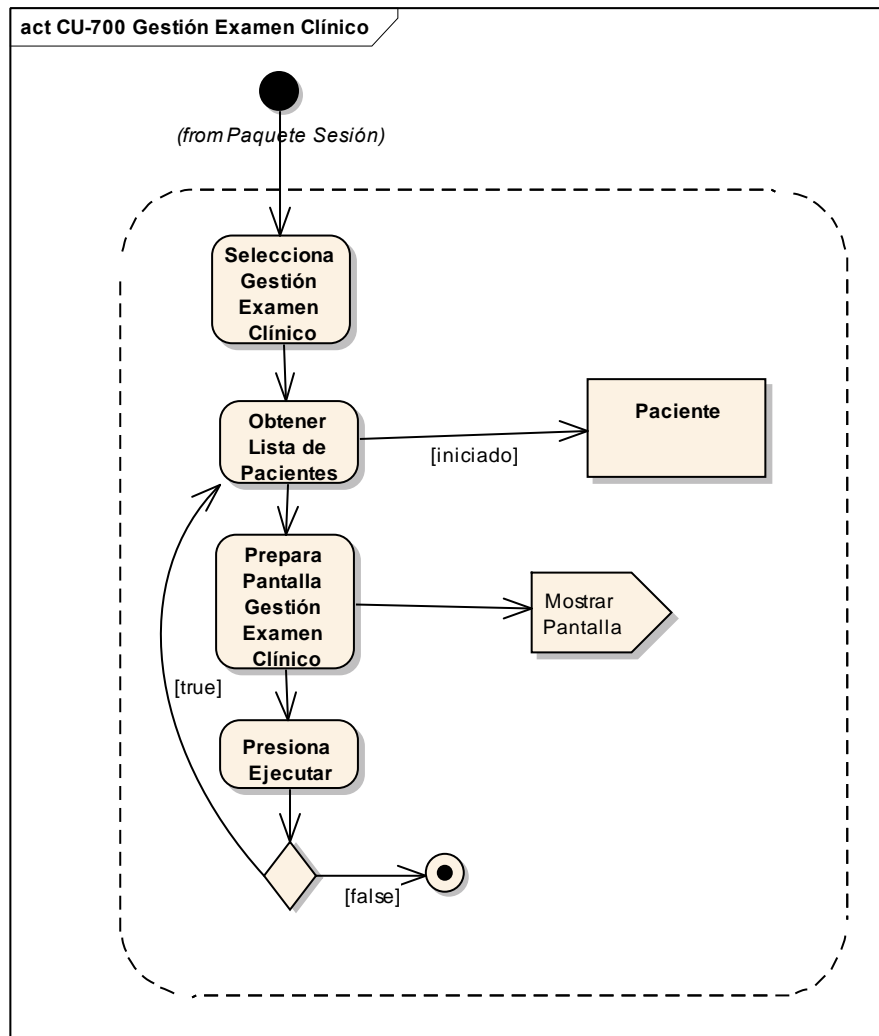


Figura 98 Diagrama de Actividad: CU-700 Gestión Examen Clínico

2.1.2.2.3.14.1.1.15 Diagrama de Actividad: CU-800 Gestión

Presupuestos

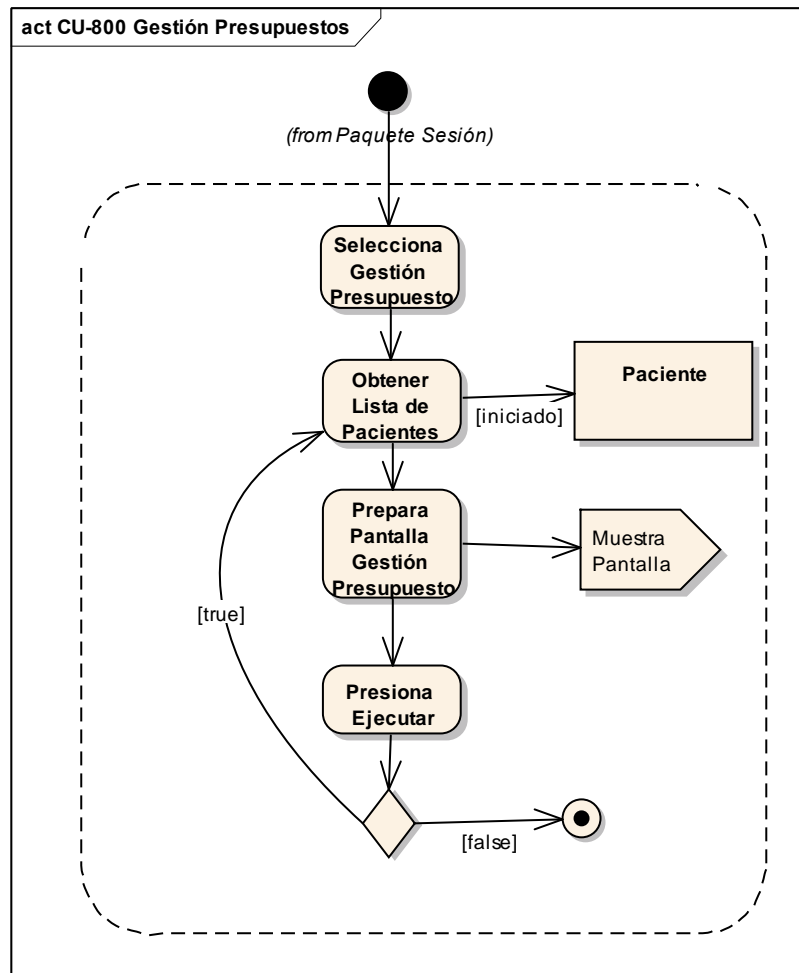


Figura 99 Diagrama de Actividad: CU-800 Gestión Presupuestos

2.1.2.2.3.14.1.1.16 Diagrama de Actividad: CU-900 Consultar

Información

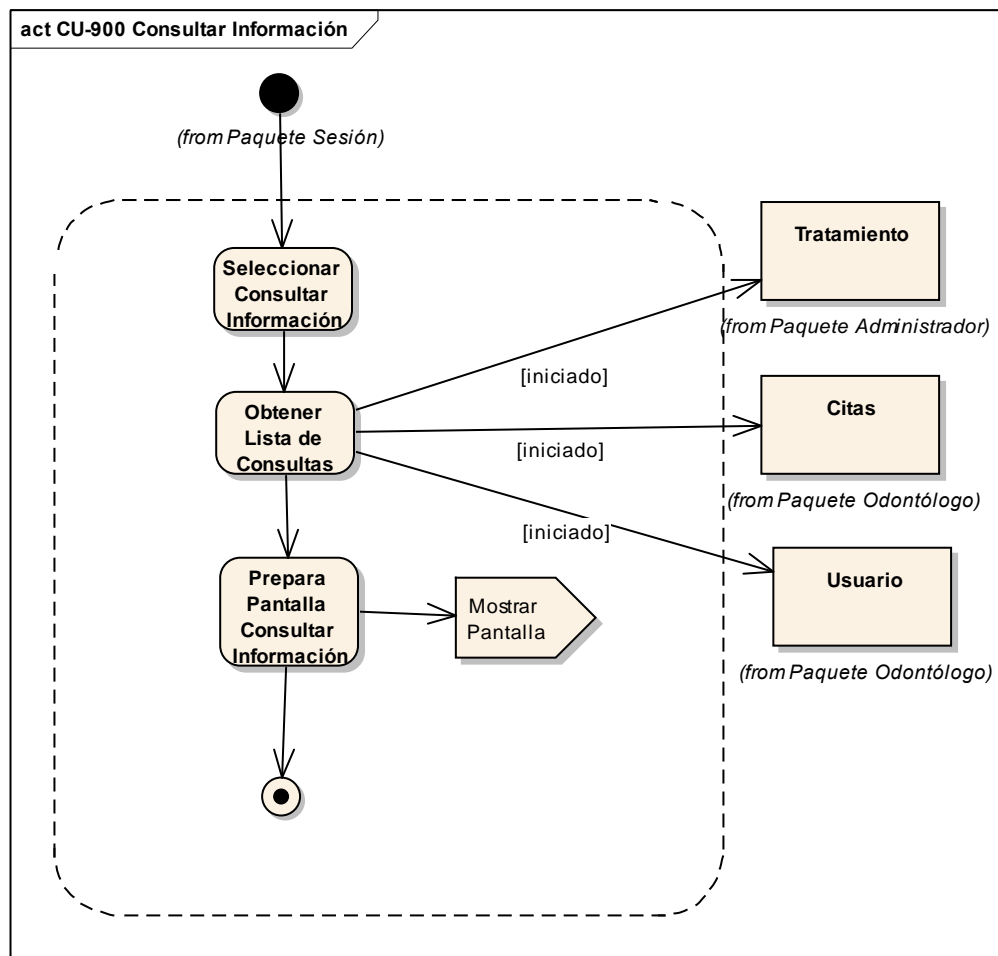


Figura 100 Diagrama de Actividad: CU-900 Consultar Información

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 cuánto 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

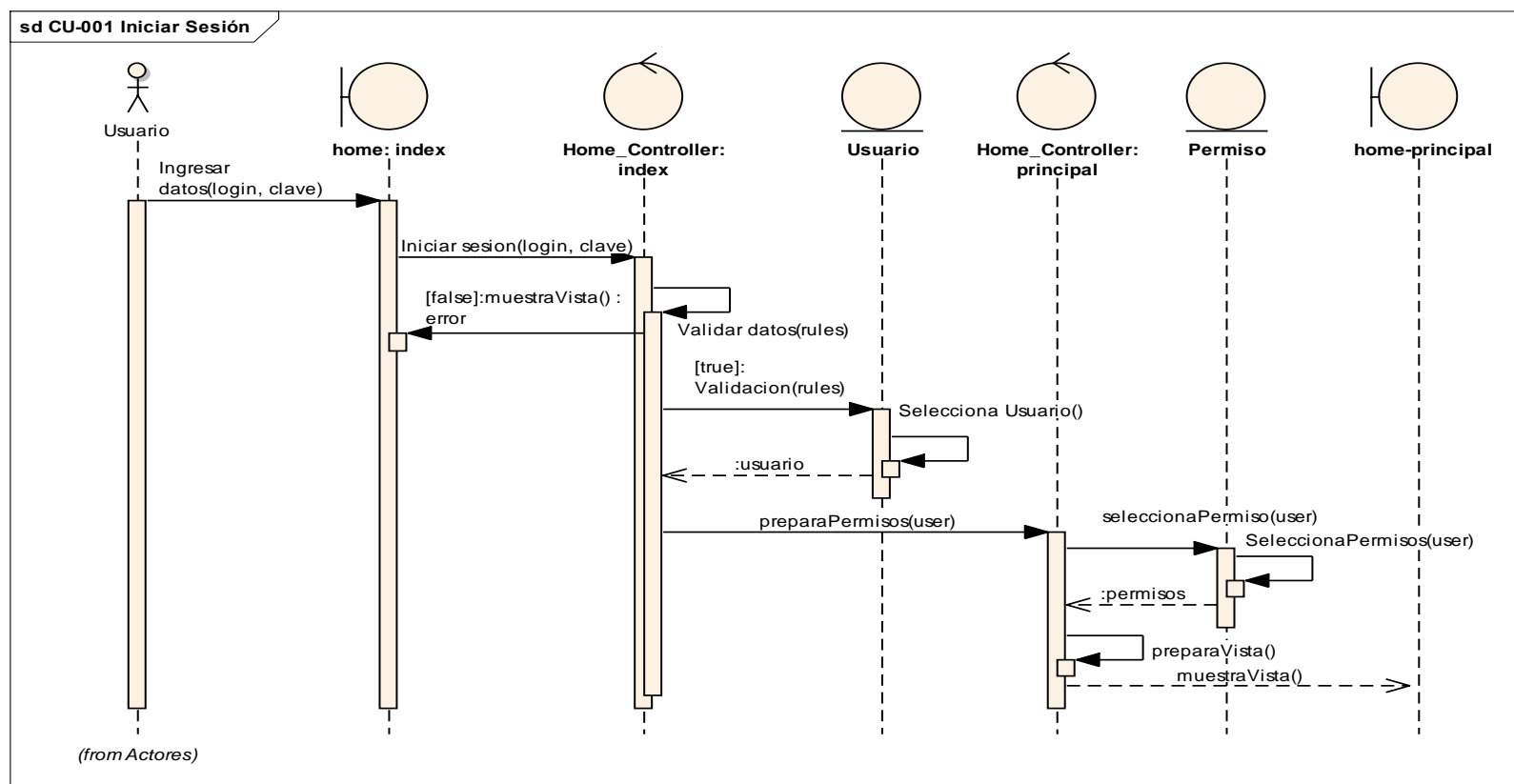


Figura 101 Diagrama de Secuencia: Caso de Uso Ingresar al Sistema

2.1.2.2.3.14.2.1.2 Diagrama de Secuencia: CU-100 Gestión Odontólogos

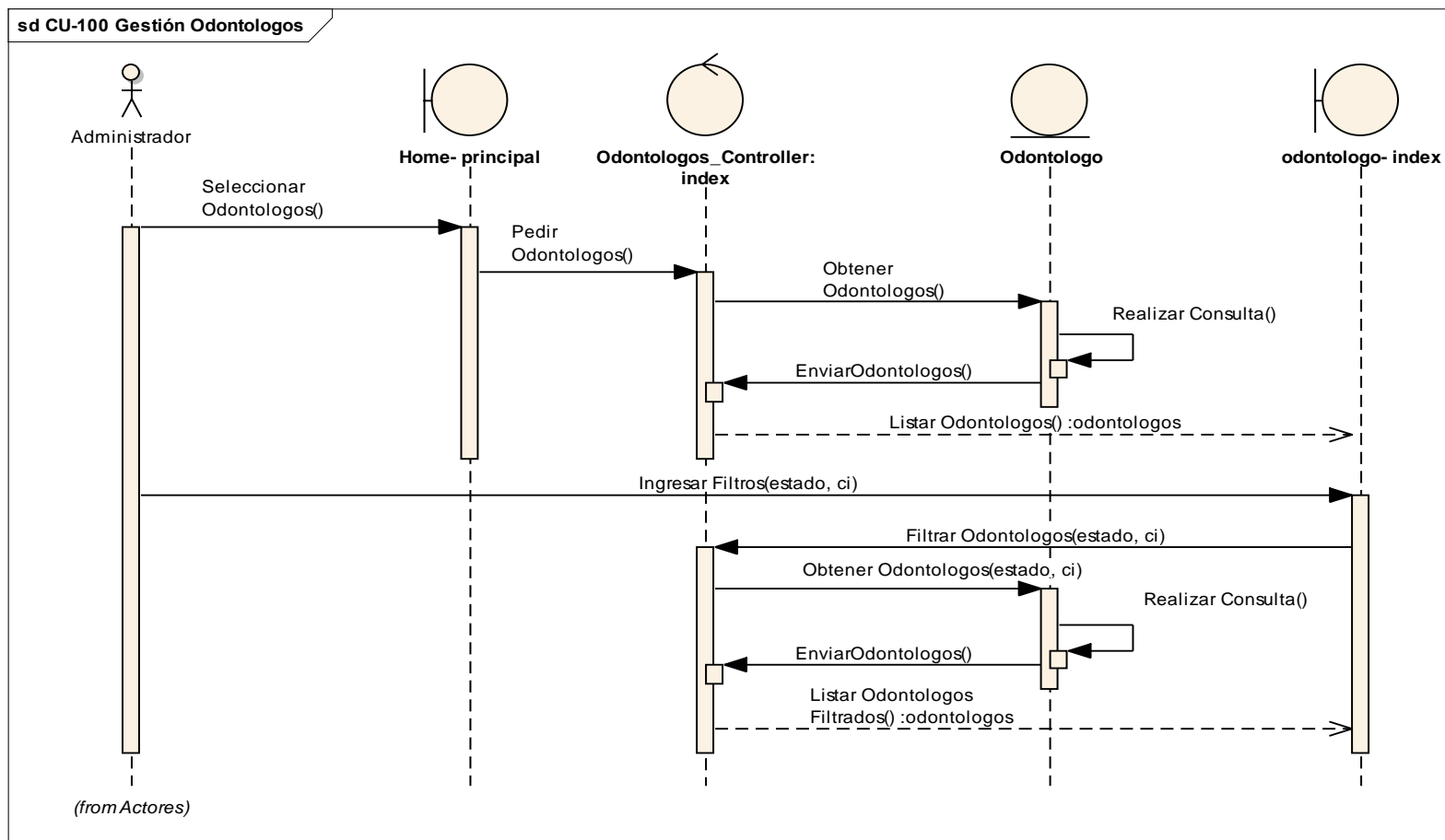


Figura 102 Diagrama de Secuencia: CU-100 Gestión Odontólogos

2.1.2.2.3.14.2.1.3 Diagrama de Secuencia: CU-101 Adicionar Odontólogo

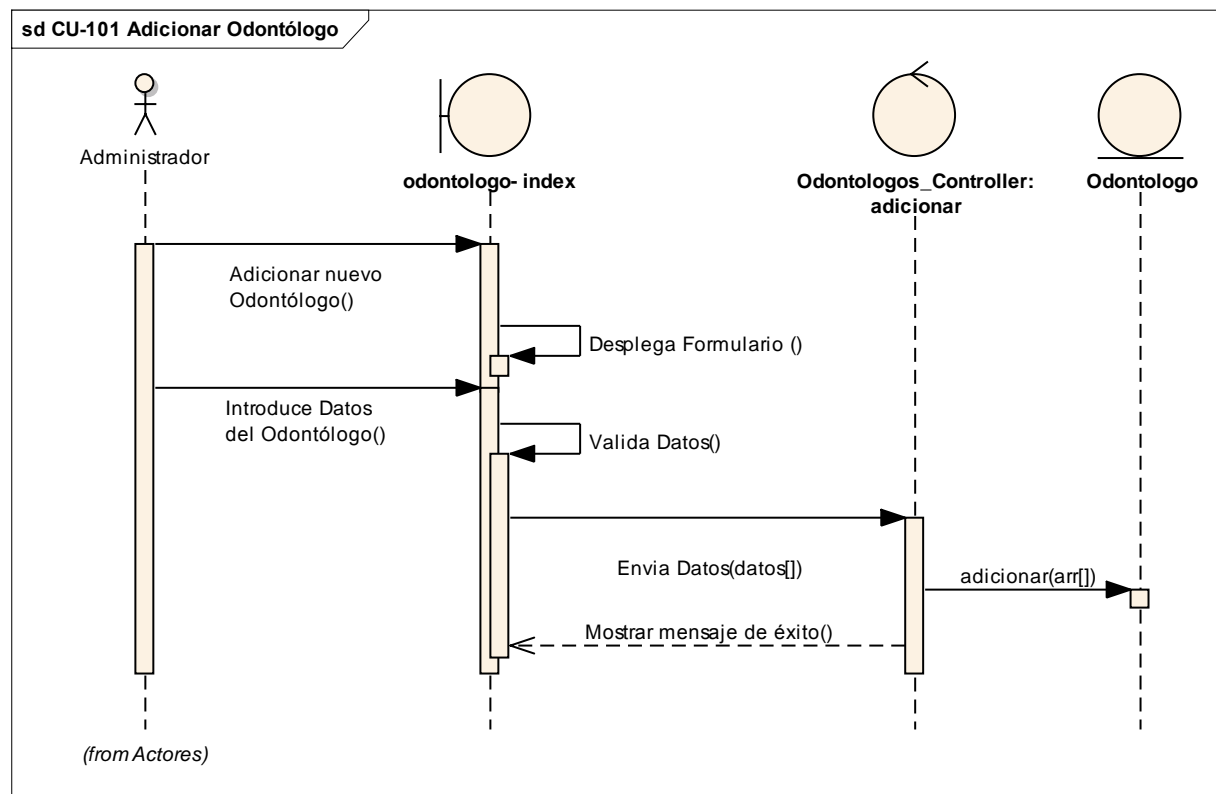


Figura 103 Diagrama de Secuencia: CU-101 Adicionar Odontólogo

2.1.2.2.3.14.2.1.4 Diagrama de Secuencia: CU-102 Modificar Odontólogo

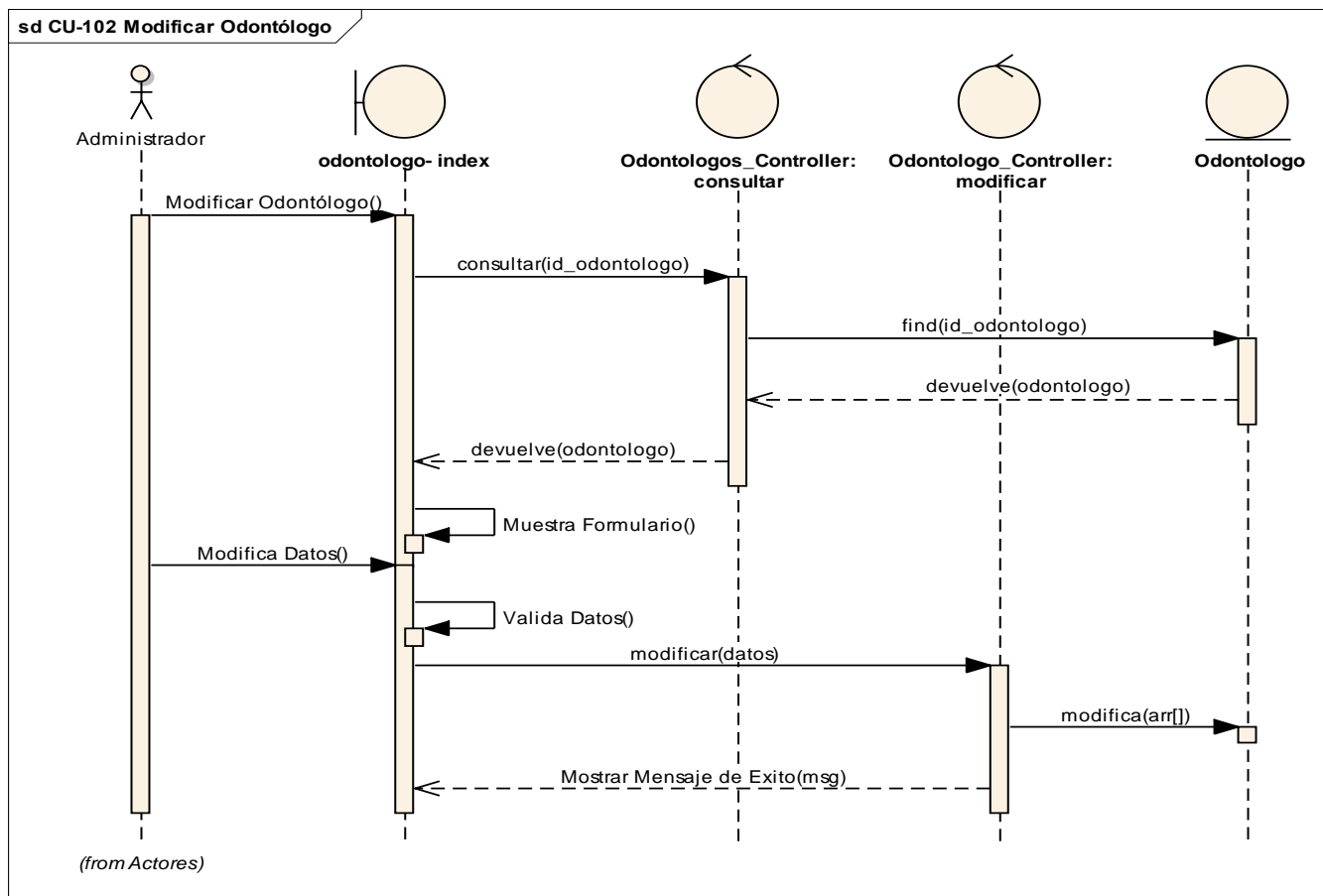


Figura 104 Diagrama de Secuencia: CU-102 Modificar Odontólogo

2.1.2.2.3.14.2.1.5 Diagrama de Secuencia: CU-103 Eliminar Odontólogo

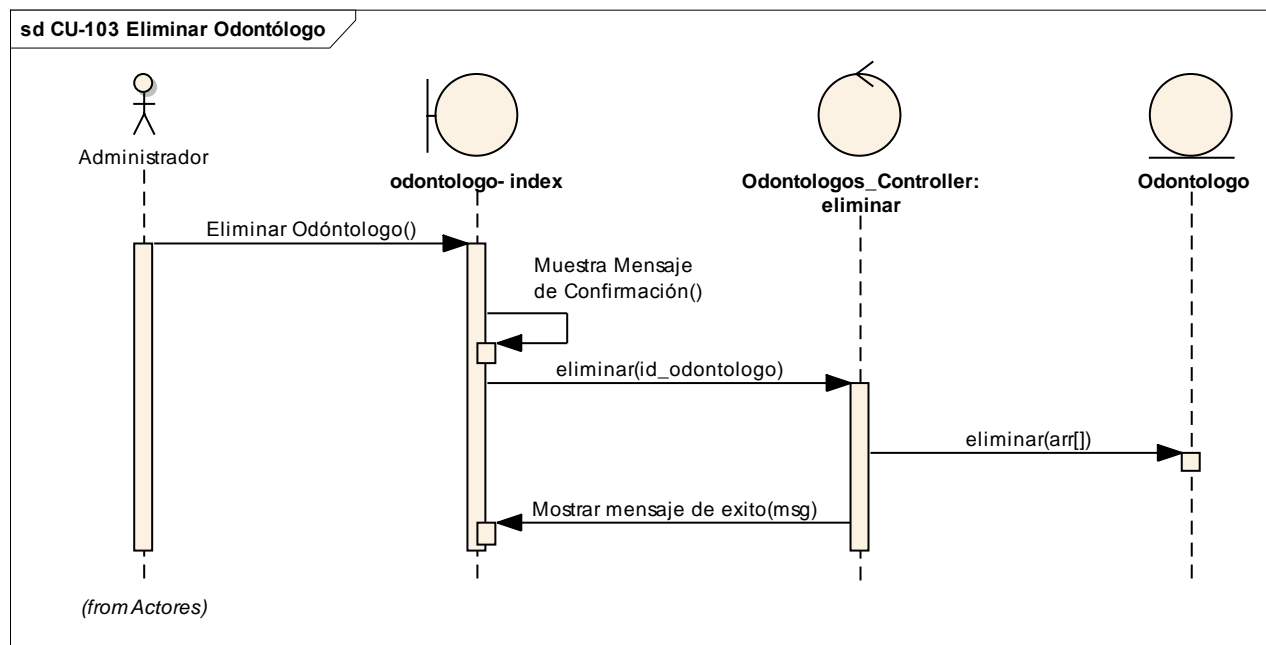


Figura 105 Diagrama de Secuencia: CU-103 Eliminar Odontólogo

2.1.2.2.3.14.2.1.6 Diagrama de Secuencia: CU-104 Asignar/Modificar Usuario

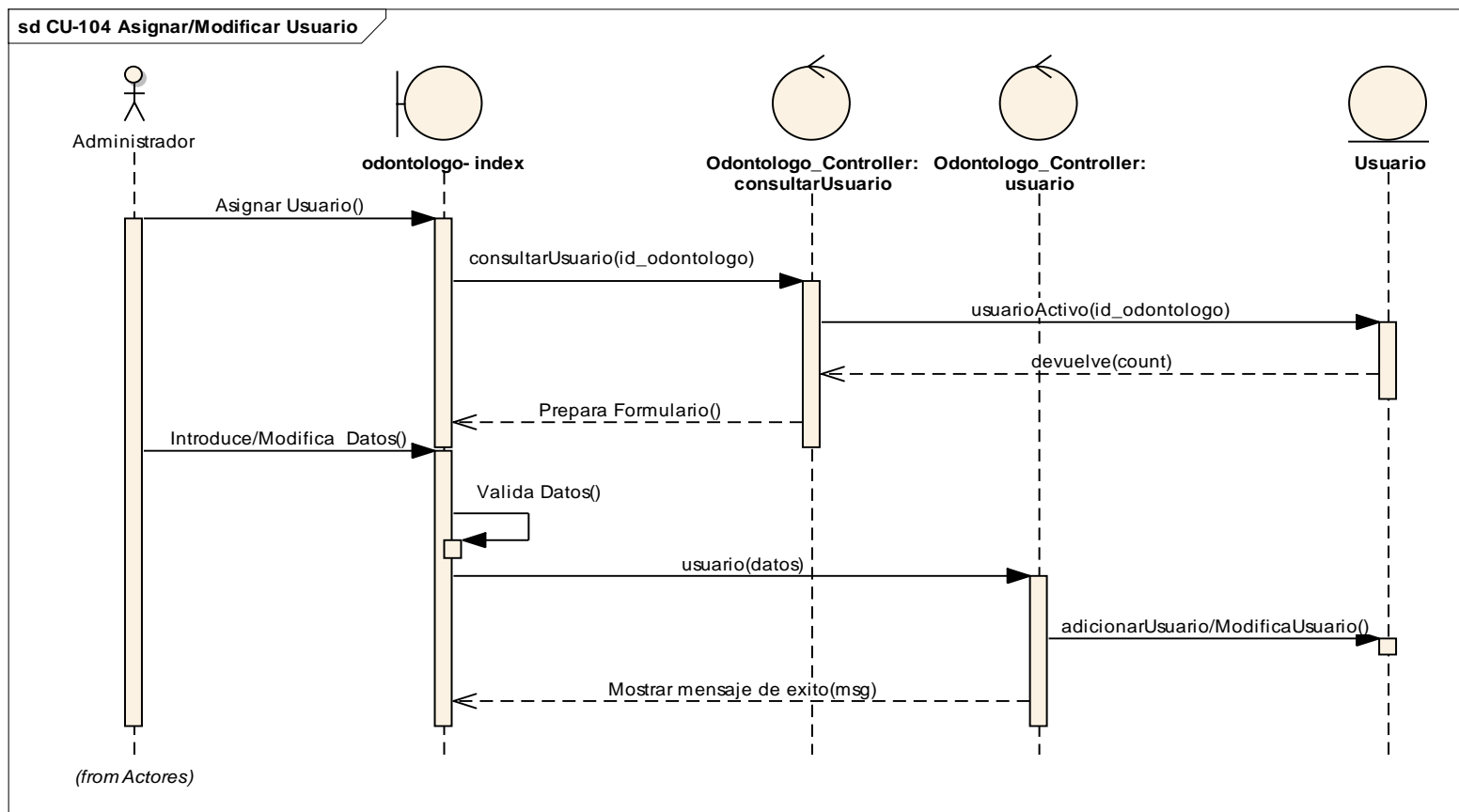


Figura 106 Diagrama de Secuencia: CU-104 Asignar/Modificar Usuario

2.1.2.2.3.14.2.1.7 Diagrama de Secuencia: CU-200 Gestión Permisos

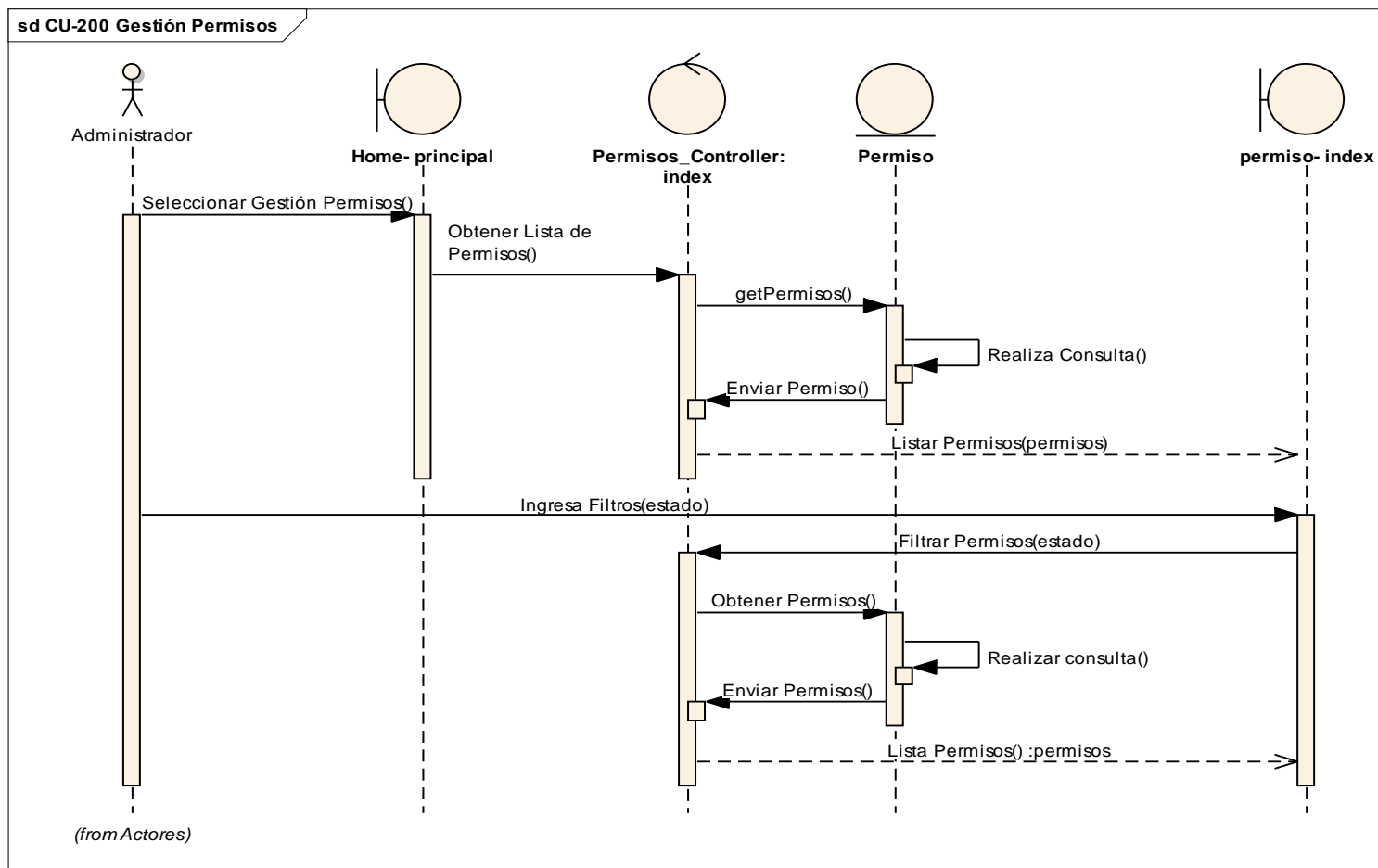


Figura 107 Diagrama de Secuencia: CU-200 Gestión Permisos

2.1.2.2.3.14.2.1.8 Diagrama de Secuencia: CU-201 Adicionar Permiso

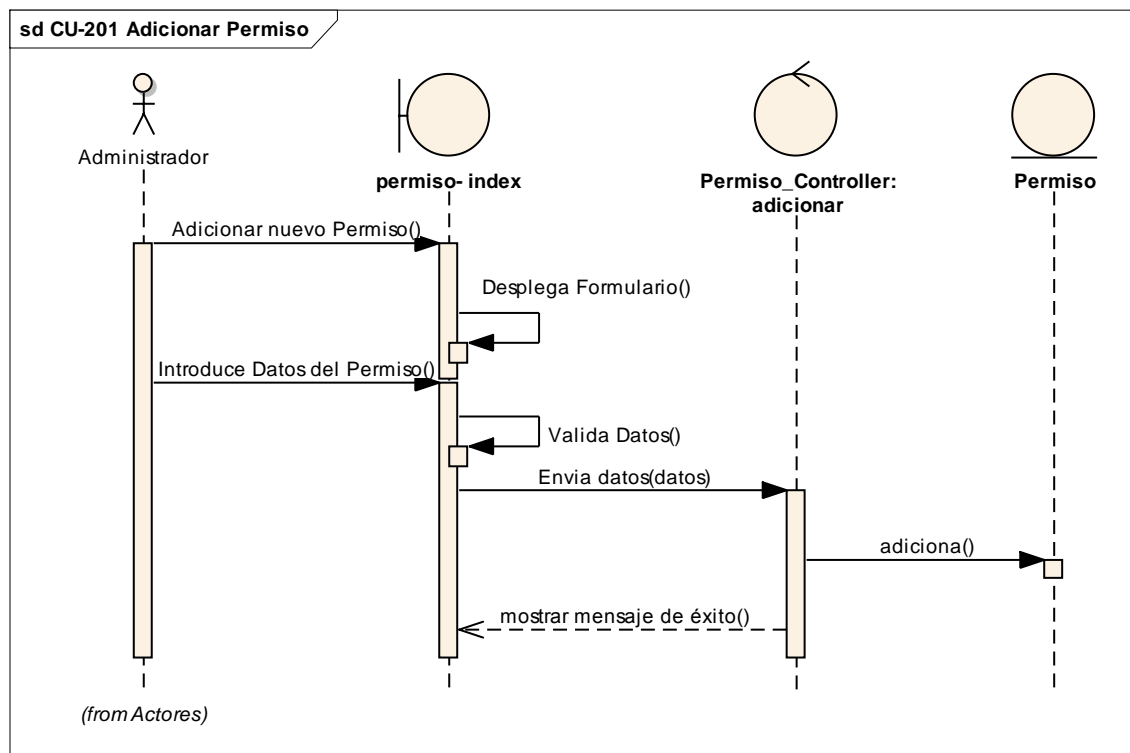


Figura 108 Diagrama de Secuencia: CU-201 Adicionar Permiso

2.1.2.2.3.14.2.1.9 Diagrama de Secuencia: CU-202 Modificar Permiso

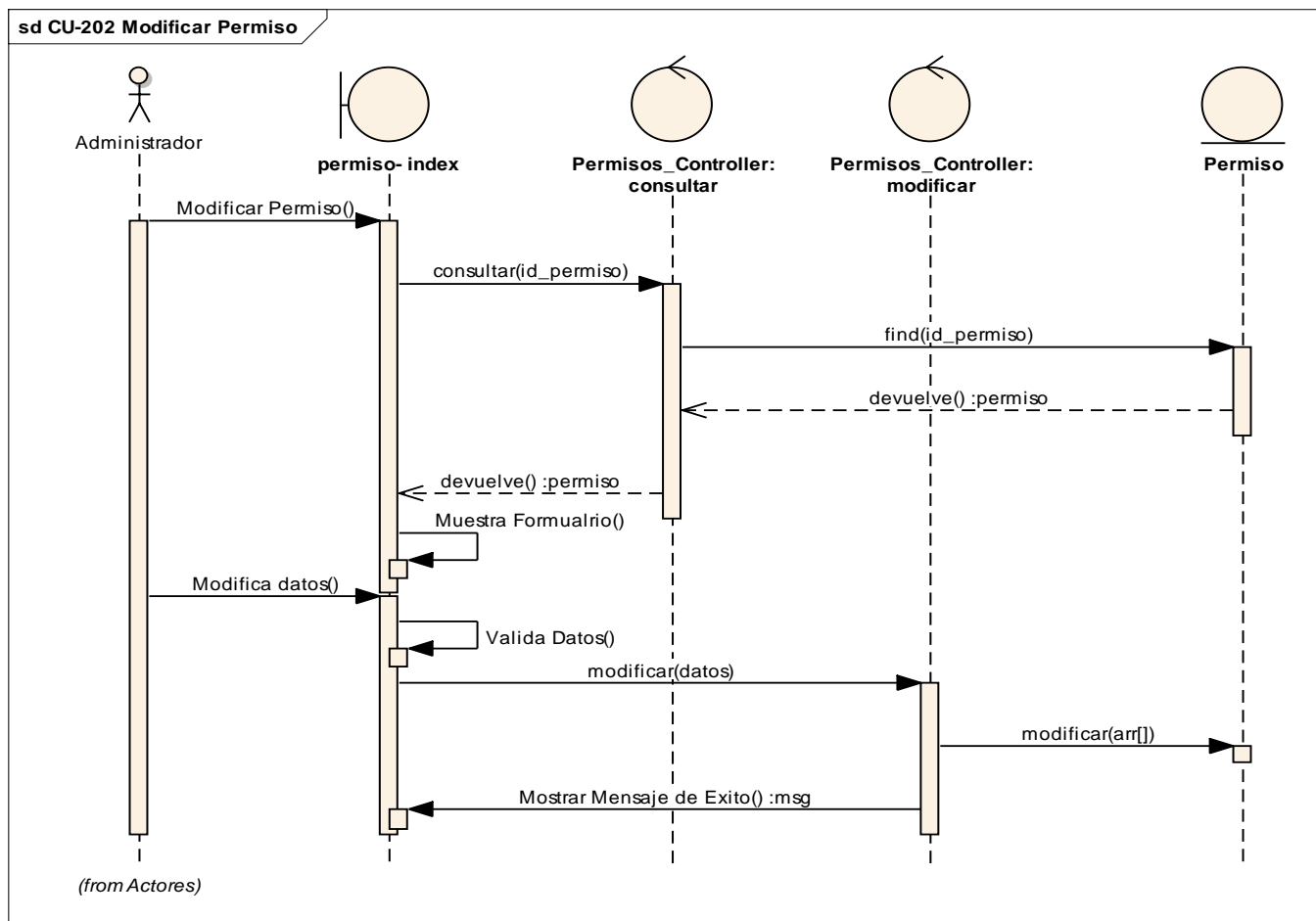


Figura 109 Diagrama de Secuencia: CU-202 Modificar Permiso

2.1.2.2.3.14.2.1.10 Diagrama de Secuencia: CU-203 Eliminar Permiso

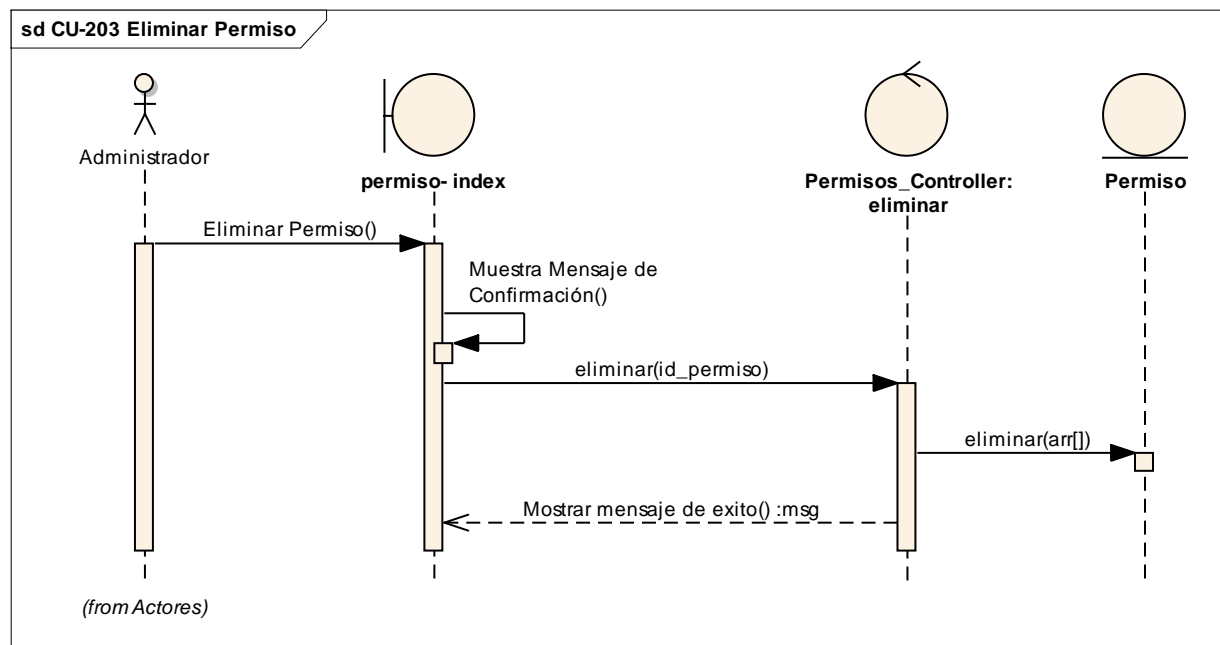


Figura 110 Diagrama de Secuencia: CU-203 Eliminar Permiso

2.1.2.2.3.14.2.1.11 Diagrama de Secuencia: CU-300 Gestión Tratamientos

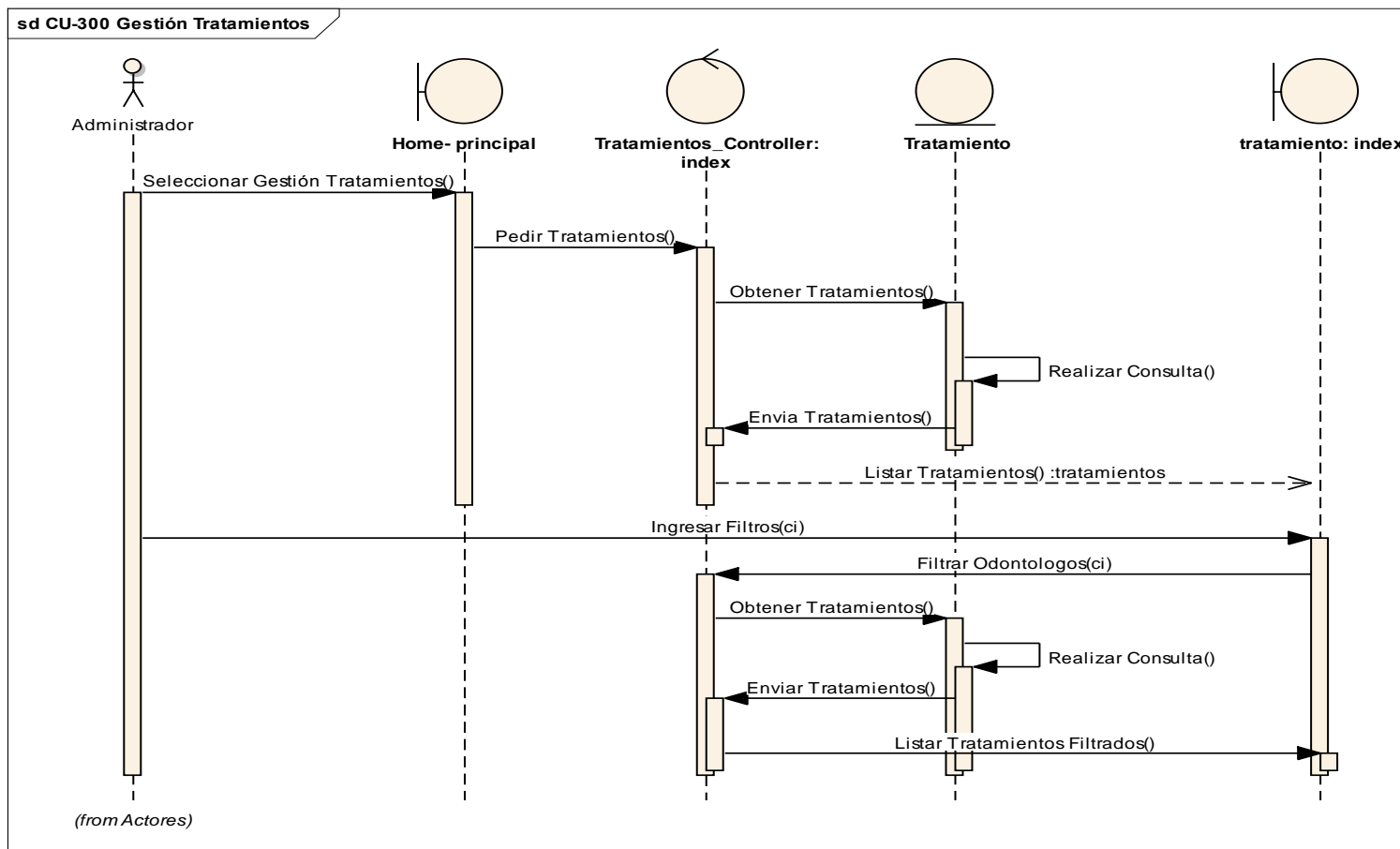


Figura 111 Diagrama de Secuencia: CU-300 Gestión Tratamientos

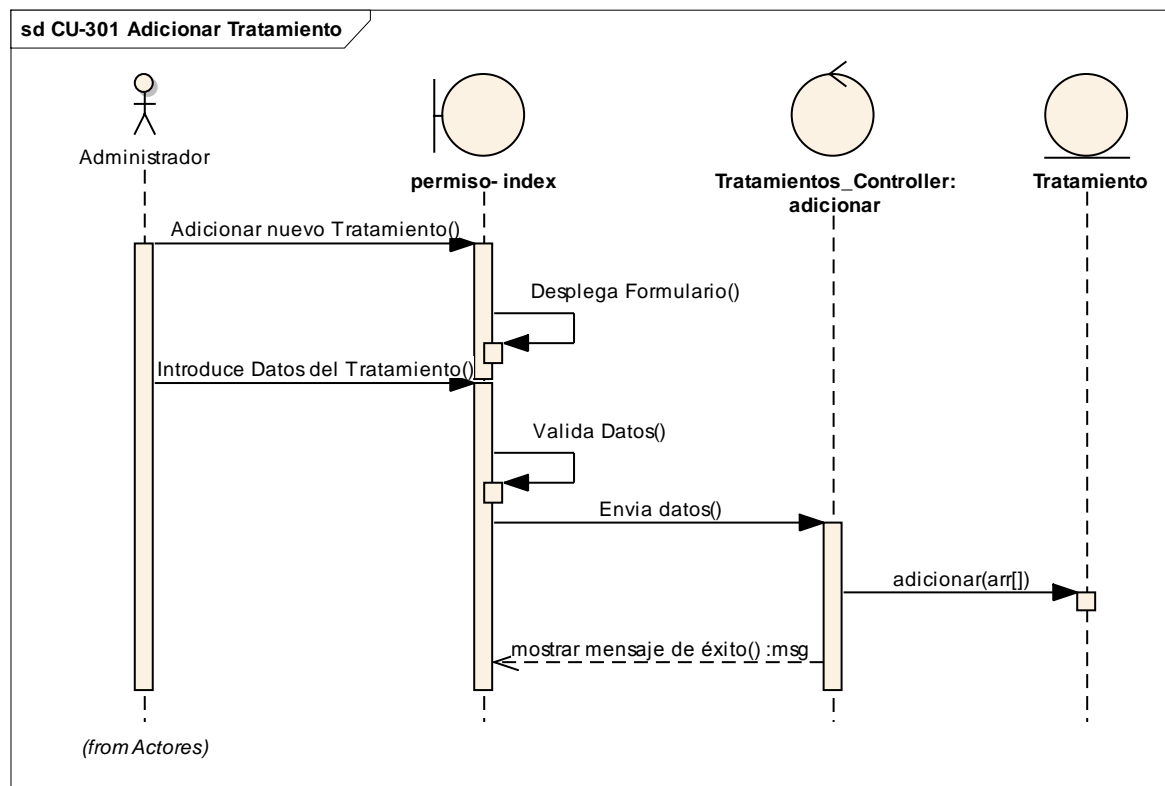
2.1.2.2.3.14.2.1.12 Diagrama de Secuencia: CU-301 Adicionar Tratamiento

Figura 112 Diagrama de Secuencia: CU-301 Adicionar Tratamiento

2.1.2.2.3.14.2.1.13 Diagrama de Secuencia: CU-302 Modificar Tratamiento

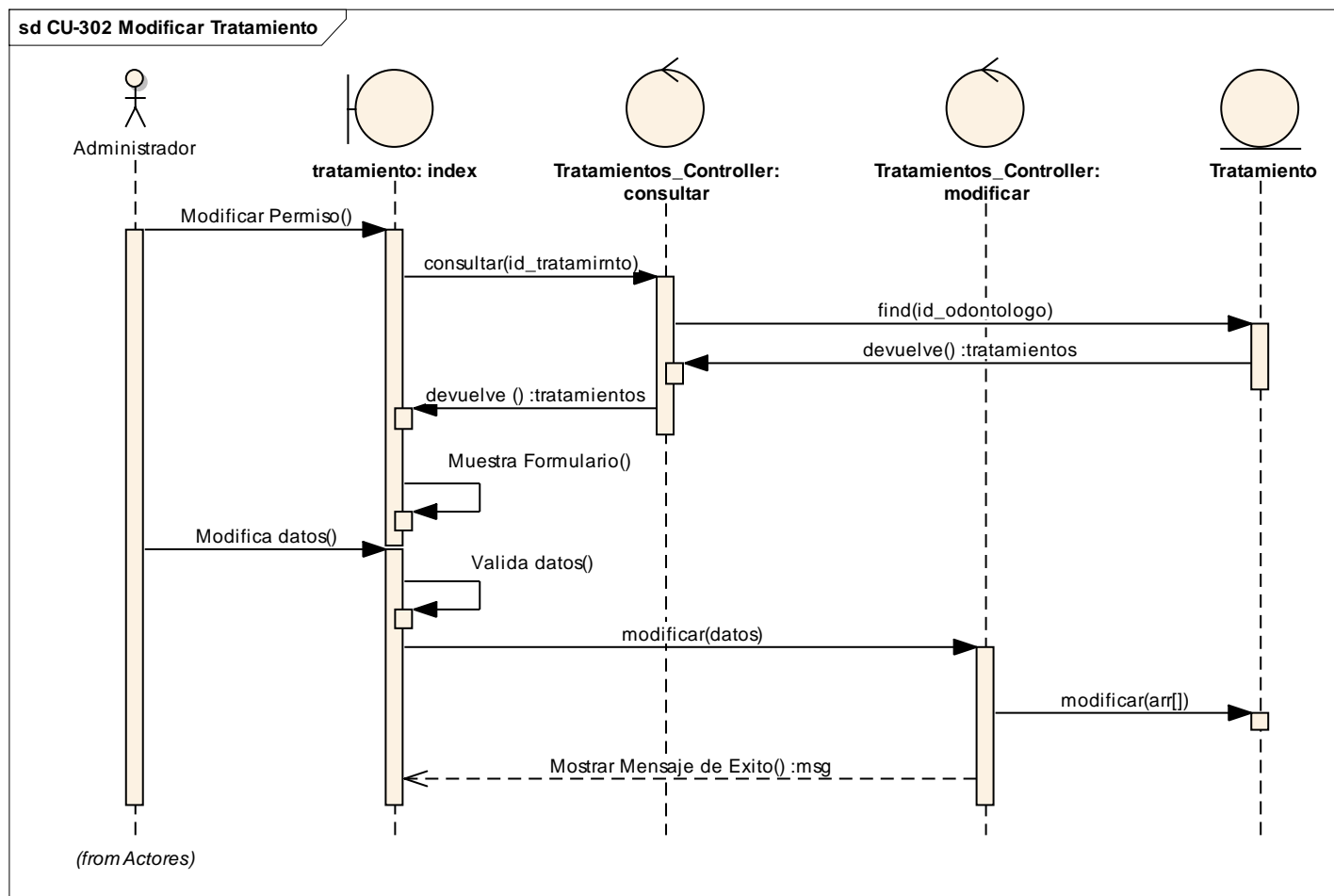


Figura 113 Diagrama de Secuencia: CU-302 Modificar Tratamiento

2.1.2.2.3.14.2.1.14 Diagrama de Secuencia: CU-303 Eliminar Tratamiento

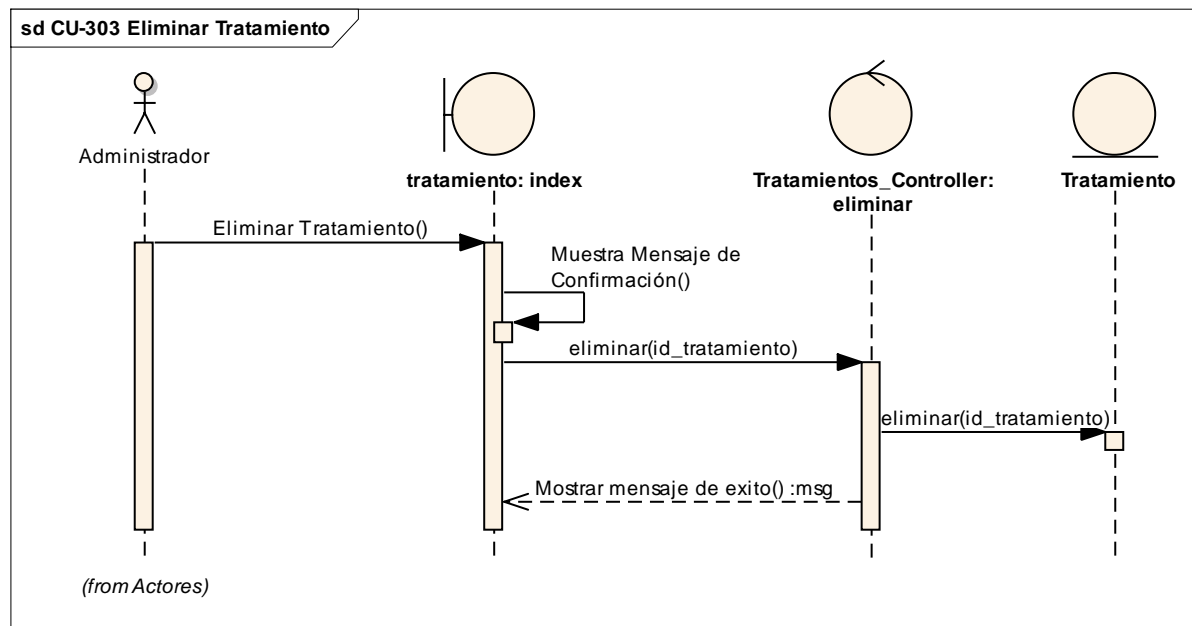


Figura 114 Diagrama de Secuencia: Caso de Uso Adicionar Ingreso

2.1.2.2.3.14.2.1.15 Diagrama de Secuencia: CU-400 Gestión Áreas

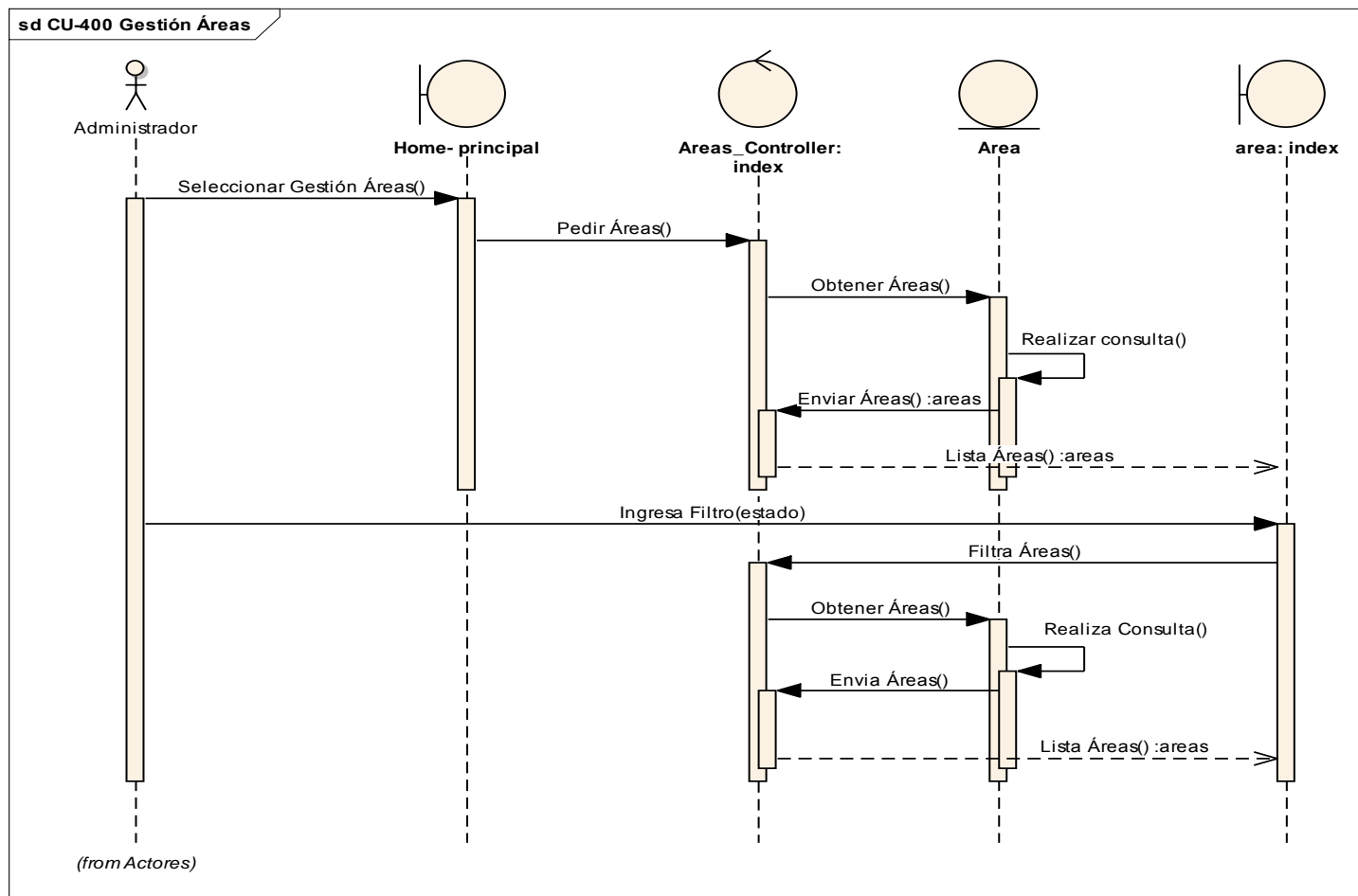


Figura 115 Diagrama de Secuencia: CU-400 Gestión Áreas

2.1.2.2.3.14.2.1.16 Diagrama de Secuencia: CU-401 Adicionar Área

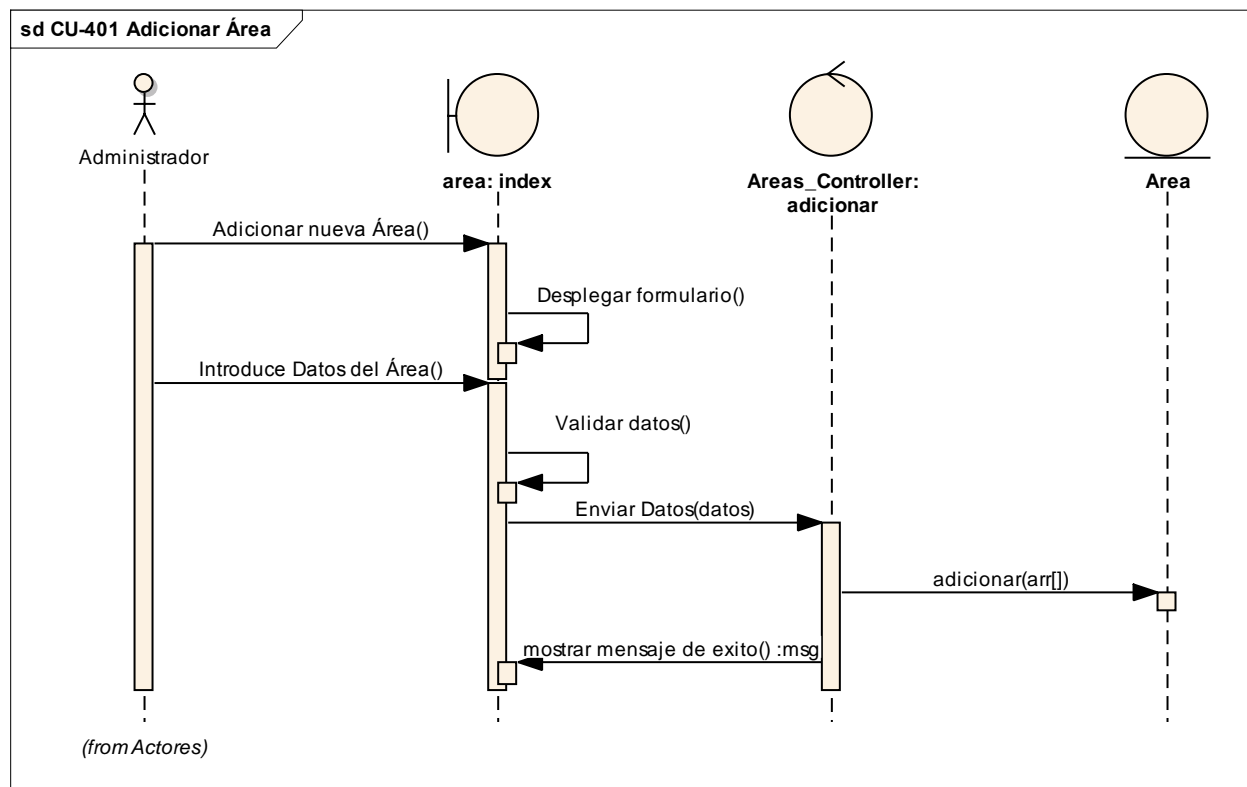


Figura 116 Diagrama de Secuencia: CU-401 Adicionar Área

2.1.2.2.3.14.2.1.17 Diagrama de Secuencia: CU-402 Modificar Área

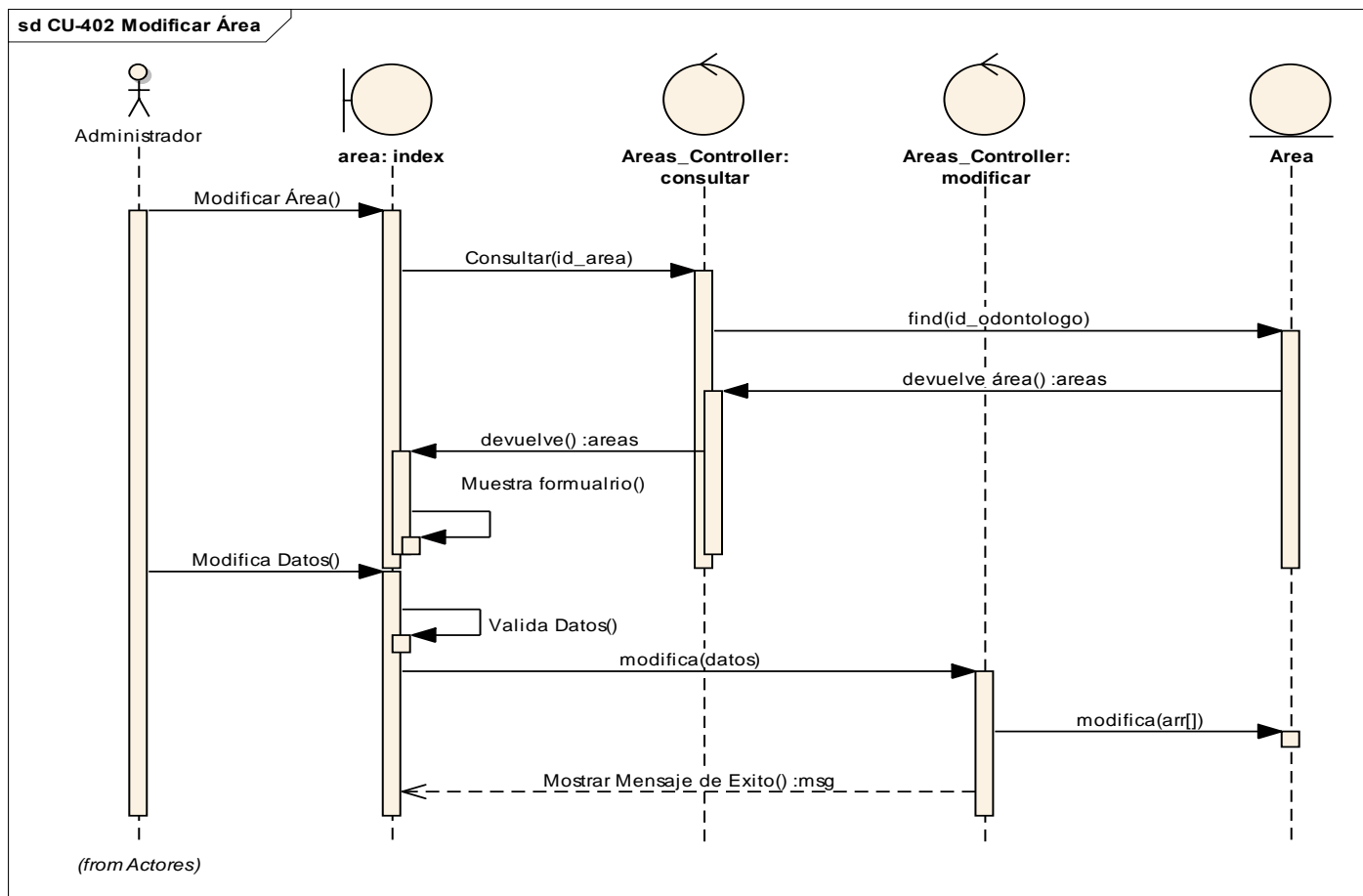


Figura 117 Diagrama de Secuencia: CU-402 Modificar Área

2.1.2.2.3.14.2.1.18 Diagrama de Secuencia: CU-403 Eliminar Área

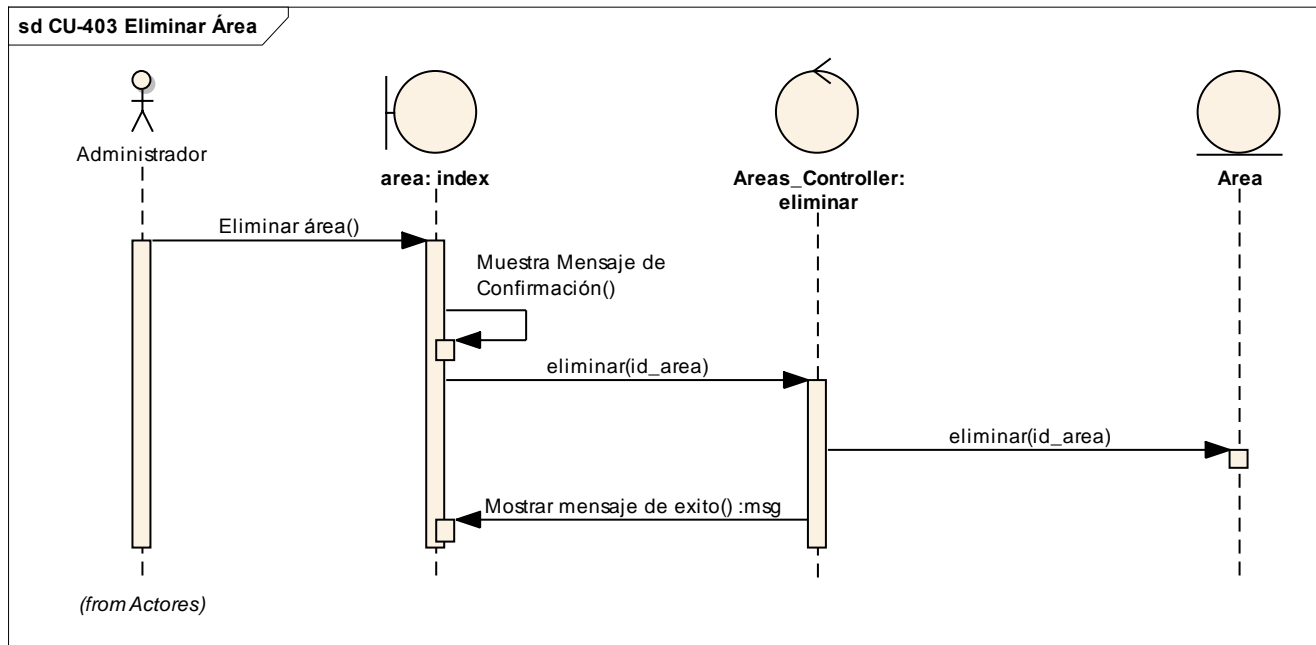


Figura 118 Diagrama de Secuencia: CU-403 Eliminar Área

2.1.2.2.3.14.2.1.19 Diagrama de Secuencia: CU-500 Gestión Pacientes

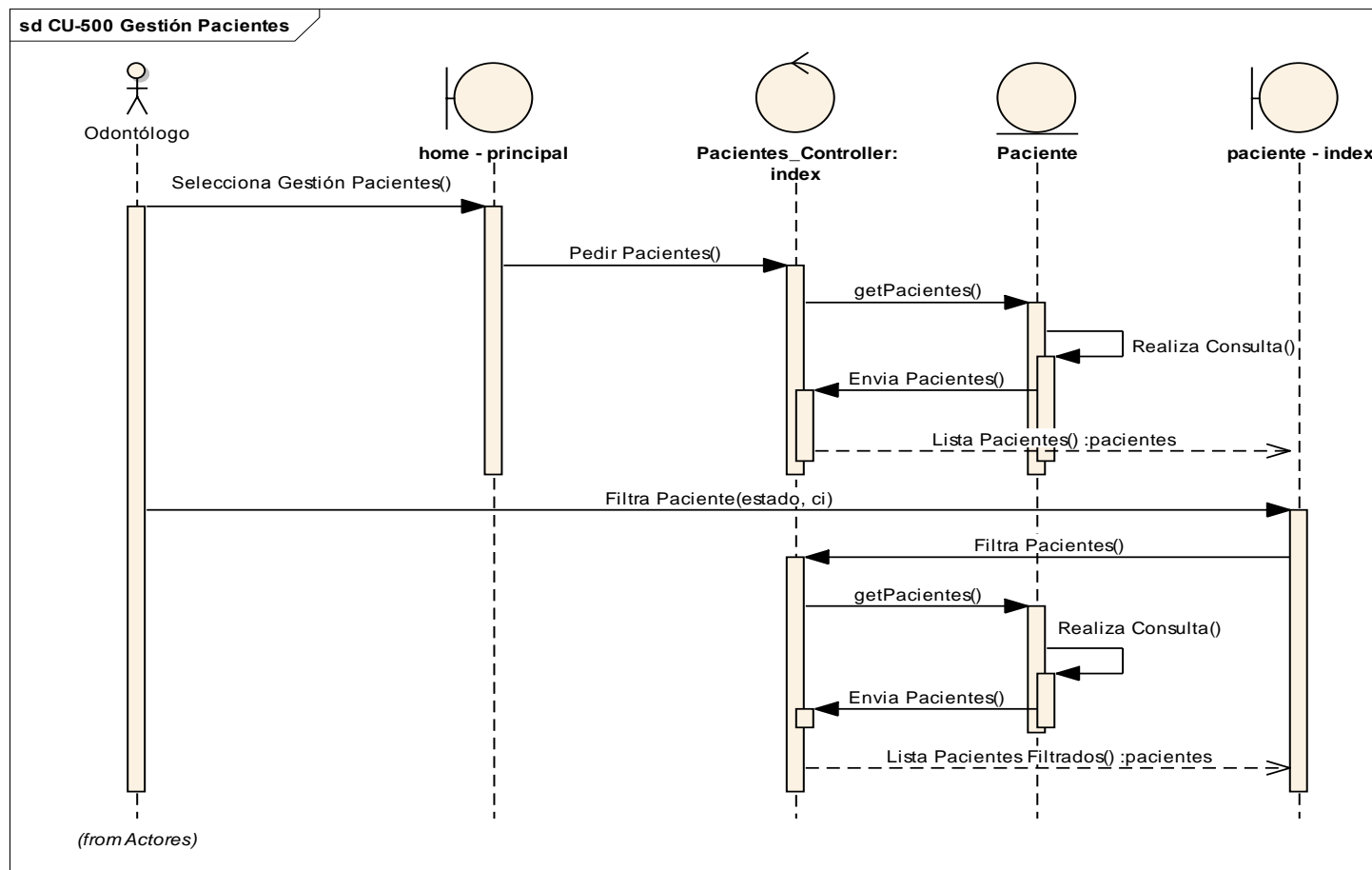


Figura 119 Diagrama de Secuencia: CU-500 Gestión Pacientes

2.1.2.2.3.14.2.1.20 Diagrama de Secuencia: CU-501 Registro Paciente

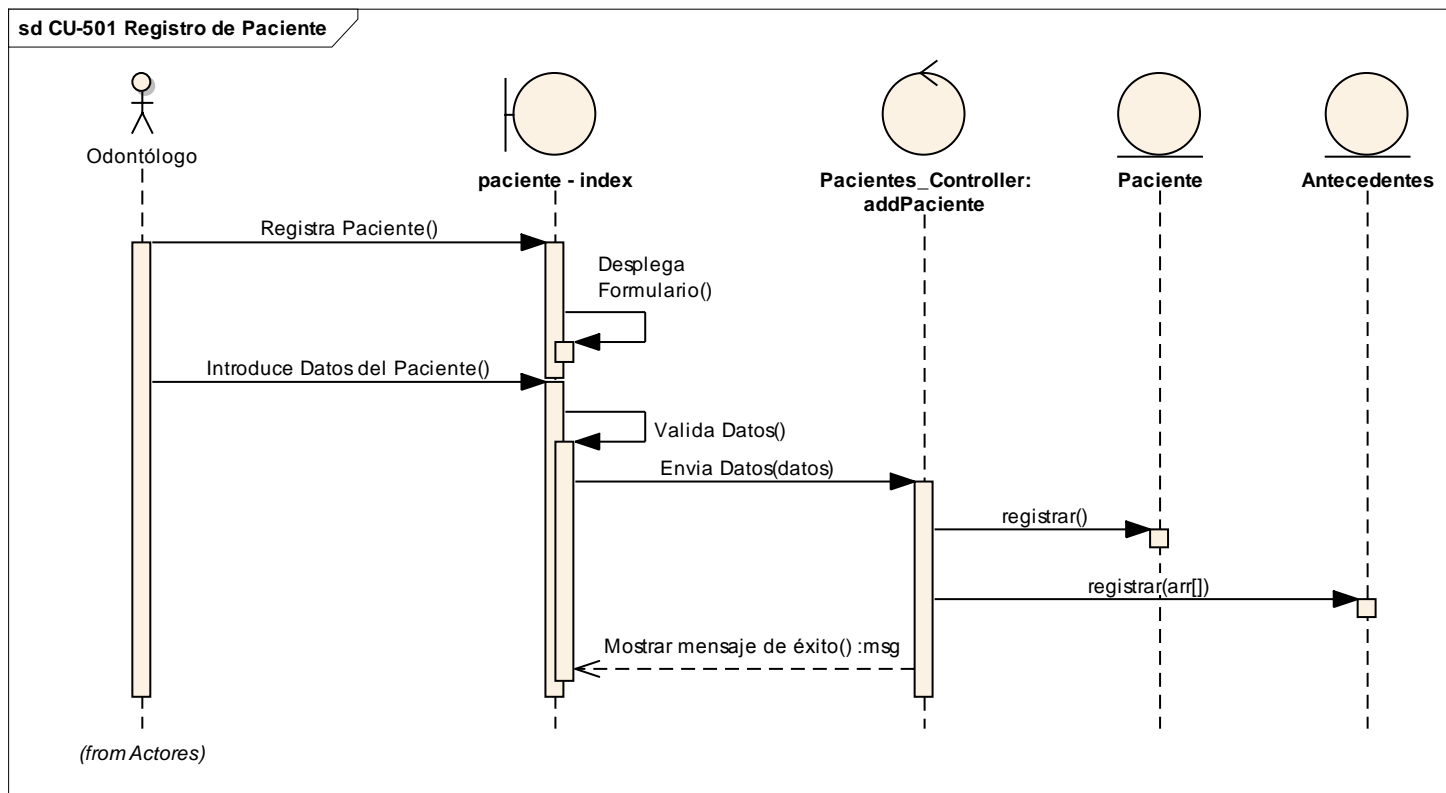


Figura 120 Diagrama de Secuencia: CU-501 Registro Paciente

2.1.2.2.3.14.2.1.21 Diagrama de Secuencia: CU-502 Modificar Registro

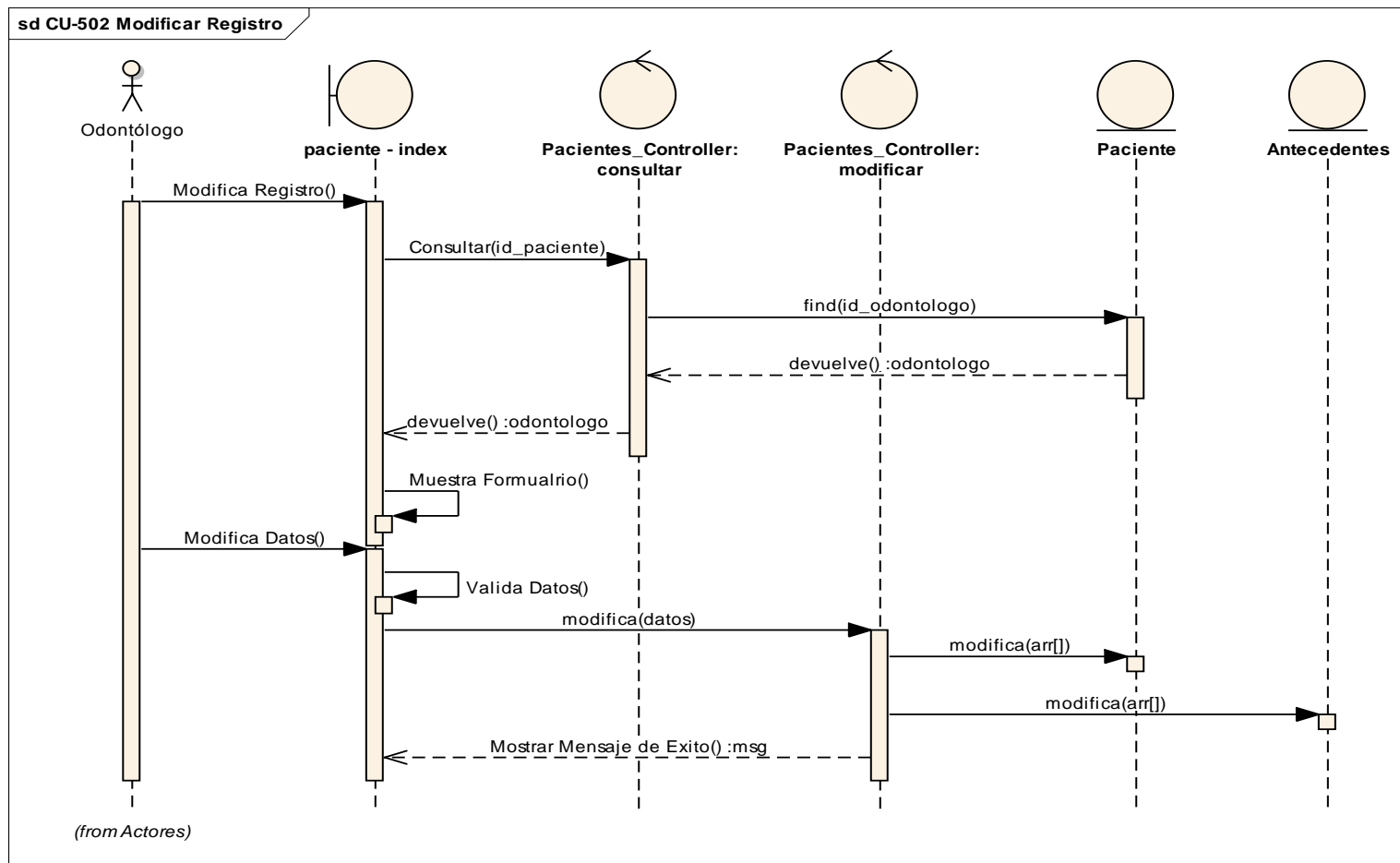


Figura 121 Diagrama de Secuencia: CU-502 Modificar Registro

2.1.2.2.3.14.2.1.22 Diagrama de Secuencia: CU-600 Asignación de Citas

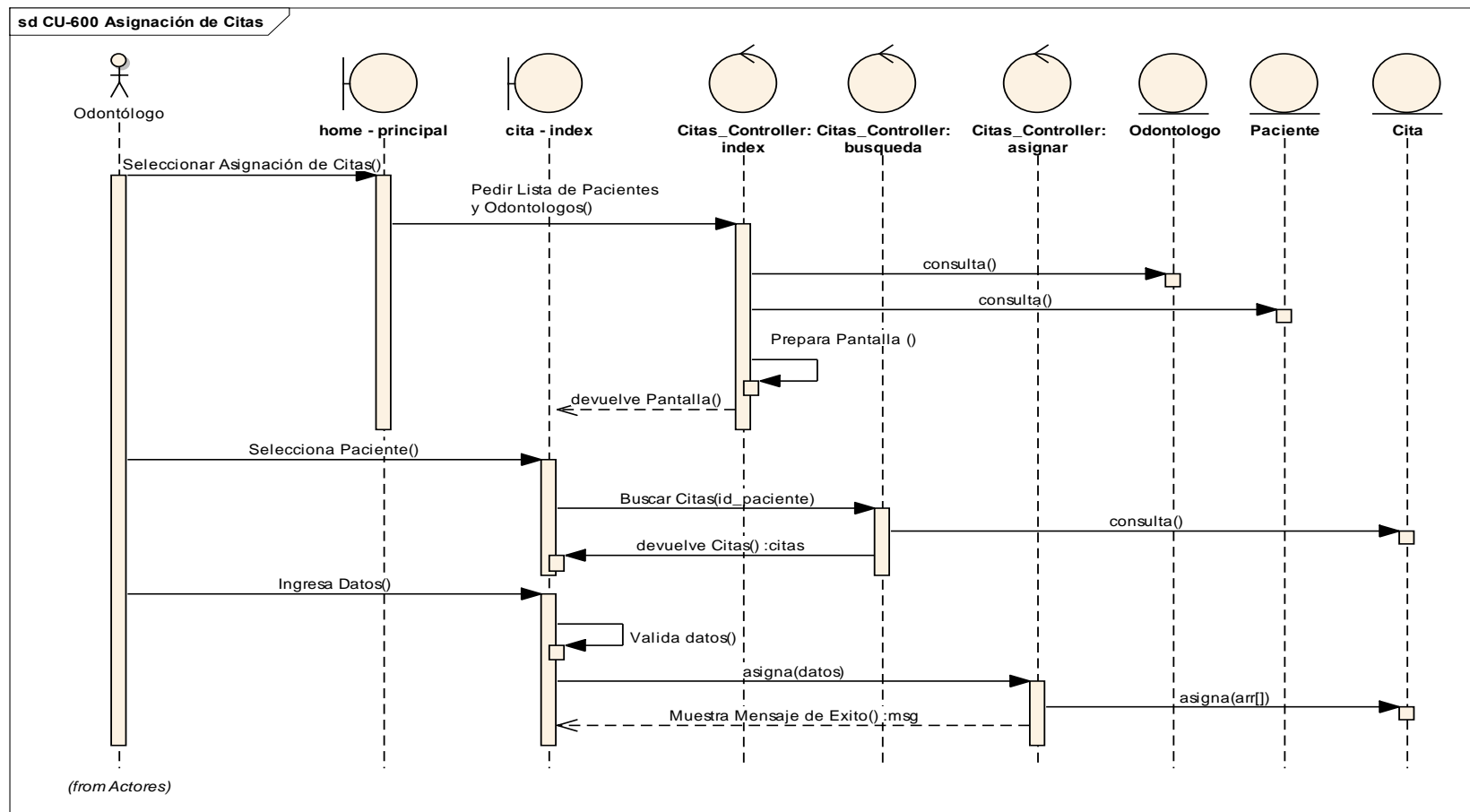


Figura 122 Diagrama de Secuencia: CU-600 Asignación de Citas

2.1.2.2.3.14.2.1.23 Diagrama de Secuencia: CU-700 Gestión Examen Clínico

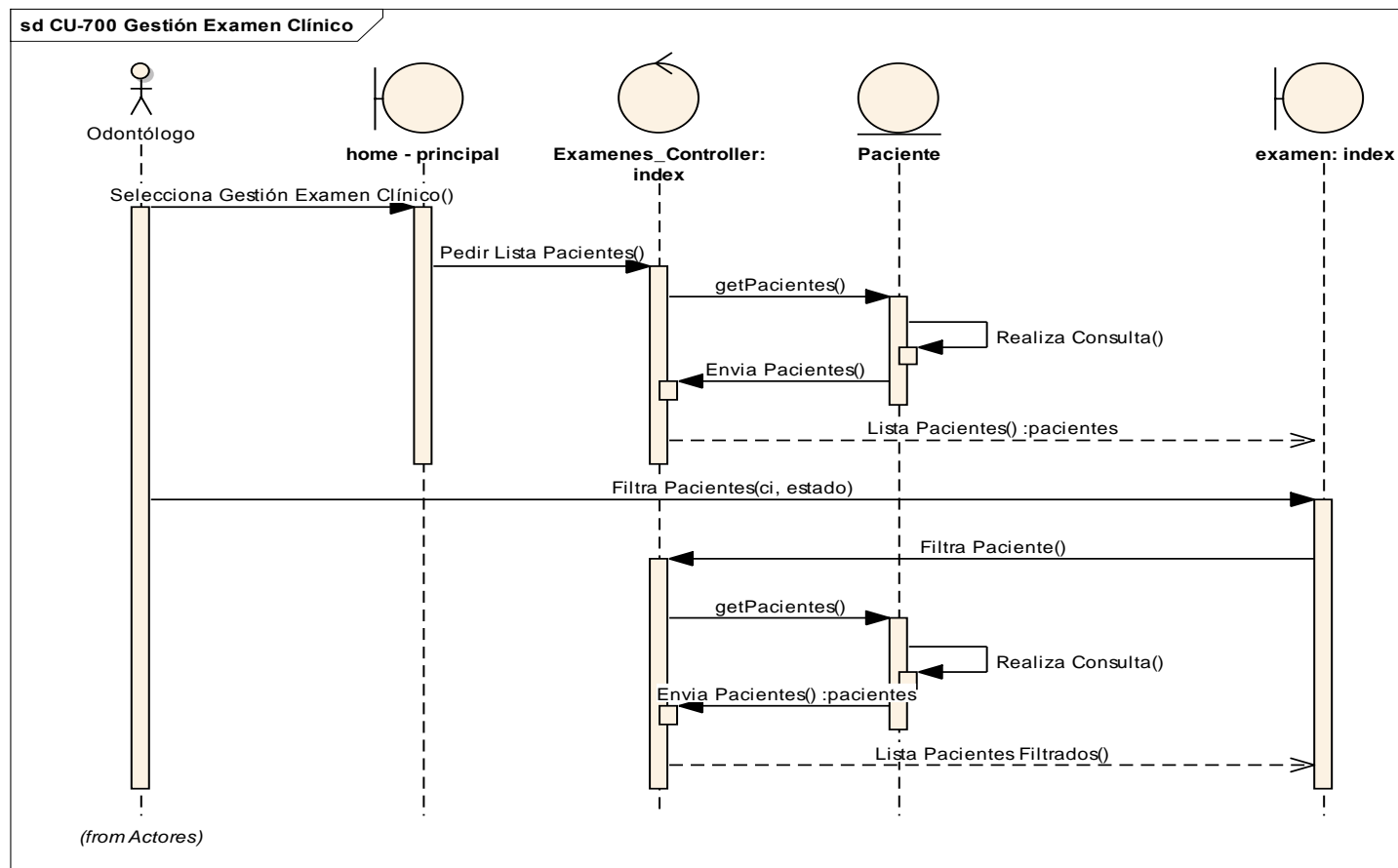


Figura 123 Diagrama de Secuencia: CU-700 Gestión Examen Clínico

2.1.2.2.3.14.2.1.24 Diagrama de Secuencia: CU-701 Seguimiento del Tratamiento

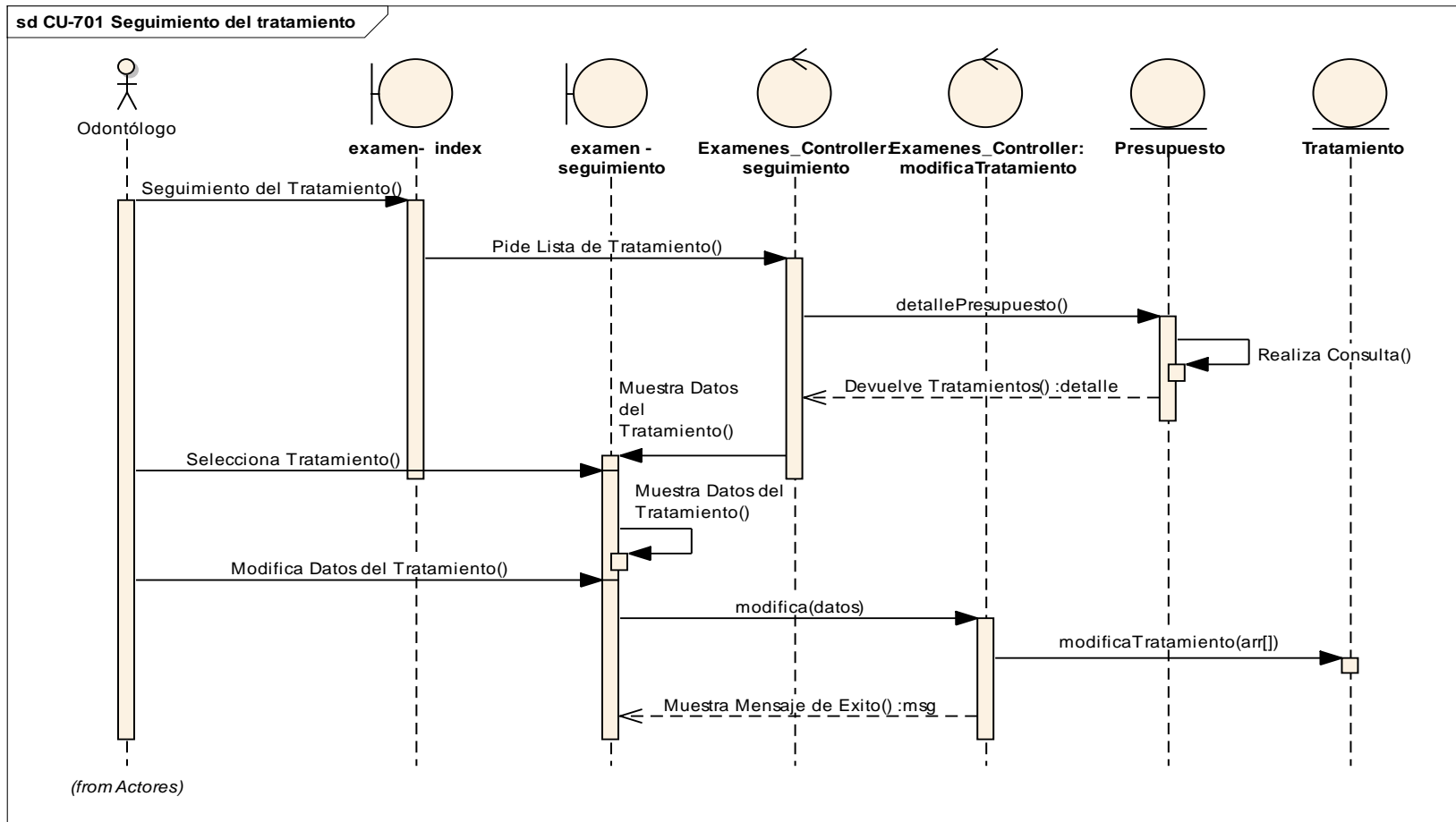


Figura 124 Diagrama de Secuencia: CU-701 Seguimiento del Tratamiento

2.1.2.2.3.14.2.1.25 Diagrama de Secuencia: CU-800 Gestión Presupuesto

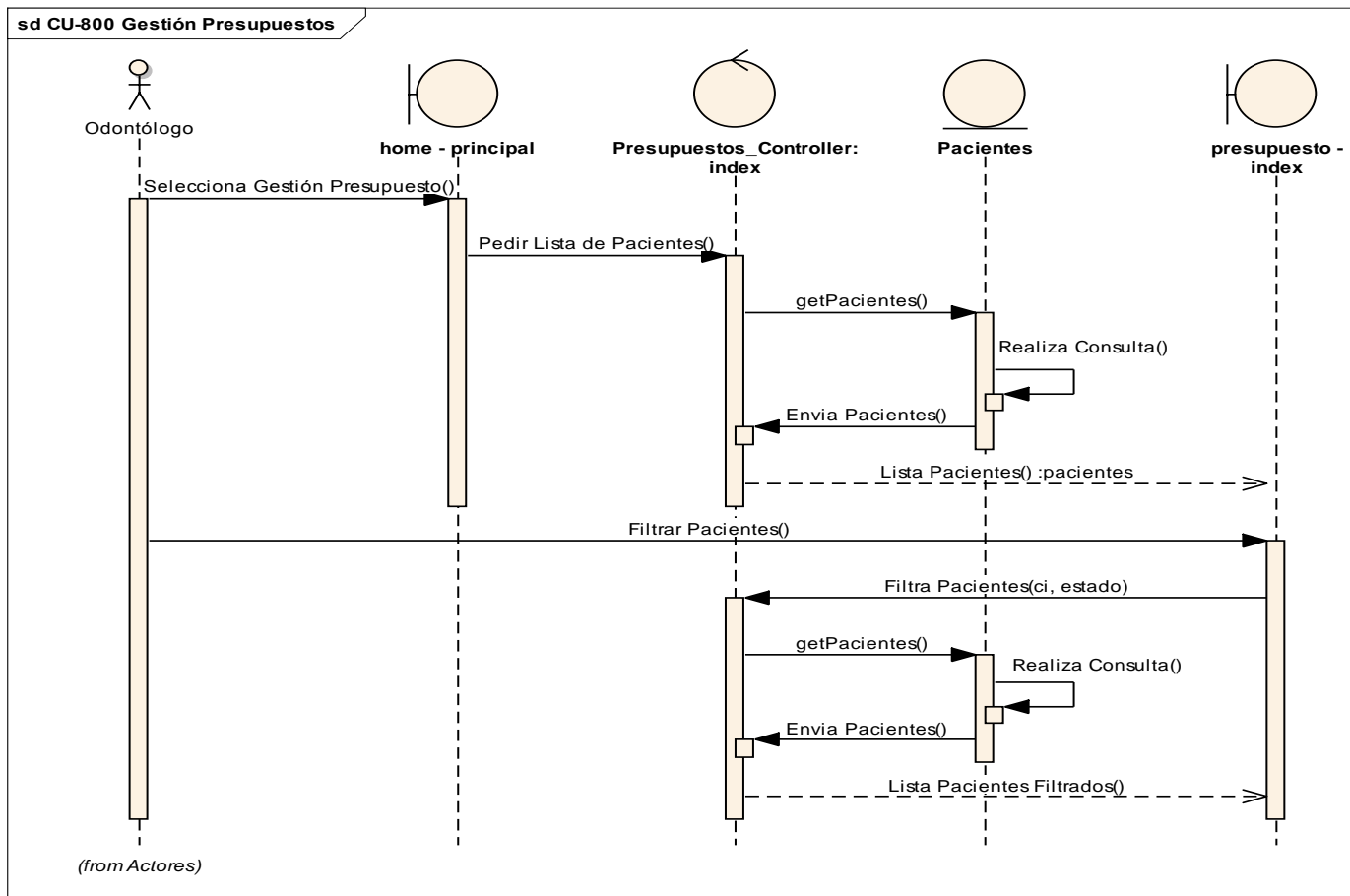


Figura 125 Diagrama de Secuencia: CU-800 Gestión Presupuesto

2.1.2.2.3.14.2.1.26 Diagrama de Secuencia: CU-900 Consultar Información

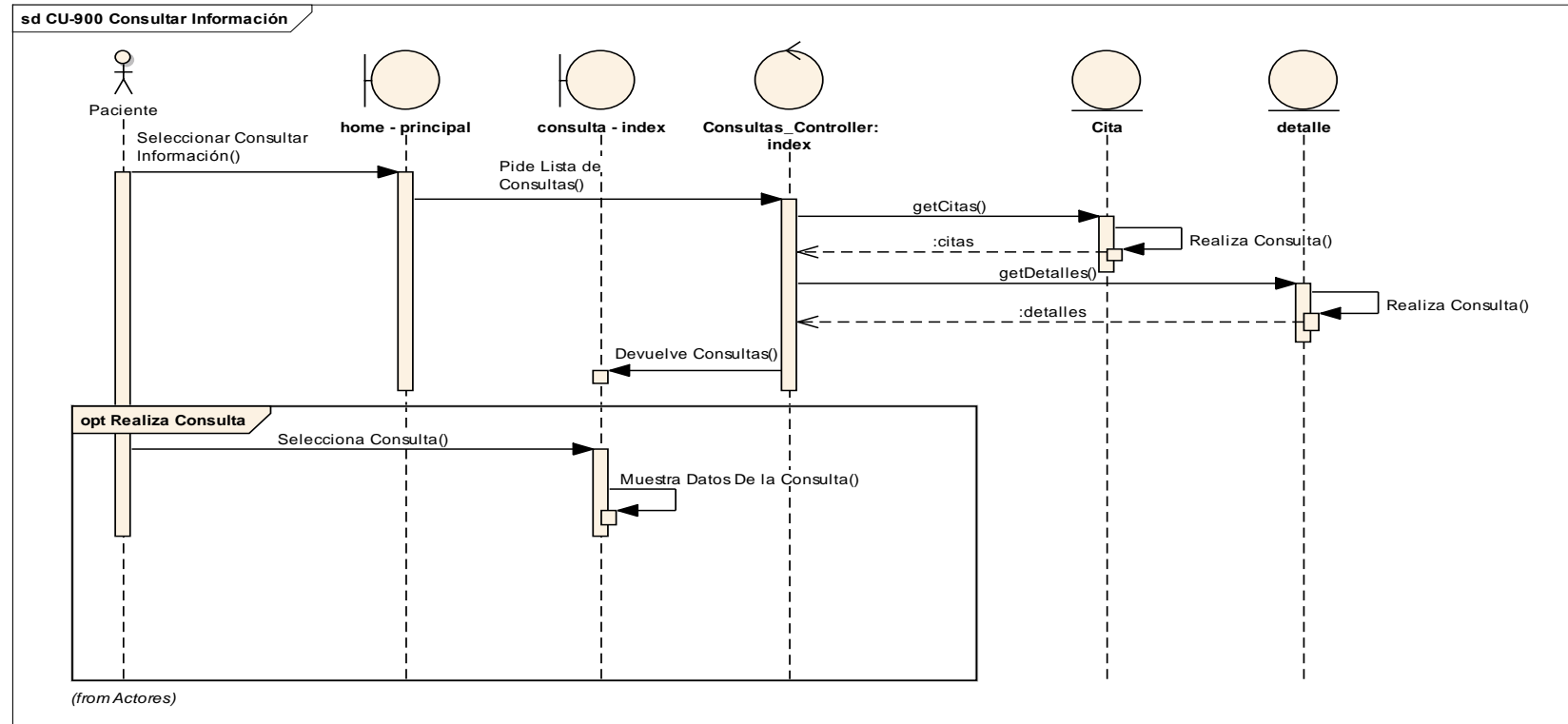


Figura 126 Diagrama de Secuencia: CU-900 Consultar Información

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.

2.1.2.2.3.15.1.4 Especificación de Métodos

A continuación una especificación de los métodos de los objetos utilizados en el proyecto.

2.1.2.2.3.15.1.4.1 Objeto Área

OBJETO:	Área
MÉTODO:	tratamientos
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
return \$this->has_many('Tratamiento','id_tratamiento')	
DESCRIPCIÓN	

OBJETO:	Área
MÉTODO:	diagnósticos
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
return \$this->has_many('Diagnostico','id_diagnostico')	
DESCRIPCIÓN	

OBJETO:	Área
MÉTODO:	áreas
ARGUMENTO:	(estado,per_page)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>\$permisos = DB::table('areas') ->where('estado','=',\$estado) ->paginate(\$per_page); return \$permisos;</pre>	
DESCRIPCIÓN	

OBJETO:	Área
MÉTODO:	adicionar
ARGUMENTO:	(user)

CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>parent::save(); \$sql=" INSERT INTO areas"; \$sql=\$sql."(area, descripcion)"; \$sql=\$sql." VALUES (".\$this->area." , ".\$this- >descripcion.") "; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,));</pre>	
DESCRIPCIÓN	

OBJETO:	Área
MÉTODO:	eliminar
ARGUMENTO:	(estado,id_area,user)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>parent::save(); \$sql=" UPDATE areas SET "; \$sql=\$sql."estado = ".\$estado ; \$sql=\$sql." WHERE id_area = ".\$id_area; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'updated_at' => \$this->updated_at,))</pre>	
DESCRIPCIÓN	

OBJETO:	Área
MÉTODO:	modificar
ARGUMENTO:	(id_area, user)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>parent::save(); \$sql=" UPDATE odontologos SET "; \$sql=\$sql."area = ".\$this->area . ', ' ; \$sql=\$sql."descripcion = ".\$this->descripcion ; \$sql=\$sql." WHERE id_area = ".\$this->\$id_area;</pre>	

<pre>DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'updated_at' => \$this->updated_at,))</pre>
DESCRIPCIÓN

Tabla 51 Objeto Área

2.1.2.2.3.15.1.4.2 Objeto Cita

OBJETO:	Cita
MÉTODO:	paciente
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
return \$this->belongs_to('Paciente','id_paciente')	
DESCRIPCIÓN	

Tabla 52 Objeto Cita

OBJETO:	Cita
MÉTODO:	adicionar
ARGUMENTO:	(user)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>parent::save(); \$sql=" INSERT INTO citas"; \$sql=\$sql."(id_odontologo,id_paciente,fecha_cita,estado,hora_cita,observaciones,tmp_min,usu ario)"; \$sql=\$sql." VALUES (".\$this->id_odontologo." , ".\$this->id_paciente." , ".\$this->fecha_cita." , ".\$this->observaciones." , ".\$this-> estado." , ".\$this->hora_cita." , ".\$this->tmp_min." , ".\$this->usuario.") "; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,));</pre>	
DESCRIPCIÓN	

2.1.2.2.3.15.1.4.3 Objeto Clínica

OBJETO:	Clínica
MÉTODO:	pacientes
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
return \$this->has_many('Paciente')	
DESCRIPCIÓN	

OBJETO:	Clínica
MÉTODO:	odontólogos
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
return \$this->has_many('Odontologo')	
DESCRIPCIÓN	

Tabla 53 Objeto Clínica

2.1.2.2.3.15.1.4.4 Objeto Detalle

OBJETO:	Detalle
MÉTODO:	presupuesto
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
return \$this->belongs_to('Presupuesto')	
DESCRIPCIÓN	

OBJETO:	Detalle
MÉTODO:	odontólogo
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
return \$this->belongs_to('Odontologo','id_odontologo')	
DESCRIPCIÓN	

OBJETO:	Detalle
MÉTODO:	diagnostico
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
return \$this->belongs_to('Diagnostico','id_diagnostico')	
DESCRIPCIÓN	

OBJETO:	Detalle
MÉTODO:	tratamiento
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
return \$this->belongs_to('Tratamiento','id_tratamiento')	
DESCRIPCIÓN	

OBJETO:	Detalle
MÉTODO:	diente
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
return \$this->belongs_to('Diente','id_diente')	
DESCRIPCIÓN	

OBJETO:	Detalle
MÉTODO:	adicionar
ARGUMENTO:	(user)
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
<pre>parent::save(); \$sql=" INSERT INTO detalles"; \$sql=\$sql."(id_presupuesto, id_diente, cara, id_diagnostico, id_odontologo,usuario)"; \$sql=\$sql." VALUES (".\$this->id_presupuesto." , ".\$this->id_diente." , ".\$this->cara." , ".\$this->id_diagnostico." , ".\$this->id_odontologo." , ".\$user.") "; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user,</pre>	

'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,));
DESCRIPCIÓN

Tabla 54 Objeto Detalle

2.1.2.2.3.15.1.4.5 Objeto Diagnostico

OBJETO:	Diagnostico
MÉTODO:	detalle
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	public
ESPECIFICACIÓN	
return \$this->has_many('Detalle','id_detalle')	
DESCRIPCIÓN	

OBJETO:	Diagnostico
MÉTODO:	área
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return \$this->belongs_to('Area','id_area')	
DESCRIPCIÓN	

OBJETO:	Diagnostico
MÉTODO:	adicionar
ARGUMENTO:	(user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
parent::save(); \$sql=" INSERT INTO diagnostico"; \$sql=\$sql."(id_area, descripcion, color, usuario)";	

<pre> \$sql=\$sql." VALUES (".\$this->id_area." , ".\$this-> >descripcion." , ".\$this->color." ,"; \$sql=\$sql." \$this->usuario." , ".\$user.")"; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,)); </pre>
DESCRIPCIÓN

Tabla 55 Objeto Diagnostico

2.1.2.2.3.15.1.4.6 Objeto Diente

OBJETO:	Diente
MÉTODO:	detalle
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return \$this->has_many('Detalle','id_detalle')	
DESCRIPCIÓN	

Tabla 56 Objeto Diente

2.1.2.2.3.15.1.4.7 Objeto Odontólogo

OBJETO:	Odontólogo
MÉTODO:	usuarios
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
return \$this->has_many('Usuario','id_usuario')	
DESCRIPCIÓN	

OBJETO:	Odontólogo
MÉTODO:	detalles
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	

return \$this->has_many('Detalle','id_detalle')
DESCRIPCIÓN

OBJETO:	Odontólogo
MÉTODO:	getOdontologos
ARGUMENTO:	(filtro,estado,per_page)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>\$od = DB::table('odontologos') ->where('ci','like','%'.strtoupper(\$filtro).%') ->where('estado','=',\$estado) ->paginate(\$per_page);</pre>	
DESCRIPCIÓN	

OBJETO:	Odontólogo
MÉTODO:	adicionar
ARGUMENTO:	(user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>\$this->id_clinica='1'; parent::save(); \$sql=" INSERT INTO odontologos"; \$sql=\$sql."(id_clinica, ci, nombres, ap, am, direccion, fecha_nacimiento, sexo, email,"; \$sql=\$sql."telef, matricula, estado, usuario)"; \$sql=\$sql." VALUES (1 , ".\$this->ci." , ".\$this->nombres." , ".\$this->ap." , ".\$this->am." , "; \$sql=\$sql." ".\$this->direccion." , ".\$this->fecha_nacimiento." , ".\$this->sexo." , ".\$this->email." , ".\$this->telef." , "; \$sql=\$sql." ".\$this->matricula." , "; \$sql=\$sql." ".\$this->estado." , ".\$user.")"; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,));</pre>	
DESCRIPCIÓN	

OBJETO:	Odontólogo
MÉTODO:	eliminar
ARGUMENTO:	(estado,id_odontologo,user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>parent::save(); \$sql=" UPDATE odontologos SET "; \$sql=\$sql."estado = ".\$estado ; \$sql=\$sql." WHERE id_odontologo =>".\$id_odontologo; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,));</pre>	
DESCRIPCIÓN	

OBJETO:	Odontólogo
MÉTODO:	modificar
ARGUMENTO:	(id_odontologo, user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>parent::save(); \$sql=" UPDATE odontologos SET "; \$sql=\$sql."nombres = ".\$this->nombres .', ,'; \$sql=\$sql."ap = ".\$this->ap .', ,'; \$sql=\$sql."am = ".\$this->am .', ,'; \$sql=\$sql."direccion = ".\$this->direccion .', ,'; \$sql=\$sql."fecha_nacimiento = ".\$this- >fecha_nacimiento .', ,'; \$sql=\$sql."sexo = ".\$this->sexo .', ,'; \$sql=\$sql."email = ".\$this->email .', ,'; \$sql=\$sql."telef = ".\$this->telef ; \$sql=\$sql." WHERE id_odontologo = ".\$this- >\$id_odontologo; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,));</pre>	
DESCRIPCIÓN	

--

2.1.2.2.3.15.1.4.8 Objeto Paciente

OBJETO:	Paciente
MÉTODO:	usuarios
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
\$this->has_many('Usuario','id_usuario')	
DESCRIPCIÓN	

OBJETO:	Paciente
MÉTODO:	antecedente
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
\$this->has_one('Antecedente','id_antecedente')	
DESCRIPCIÓN	

OBJETO:	Paciente
MÉTODO:	citas
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
\$this->has_many('Cita')	
DESCRIPCIÓN	

OBJETO:	Paciente
MÉTODO:	presupuesto
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
\$this->has_many('Presupuesto')	
DESCRIPCIÓN	

OBJETO:	Paciente
MÉTODO:	getPacientes
ARGUMENTO:	(user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>\$od = DB::table('pacientes') ->where('ci','like','%'.strtoupper(\$filtro).'%') ->where('estado','=',\$estado) ->paginate(\$per_page);</pre>	
DESCRIPCIÓN	

OBJETO:	Paciente
MÉTODO:	adicionar
ARGUMENTO:	(user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>\$this->id_clinica='1'; parent::save(); \$sql=" INSERT INTO pacientes"; \$sql=\$sql."(id_clinica, ci, nombres, ap, am, direccion, fecha_nacimiento, sexo, email,"; \$sql=\$sql."telef, autoriza,ocupacion, estado, usuario)"; \$sql=\$sql." VALUES (".\$this->id_clinica." ,".\$this->ci." , ".\$this->nombres." , ".\$this->ap." , ".\$this->am." , "; \$sql=\$sql. \$this->direccion." , ".\$this->fecha_nacimiento." , ".\$this->sexo." , ".\$this->email." , ".\$this->telef." , "; \$sql=\$sql. \$this->autoriza. " , ".\$this->ocupacion. " , "; \$sql=\$sql. \$this->estado. " , ".\$user.""); DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,));</pre>	
DESCRIPCIÓN	

OBJETO:	Paciente
MÉTODO:	eliminar

ARGUMENTO:	(user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>parent::save(); \$sql=" UPDATE pacientes SET "; \$sql=\$sql."estado = ".\$estado ; \$sql=\$sql." WHERE id_paciente =".\$id_paciente; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,));</pre>	
DESCRIPCIÓN	

OBJETO:	Paciente
MÉTODO:	modificar
ARGUMENTO:	(id_paciente, user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>parent::save(); \$sql=" UPDATE pacientes SET "; \$sql=\$sql."nombres = ".\$this->nombres .', ,'; \$sql=\$sql."ap = ".\$this->ap .', ,'; \$sql=\$sql."am = ".\$this->am .', ,'; \$sql=\$sql."direccion = ".\$this->direccion .', ,'; \$sql=\$sql."fecha_nacimiento = ".\$this-> >fecha_nacimiento .', ,'; \$sql=\$sql."sexo = ".\$this->sexo .', ,'; \$sql=\$sql."email = ".\$this->email .', ,'; \$sql=\$sql."telef = ".\$this->telef .', ,'; \$sql=\$sql."autoriza = ".\$this->autoriza .', ,'; \$sql=\$sql."ocupacion = ".\$this->ocupacion ; \$sql=\$sql." WHERE id_paciente = ".\$this-> >id_paciente; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,));</pre>	
DESCRIPCIÓN	

OBJETO:	Paciente
MÉTODO:	get_fecha_ingreso
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>\$date = new DateTime(\$this->created_at); return \$date->format('d-m-Y');</pre>	
DESCRIPCIÓN	

OBJETO:	Paciente
MÉTODO:	get_edad
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>\$fecha = \$this->fecha_nacimiento; \$fecha = str_replace("/", "-", \$fecha); \$fecha = date('Y/m/d', strtotime(\$fecha)); \$hoy = date('Y/m/d'); \$edad = \$hoy - \$fecha; return \$edad;</pre>	
DESCRIPCIÓN	

Tabla 57 Objeto Paciente

2.1.2.2.3.15.1.4.9 Objeto Permiso

OBJETO:	Permiso
MÉTODO:	usuarios
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>\$this->has_many_and_belongs_to('Usuario')</pre>	
DESCRIPCIÓN	

OBJETO:	Permiso
MÉTODO:	procesos

ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<code>\$this->has_many_and_belongs_to('Proceso')</code>	
DESCRIPCIÓN	

OBJETO:	Permiso
MÉTODO:	getPermisos
ARGUMENTO:	(estado, per_page)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre> \$permisos = DB::table('permisos') ->where('estado','=',\$estado) ->paginate(\$per_page); </pre>	
DESCRIPCIÓN	

OBJETO:	Permiso
MÉTODO:	adicionar
ARGUMENTO:	(user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre> parent::save(); \$sql=" INSERT INTO permisos"; \$sql=\$sql."(permiso, descripcion,usuario)"; \$sql=\$sql." VALUES (".\$this->permiso." , "; \$sql=\$sql.\$this->descripcion." , ".\$user.")"; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,)); </pre>	
DESCRIPCIÓN	

OBJETO:	Permiso
MÉTODO:	modificar
ARGUMENTO:	(id_permiso, user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	

<pre>parent::save(); \$sql=" UPDATE permisos SET "; \$sql=\$sql."permiso = ".\$this->permiso . ', '; \$sql=\$sql."descripcion = ".\$this->descripcion . ', '; \$sql=\$sql." WHERE id_permiso = ".\$this-> >\$id_permiso; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,));</pre>
DESCRIPCIÓN

OBJETO:	Permiso
MÉTODO:	in_objeto
ARGUMENTO:	(valor,conjunto)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>\$flag =false; if (is_object(\$valor)) { foreach (\$conjunto as \$key => \$value) { if (\$valor->id_permiso==\$value->id_permiso) { \$flag=true; } } } return \$flag;</pre>	
DESCRIPCIÓN	

Tabla 58 Objeto Permiso

2.1.2.2.3.15.1.4.10 Objeto Presupuesto

OBJETO:	Presupuesto
MÉTODO:	detalles
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
\$this->has_many('Detalle','id_detalle')	

DESCRIPCIÓN

OBJETO:	Presupuesto
MÉTODO:	paciente
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
\$this->belongs('Paciente')	
DESCRIPCIÓN	

OBJETO:	Presupuesto
MÉTODO:	detallePresupuesto
ARGUMENTO:	(id_paciente, per_page)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre> \$presupuesto=Presupuesto::where('id_paciente','=',\$id)->where('estado','=','1')->first(); if (is_null(\$presupuesto)) { \$detalles=null; } else { \$detalles=Detalle::where('id_presupuesto','=',\$presupuesto->id_presupuesto)->paginate(\$per_page); } </pre>	
DESCRIPCIÓN	

Tabla 59 Objeto Presupuesto

2.1.2.2.3.15.1.4.11 Objeto Proceso

OBJETO:	Proceso
MÉTODO:	permisos
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
\$this->has_many_and_belongs_to('Permiso')	
DESCRIPCIÓN	

OBJETO:	Proceso
MÉTODO:	in_objeto
ARGUMENTO:	(valor,conjunto)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre> \$flag =false; if (is_object(\$valor)) { foreach (\$conjunto as \$key => \$value) { if (\$valor->id_proceso==\$value->id_proceso) { \$flag=true; } } } return \$flag; </pre>	
DESCRIPCIÓN	

Tabla 60 Objeto Proceso

2.1.2.2.3.15.1.4.12 Objeto Tratamiento

OBJETO:	Tratamiento
MÉTODO:	detalles
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre> \$this->has_many('Detalle','id_detalle') </pre>	
DESCRIPCIÓN	

OBJETO:	Tratamiento
MÉTODO:	getTratamientos
ARGUMENTO:	(filtro, estado, per_page)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre> \$od = DB::table('tratamientos') ->where('tratamiento','like','%'.strtoupper(\$filtro).'%') ->where('estado','=',\$estado) ->paginate(\$per_page); </pre>	

DESCRIPCIÓN

OBJETO:	Tratamiento
MÉTODO:	adicionar
ARGUMENTO:	(user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>parent::save(); \$sql=" INSERT INTO tratamiento"; \$sql=\$sql.(id_area, tratamiento, costo, color, padre, tipo, usuario); \$sql=\$sql." VALUES (".\$this->id_area." , ".\$this- >tratamiento." , ".\$this->costo." , ".\$this->color." ,"; \$sql=\$sql. \$this->padre. " , "; \$sql=\$sql. \$this->tipo. " , ".\$user.""); DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,));</pre>	
DESCRIPCIÓN	

OBJETO:	Tratamiento
MÉTODO:	adicionar
ARGUMENTO:	(user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>parent::save(); \$sql=" INSERT INTO tratamiento"; \$sql=\$sql.(id_area, tratamiento, costo, color, padre, tipo, usuario); \$sql=\$sql." VALUES (".\$this->id_area." , ".\$this- >tratamiento." , ".\$this->costo." , ".\$this->color." ,"; \$sql=\$sql. \$this->padre. " , "; \$sql=\$sql. \$this->tipo. " , ".\$user.""); DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,</pre>	

DESCRIPCIÓN

Tabla 61 Objeto Tratamiento

2.1.2.2.3.15.1.4.13 Objeto Usuario

OBJETO:	Usuario
MÉTODO:	permisos
ARGUMENTO:	()
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
\$this->has_many_and_belongs_to('Permiso')	
DESCRIPCIÓN	

OBJETO:	Usuario
MÉTODO:	login
ARGUMENTO:	(user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre> \$u = null; \$users=Usuario::all(); foreach (\$users as \$user) { if(\$user->username==\$user2['username'] && \$user->password==\$user2['password']){ \$u= \$user; break; } } </pre>	
DESCRIPCIÓN	

OBJETO:	Usuario
MÉTODO:	getVerificarUsuario
ARGUMENTO:	(username, password)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
\$usuario = Usuario::where('username', '=', \$username)->where('password', '=', \$password)->first();	
DESCRIPCIÓN	

--

OBJETO:	Usuario
MÉTODO:	usuarioActivos
ARGUMENTO:	(id)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
\$count = Usuario::where('id_odontologo', '=', \$id)->where('estado', '=', '1')->count();	
DESCRIPCIÓN	

OBJETO:	Usuario
MÉTODO:	ModificarUsuario
ARGUMENTO:	(id , user)
CONTROL DE EXPORTACIÓN:	Public
ESPECIFICACIÓN	
<pre>parent::save(); \$sql=" UPDATE usuarios SET "; \$sql=\$sql."(username, password, usuario)"; \$sql=\$sql."username = ".\$this->username .',', '; \$sql=\$sql."usuario = ".\$this->usuario .',', '; \$sql=\$sql."password = ".\$this->password ; \$sql=\$sql." WHERE id_odontologo = ".\$id; DB::table('registro_huella')->insert(array('consulta' => \$sql, 'usuario' => \$user, 'created_at' => \$this->created_at, 'updated_at' => \$this->updated_at,));</pre>	
DESCRIPCIÓN	

Tabla 62 Objeto Usuario

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.

2.1.2.2.3.15.2.3 Creación de la Base de Datos

```
CREATE TABLE clinicas (
  id_clinica int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo de la
  clinica dental',
  clinica varchar(30) NOT NULL COMMENT 'Nombre de la clinica dental',
  ruc varchar(30) NOT NULL COMMENT 'registro unico de la clinica dental',
  prop varchar(50) NOT NULL COMMENT 'nombres del propietario(a) de la clinica ',
  direccion varchar(50) NOT NULL COMMENT ' direccionde la clinica dental',
  telef varchar(30) NOT NULL COMMENT ' telefono de la clinica dental',
  pag_web varchar(30) COMMENT 'pagina Web de la clinica dental',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del registro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el
  sistema',
  PRIMARY KEY (id_clinica)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena datos de la clinica dental';
```

```
CREATE TABLE odontologos (
  id_odontologo int NOT NULL AUTO_INCREMENT ,
  id_clinica int NOT NULL COMMENT 'Identificativo de la clinica dental',
  ci varchar(20) NOT NULL COMMENT 'cedula de Identidad de la odontologo',
  nombres varchar(20) NOT NULL COMMENT 'nombres de de la odontologo',
  ap varchar(30) NOT NULL COMMENT 'apellido Paterno de la odontologo',
  am varchar(30) COMMENT 'apellido Materno de de la odontologo',
  direccion varchar(50) NOT NULL COMMENT ' direccion de la odontologo',
  fecha_nacimiento date NOT NULL COMMENT 'fecha de Nacimiento de la
  odontologo',
  sexo varchar(5) COMMENT 'genero de la odontologo',
  email varchar(50) COMMENT 'correo Electronico de la odontologo',
  telef varchar(30) COMMENT 'telefono de la odontologo',
  matricula varchar(15) NOT NULL COMMENT 'matricula profesional del
  odontologo',
  estado varchar(1) DEFAULT '1' COMMENT 'estado activo 1, incativo 0',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del registro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el
  sistema',
  PRIMARY KEY (id_odontologo),
  FOREIGN KEY (id_clinica) REFERENCES clinicas (id_clinica)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena datos del medico odontolo que
integran la clinica dental';
```

```

CREATE TABLE pacientes (
  id_paciente int NOT NULL AUTO_INCREMENT ,
  id_clinica int NOT NULL COMMENT 'Identificativo de la clinica dental',
  ci varchar(20) NOT NULL COMMENT 'cedula de Identidad de la odontologo',
  nombres varchar(20) NOT NULL COMMENT 'nombres de de la odontologo',
  ap varchar(30) COMMENT 'apellido Paterno de la odontologo',
  am varchar(30) COMMENT 'apellido Materno de de la odontologo',
  direccion varchar(50) NOT NULL COMMENT ' direccion de la odontologo',
  fecha_nacimiento date NOT NULL COMMENT 'fecha de Nacimiento de la
odontologo',
  sexo varchar(5) COMMENT 'genero de la odontologo',
  email varchar(50) COMMENT 'correo Electronico de la odontologo',
  telef varchar(30) COMMENT 'telefono de la odontologo',
  autoriza varchar(50) COMMENT 'nombres de la persona que autoriza los
tratamientos a ejecutar ',
  ocupacion varchar(50) NOT NULL COMMENT 'ocupacion del paciente',
  estado varchar(1) DEFAULT '1' COMMENT 'estado activo 1, incativo 0',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del registro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja es
sistema',
  PRIMARY KEY (id_paciente),
  FOREIGN KEY (id_clinica) REFERENCES clinicas (id_clinica)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena datos del paciente que
integran la clinica dental';

```

```

CREATE TABLE usuarios (
  id_usuario int NOT NULL AUTO_INCREMENT COMMENT 'identificativo de
usuario',
  id_odontologo int COMMENT 'Identificativo del odontologo',
  id_paciente int COMMENT 'Identificativo del paciente',
  username varchar(20) NOT NULL COMMENT 'username del usuario',
  password varchar(20) NOT NULL COMMENT 'password del usuario',
  estado varchar(1) DEFAULT '1' COMMENT 'estado activo 1, incativo 0',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del registro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el
sistema',
  PRIMARY KEY (id_usuario),
  FOREIGN KEY (id_odontologo) REFERENCES odontologos (id_odontologo),
  FOREIGN KEY (id_paciente) REFERENCES pacientes (id_paciente)

```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena informacion del usuario del sistema';
```

```
CREATE TABLE permisos (
  id_permiso int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo
delpermiso',
  permiso varchar(30) NOT NULL COMMENT 'nombre del permiso',
  descripcion varchar(30) NOT NULL COMMENT 'descripcion del permiso',
  estado varchar(1) DEFAULT '1' COMMENT 'estado del permiso activo o inactivo',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del registro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el
sistema',
  PRIMARY KEY (id_permiso)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena informacion de los permisos a los usuarios';
```

```
CREATE TABLE permiso_usuario (
  id int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo de la reacion
usuarios permisos',
  usuario_id int NOT NULL COMMENT 'identificativo del usuario',
  permiso_id int NOT NULL COMMENT 'identificativo del permiso',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del registro',
  PRIMARY KEY (id),
  FOREIGN KEY (usuario_id) REFERENCES usuarios (id_usuario),
  FOREIGN KEY (permiso_id) REFERENCES permisos (id_permiso)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena relacion entre usuarios y permisos';
```

```
CREATE TABLE procesos (
  id_proceso int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo del
procesos',
  proceso varchar(30) NOT NULL COMMENT 'nombre del proceso',
  descripcion varchar(30) NOT NULL COMMENT 'descripcion del proceso',
  url varchar(30) NOT NULL COMMENT ' url del proceso',
  estado varchar(1) DEFAULT '1' COMMENT 'estado del proceso activo o inactivo',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del registro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el
sistema',
  PRIMARY KEY (id_proceso)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena informacion de los procesos de los permisos';
```

```

CREATE TABLE permiso_proceso (
  id int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo de la relacion
permisos procesos',
  permiso_id int NOT NULL COMMENT 'identificativo del permiso',
  proceso_id int NOT NULL COMMENT 'identificativo del proceso',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del regsitro',
  PRIMARY KEY (id),
  FOREIGN KEY (proceso_id) REFERENCES procesos (id_proceso),
  FOREIGN KEY (permiso_id) REFERENCES permisos (id_permiso)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena relacion entre permisos y procesos';

```

```

CREATE TABLE citas (
  id_cita int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo de la
clinica dental',
  id_paciente int NOT NULL COMMENT 'identificativo del paciente',
  id_odontologo int NOT NULL COMMENT 'identificativo del odontologo',
  fecha_cita date NOT NULL COMMENT 'fecha de la cita',
  estado varchar(1) NOT NULL COMMENT 'estado de la cita Pendiente Cancelada
Confirmada',
  hora_cita time NOT NULL COMMENT 'hora de la cita',
  tmp_min varchar(30) NOT NULL COMMENT 'tiempo de la cita en minutos',
  observaciones varchar(50) NOT NULL COMMENT 'observaciones de la cita',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del regsitro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el
sistema',
  PRIMARY KEY (id_cita),
  FOREIGN KEY (id_paciente) REFERENCES pacientes (id_paciente),
  FOREIGN KEY (id_odontologo) REFERENCES odontologos (id_odontologo)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena informacion de las citas';

```

```

CREATE TABLE antecedentes (
  id_antecedente int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo
del antecedente personal de un paciente',
  id_paciente int NOT NULL COMMENT 'identificativo del paciente',
  fecha_rev date NOT NULL COMMENT 'ultima fecha de revision',
  ant_fam varchar(200) COMMENT 'descripcion de antecedentes familiares',
  enf_inf varchar(200) COMMENT 'descripcion de enfermedades de la infancia',
  form_diente varchar(200) COMMENT 'descripcion de la forma de los dientes',
  colo_diente varchar(200) COMMENT 'descripcion del color de los dientes',

```

```
tama_diente varchar(200) COMMENT 'descripcion del tamaño de los dientes',
tipo_sangre varchar(5) COMMENT 'tipo de sangre',
tiempo_coag int(10) COMMENT 'tiempo de coagulacion',
tiempo_sang int(10) COMMENT 'tiempo de sangria',
estado varchar(1) DEFAULT '1' COMMENT 'estado activo 1, inactivo 0',
created_at datetime NOT NULL COMMENT 'fecha de insercion',
updated_at datetime NOT NULL COMMENT 'estado del registro',
usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el sistema',
PRIMARY KEY (id_antecedente),
FOREIGN KEY (id_paciente) REFERENCES pacientes (id_paciente)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena informacion de los antecedentes personales del paciente';
```

```
CREATE TABLE presupuestos (
  id_presupuesto int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo
del presupuesto de un paciente',
  id_paciente int NOT NULL COMMENT 'identificativo del paciente',
  fecha_pres date NOT NULL COMMENT 'fecha que se realizo el presupuesto',
  coste_apro float NOT NULL COMMENT 'coste total apronado',
  coste_obt float NOT NULL COMMENT 'coste total obtenido por la suma de todos
los trat asignados sujeto a descuentos',
  estado varchar(1) DEFAULT '1' COMMENT 'estado del presupuesto activo o
inactivo',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del registro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el
sistema',
PRIMARY KEY (id_presupuesto),
FOREIGN KEY (id_paciente) REFERENCES pacientes (id_paciente)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena informacion de los presupuestos del paciente del paciente';
```

```
CREATE TABLE areas (
  id_area int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo de la area
odontologica',
  area varchar(1) NOT NULL COMMENT 'tipo de area: Superficie - Raiz - todo Diente
- Corona(O) - Cavidad dentaria',
  descripcion varchar(30) NOT NULL COMMENT 'descripcion textual de la area
odontologica',
  estado varchar(1) DEFAULT '1' COMMENT 'estado activo 1, inactivo 0',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del registro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el
sistema',
```

```

PRIMARY KEY (id_area)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena datos de la area odontologica';

```

```

CREATE TABLE tratamientos (
  id_tratamiento int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo del
tratamiento',
  id_area int COMMENT 'area odontologica a la que pertenece el tratamiento',
  tratamiento varchar(30) NOT NULL COMMENT 'nombre del tratamiennto',
  costo float COMMENT 'costo del tratamiento',
  color varchar(10) COMMENT 'color que identifica el tratamiento',
  padre int(10) COMMENT 'Identificativo del tratamiento padre',
  tipo varchar(1) NOT NULL COMMENT 'tipo del tratamiento Detalle o Grupo',
  estado varchar(1) DEFAULT '1' NOT NULL COMMENT 'estado del tratamiento
activo inactivo',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del regsitro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el
sistema',
  PRIMARY KEY (id_tratamiento),
  FOREIGN KEY (id_area) REFERENCES areas (id_area)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena informacion de los tratamientos de la clinica';

```

```

CREATE TABLE diagnosticos (
  id_diagnostico int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo
del diagnostico',
  id_area int COMMENT 'area odontologica a la que pertenece el tratamiento',
  descripcion varchar(30) NOT NULL COMMENT 'descripcion textual del
diagnostico',
  color varchar(10) COMMENT 'representa el codigo de un color unico para un
diagnostico dentro de una rama de la odontologia',
  estado varchar(1) DEFAULT '1' NOT NULL COMMENT 'estado del diagnostico
activo inactivo',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del regsitro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el
sistema',
  PRIMARY KEY (id_diagnostico),
  FOREIGN KEY (id_area) REFERENCES areas (id_area)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena informacion de los diagnosticos de la clinica';

```

```

CREATE TABLE dientes (
  id_diente int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo de la
  cara de la cavidad dentaria, primer cifra cuadrante, segunda cifra posicion (cavidad)',
  descripcion varchar(30) NOT NULL COMMENT 'descripcion textual de la cara de
  la cavidad',
  estado varchar(1) DEFAULT '1' COMMENT 'estado activo 1, inactivo 0',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del regsitro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el
  sistema',
  PRIMARY KEY (id_diente)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena datos de la cara de las cavidades dentarias';

```

```

INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (11,'CENTRAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (12,'LATERAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (13,'CANINO','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (14,'PRIMER BICUSPIDE O PREMOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (15,'SEGUNDO BICUSPIDE O PREMOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (16,'PRIEMR MOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (17,'SEGUNDO MOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (18,'TERCER MOLAR O MUELA CORDAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (21,'CENTRAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (22,'LATERAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (23,'CANINO','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (24,'PRIMER BICUSPIDE O PREMOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (25,'SEGUNDO BICUSPIDE O PREMOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (26,'PRIEMR MOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (27,'SEGUNDO MOLAR','');

```

```
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (28,'TERCER MOLAR O MUELA CORDAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (31,'CENTRAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (32,'LATERAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (33,'CANINO','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (34,'PRIMER BICUSPIDE O PREMOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (35,'SEGUNDO BICUSPIDE O PREMOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (36,'PRIMER MOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (37,'SEGUNDO MOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (38,'TERCER MOLAR O MUELA CORDAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (41,'CENTRAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (42,'LATERAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (43,'CANINO','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (44,'PRIMER BICUSPIDE O PREMOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (45,'SEGUNDO BICUSPIDE O PREMOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (46,'PRIEMR MOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (47,'SEGUNDO MOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (48,'TERCER MOLAR O MUELA CORDAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (51,'CENTRAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (52,'LATERAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (53,'CANINO','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (54,'PRIMER MOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (55,'SEGUNDO','');
INSERT INTO dientes( id_diente, descripcion , usuario)
```

```

VALUES (61,'CENTRAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (62,'LATERAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (63,'CANINO','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (64,'PRIMER MOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (65,'SEGUNDO','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (71,'CENTRAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (72,'LATERAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (73,'CANINO','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (74,'PRIMER MOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (75,'SEGUNDO','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (81,'CENTRAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (82,'LATERAL','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (83,'CANINO','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (84,'PRIMER MOLAR','');
INSERT INTO dientes( id_diente, descripcion , usuario)
VALUES (85,'SEGUNDO','');

```

```

CREATE TABLE detalles (
  id_detalle int(10) NOT NULL AUTO_INCREMENT COMMENT 'Identificativo del
  datalle del diagnostico',
  id_presupuesto int(10) NOT NULL COMMENT 'Identificativo del presupuesto',
  id_diente int(10) NOT NULL COMMENT 'numero de cavidad dentaria a la que se la
  aplica un diagnostico',
  cara varchar(10) NOT NULL COMMENT 'identificativo de cara del diente que tiene
  el diagnostico',
  id_diagnostico int(10) NOT NULL COMMENT 'identificativo del diagnostico
  otorgado al paciente por cada cavidad y cara',
  id_tratamiento int(10) COMMENT 'identificativo del tratamiento que se aplicara ',
  id_odontologo int(10) NOT NULL COMMENT 'identificativo del odontolog que
  realiza el examen',
  fecha_det date NOT NULL COMMENT 'fecha de ultima revision',
  observaciones varchar(30) COMMENT 'Observacion del seguimiento',

```

```

estado varchar(1) DEFAULT '1' COMMENT 'estado activo 1, inactivo 0',
created_at datetime NOT NULL COMMENT 'fecha de insercion',
updated_at datetime NOT NULL COMMENT 'estado del registro',
usuario varchar(20) NOT NULL COMMENT 'identificativo del usuario',
PRIMARY KEY (id_detalle),
FOREIGN KEY (id_presupuesto) REFERENCES presupuestos (id_presupuesto),
FOREIGN KEY (id_diente) REFERENCES dientes (id_diente),
FOREIGN KEY (id_diagnostico) REFERENCES diagnosticos (id_diagnostico),
FOREIGN KEY (id_tratamiento) REFERENCES tratamientos (id_tratamiento),
FOREIGN KEY (id_odontologo) REFERENCES odontologos (id_odontologo)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='contiene ralacion con las piezas dentaria, diagnosticos y tratamientos
realizados';

```

```

CREATE TABLE genera_fac (
  id_genera int NOT NULL AUTO_INCREMENT ,
  autorizacion varchar(15) NOT NULL COMMENT 'autorizacion para generacion de
facturas',
  nit varchar(15) NOT NULL COMMENT 'NIT de la empresa',
  num_inicio int COMMENT'numero inicial de facturas',
  num_fin int COMMENT'numero final de la generacion de facturas',
  fecha_emision date NOT NULL COMMENT 'Fecha emision para controlar el limite
de validez',
  llave_dosificacion varchar(50) NOT NULL COMMENT 'Para generar el codigo de
control',
  num_emitidos int COMMENT 'contador automatico para saber cuantas facturas se
realizaron',
  estado varchar(1) DEFAULT '1' COMMENT 'estado activo 1, inactivo 0',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del registro',
  usuario int NOT NULL COMMENT 'identificativo del usuario que maneja el
sistema',
  PRIMARY KEY (id_genera),
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Contiene informacion de la generacion de facturas que autoriza la renta';

```

```

CREATE TABLE facturas (
  id_factura int(10) NOT NULL COMMENT 'numero de factura',
  id_genera int(10) NOT NULL COMMENT 'Identificativo de la generacion de
factura',
  ci varchar(20) NOT NULL COMMENT 'ci del paciente',
  ci_u varchar(20) NOT NULL COMMENT 'ci del usuario para extraer informacion de
quien realizo la transaccion',
  nit_ci varchar(20) NOT NULL COMMENT 'nit o ci del paciente',
  nombre varchar(50) NOT NULL COMMENT 'nombre del paciente',

```

```

fecha date NOT NULL COMMENT 'fecha de emision de factura',
cod_control varchar(50) NOT NULL COMMENT 'codigo de control generado',
estado varchar(1) DEFAULT '1' COMMENT 'estado activo 1, inactivo 0',
created_at datetime NOT NULL COMMENT 'fecha de insercion',
updated_at datetime NOT NULL COMMENT 'estado del registro',
usuario varchar(20) NOT NULL COMMENT 'identificativo del usuario',
PRIMARY KEY (id_factura),
FOREIGN KEY (id_genera) REFERENCES genera_fac (id_genera)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Es la factura que se emite al cliente';

```

```

CREATE TABLE det_facturas (
  id_detalle int(10) NOT NULL AUTO_INCREMENT ,
  id_factura int(10) NOT NULL COMMENT 'identificativo de factura',
  id_tratamiento int(10) NOT NULL COMMENT 'Identificador del tratamiento',
  monto varchar(20) NOT NULL COMMENT 'precio real que se cobra del arancel
modificado por el odontologo',
  descripcion varchar(50) NOT NULL COMMENT 'descripcion del tratamiento que
hizo',
  estado varchar(1) DEFAULT '1' COMMENT 'se encuentra en estado inactivo ya que
puede ser que este anulado',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del registro',
  usuario varchar(20) NOT NULL COMMENT 'identificativo del usuario',
  PRIMARY KEY (id_detalle),
  FOREIGN KEY (id_factura) REFERENCES facturas (id_factura),
  FOREIGN KEY (id_tratamiento) REFERENCES tratamientos (id_tratamiento)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Es la factura que se emite al cliente';

```

```

CREATE TABLE registro_huella (
  id int NOT NULL AUTO_INCREMENT COMMENT 'Identificativo del registro de
huella',
  consulta varchar(250) NOT NULL COMMENT 'consulta realizada por el usuario',
  usuario varchar(20) NOT NULL COMMENT 'usuario',
  created_at datetime NOT NULL COMMENT 'fecha de insercion',
  updated_at datetime NOT NULL COMMENT 'estado del registro',
  PRIMARY KEY (id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1
COMMENT='Almacena informacion del registro de la huella';

```

2.1.2.2.3.15.2.4 Diccionario de Datos

2.1.2.2.3.15.2.4.1 Antecedentes

Columna	Tipo	Nulo	Predeterminado	Enlaces a
id_antecedente	int(11)	No		
id_paciente	int(11)	No		pacientes - > id_paciente
fecha_rev	date	No		
ant_fam	varchar(200)	Sí	NULL	
enf_inf	varchar(200)	Sí	NULL	
form_diente	varchar(200)	Sí	NULL	
colo_diente	varchar(200)	Sí	NULL	
tama_diente	varchar(200)	Sí	NULL	
tipo_sangre	varchar(5)	Sí	NULL	
tiempo_coag	int(10)	Sí	NULL	
tiempo_sang	int(10)	Sí	NULL	
estado	varchar(1)	Sí	1	
created_at	datetime	No		
updated_at	datetime	No		
usuario	int(11)	No		

2.1.2.2.3.15.2.4.2 Áreas

Columna	Tipo	Nulo	Predeterminado
id_area	int(11)	No	
area	varchar(1)	No	
descripcion	varchar(30)	No	
estado	varchar(1)	Sí	1
created_at	datetime	No	
updated_at	datetime	No	
usuario	int(11)	No	

2.1.2.2.3.15.2.4.3 Citas

Columna	Tipo	Nulo	Predeterminado	Enlaces a
id_cita	int(11)	No		
id_paciente	int(11)	No		pacientes -> id_paciente

id_odontologo	int(11)	No	odontologos - > id_odontologo
fecha_cita	date	No	
estado	varchar(1)	No	
hora_cita	time	No	
tmp_min	varchar(30)	No	
observaciones	varchar(50)	No	
created_at	datetime	No	
updated_at	datetime	No	
usuario	int(11)	No	

2.1.2.2.3.15.2.4.4 Clínicas

Columna	Tipo	Nulo	Predeterminado
id_clinica	int(11)	No	
clinica	varchar(30)	No	
ruc	varchar(30)	No	
prop	varchar(50)	No	
direccion	varchar(50)	No	
telef	varchar(30)	No	
pag_web	varchar(30)	Sí	<i>NULL</i>
created_at	datetime	No	
updated_at	datetime	No	
usuario	int(11)	No	

2.1.2.2.3.15.2.4.5 Detalles

Columna	Tipo	Nulo	Predeterminado	Enlaces a
id_detalle	int(10)	No		
id_presupuesto	int(10)	No		presupuestos - > id_presupuesto
id_diente	int(10)	No		dientes -> id_diente
cara	varchar(10)	No		
id_diagnostico	int(10)	No		diagnosticos - > id_diagnostico

id_tratamiento	int(10)	No		tratamientos - > id_tratamiento
id_odontologo	int(10)	No		odontologos - > id_odontologo
fecha_det	date	No		
dibu	varchar(1)	No		
estado	varchar(1)	Sí	1	
created_at	datetime	No		
updated_at	datetime	No		
usuario	varchar(20)	No		

2.1.2.2.3.15.2.4.6 Diagnósticos

Columna	Tipo	Nulo	Predeterminado	Enlaces a
id_diagnostico	int(11)	No		
id_area	int(11)	Sí	<i>NULL</i>	areas -> id_area
descripcion	varchar(30)	No		
color	varchar(10)	Sí	<i>NULL</i>	
estado	varchar(1)	No	1	
created_at	datetime	No		
updated_at	datetime	No		
usuario	int(11)	No		

2.1.2.2.3.15.2.4.7 Dientes

Columna	Tipo	Nulo	Predeterminado
id_diente	int(11)	No	
descripcion	varchar(30)	No	
estado	varchar(1)	Sí	1
created_at	datetime	No	
updated_at	datetime	No	
usuario	int(11)	No	

2.1.2.2.3.15.2.4.8 Odontólogos

Columna	Tipo	Nulo	Predeterminado	Enlaces a
id_odontologo	int(11)	No		
id_clinica	int(11)	No		clinicas - > id_clinica
ci	varchar(20)	No		
nombres	varchar(20)	No		
ap	varchar(30)	Sí	NULL	
am	varchar(30)	Sí	NULL	
direccion	varchar(50)	No		
fecha_nacimiento	date	No		
sexo	varchar(5)	Sí	NULL	
email	varchar(50)	Sí	NULL	
telef	varchar(30)	Sí	NULL	
matricula	varchar(15)	No		
estado	varchar(1)	Sí	1	
created_at	datetime	No		
updated_at	datetime	No		
usuario	int(11)	No		

2.1.2.2.3.15.2.4.9 Pacientes

Columna	Tipo	Nulo	Predeterminado	Enlaces a
id_paciente	int(11)	No		
id_clinica	int(11)	No		clinicas - > id_clinica
ci	varchar(20)	No		
nombres	varchar(20)	No		
ap	varchar(30)	Sí	NULL	
am	varchar(30)	Sí	NULL	
direccion	varchar(50)	No		
fecha_nacimiento	date	No		
sexo	varchar(5)	Sí	NULL	
email	varchar(50)	Sí	NULL	
telef	varchar(30)	Sí	NULL	
autoriza	varchar(50)	Sí	NULL	

ocupacion	varchar(50)	No	
estado	varchar(1)	Sí	1
created_at	datetime	No	
updated_at	datetime	No	
usuario	int(11)	No	

2.1.2.2.3.15.2.4.10 Permisos

Columna	Tipo	Nulo	Predeterminado
id_permiso	int(11)	No	
permiso	varchar(30)	No	
descripcion	varchar(30)	No	
estado	varchar(1)	Sí	1
created_at	datetime	No	
updated_at	datetime	No	
usuario	int(11)	No	

2.1.2.2.3.15.2.4.11 Presupuestos

Columna	Tipo	Nulo	Predeterminado	Enlaces a
id_presupuesto	int(11)	No		
id_paciente	int(11)	No		pacientes - > id_paciente
fecha_pres	date	No		
coste_apro	float	No		
coste_obt	float	No		
estado	varchar(1)	Sí	1	
created_at	datetime	No		
updated_at	datetime	No		
usuario	int(11)	No		

2.1.2.2.3.15.2.4.12 Procesos

Columna	Tipo	Nulo	Predeterminado
id_proceso	int(11)	No	

proceso	varchar(30)	No
descripcion	varchar(30)	No
url	varchar(30)	No
estado	varchar(1)	Sí 1
created_at	datetime	No
updated_at	datetime	No
usuario	int(11)	No

2.1.2.2.3.15.2.4.13 Registro_Huella

Columna	Tipo	Nulo	Predeterminado
id	int(11)	No	
consulta	varchar(250)	No	
usuario	varchar(20)	No	
created_at	datetime	No	
updated_at	datetime	No	

2.1.2.2.3.15.2.4.14 Tratamientos

Columna	Tipo	Nulo	Predeterminado	Enlaces a
id_tratamiento	int(11)	No		
id_area	int(11)	Sí	<i>NULL</i>	areas -> id_area
tratamiento	varchar(30)	No		
costo	float	Sí	<i>NULL</i>	
color	varchar(10)	Sí	<i>NULL</i>	
padre	int(10)	Sí	<i>NULL</i>	
tipo	varchar(1)	No		
estado	varchar(1)	No	1	
created_at	datetime	No		
updated_at	datetime	No		
usuario	int(11)	No		

2.1.2.2.3.15.2.4.15 Usuarios

Columna	Tipo	Nulo	Predeterminado	Enlaces a
id_usuario	int(11)	No		

id_odontologo	int(11)	Sí	<i>NULL</i>	odontologos - > id_odontologo
id_paciente	int(11)	Sí	<i>NULL</i>	pacientes -> id_paciente
username	varchar(20)	No		
password	varchar(20)	No		
estado	varchar(1)	Sí	1	
created_at	datetime	No		
updated_at	datetime	No		
usuario	int(11)	No		

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

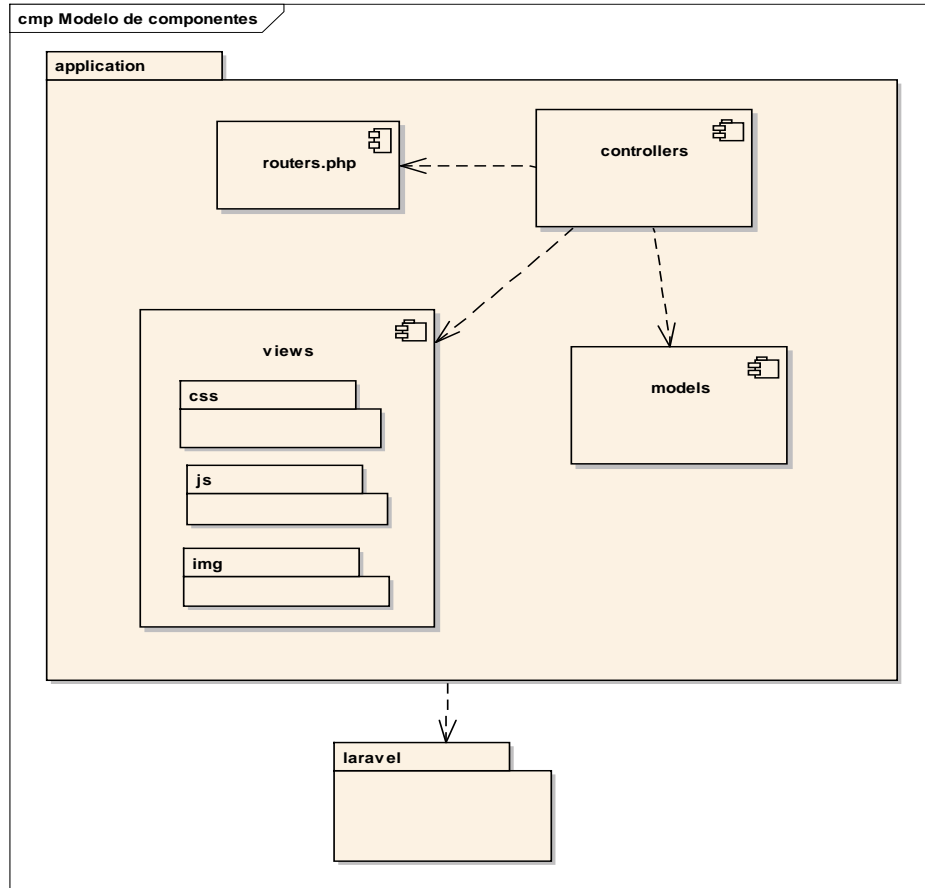


Figura 130 Diagrama de Componentes

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

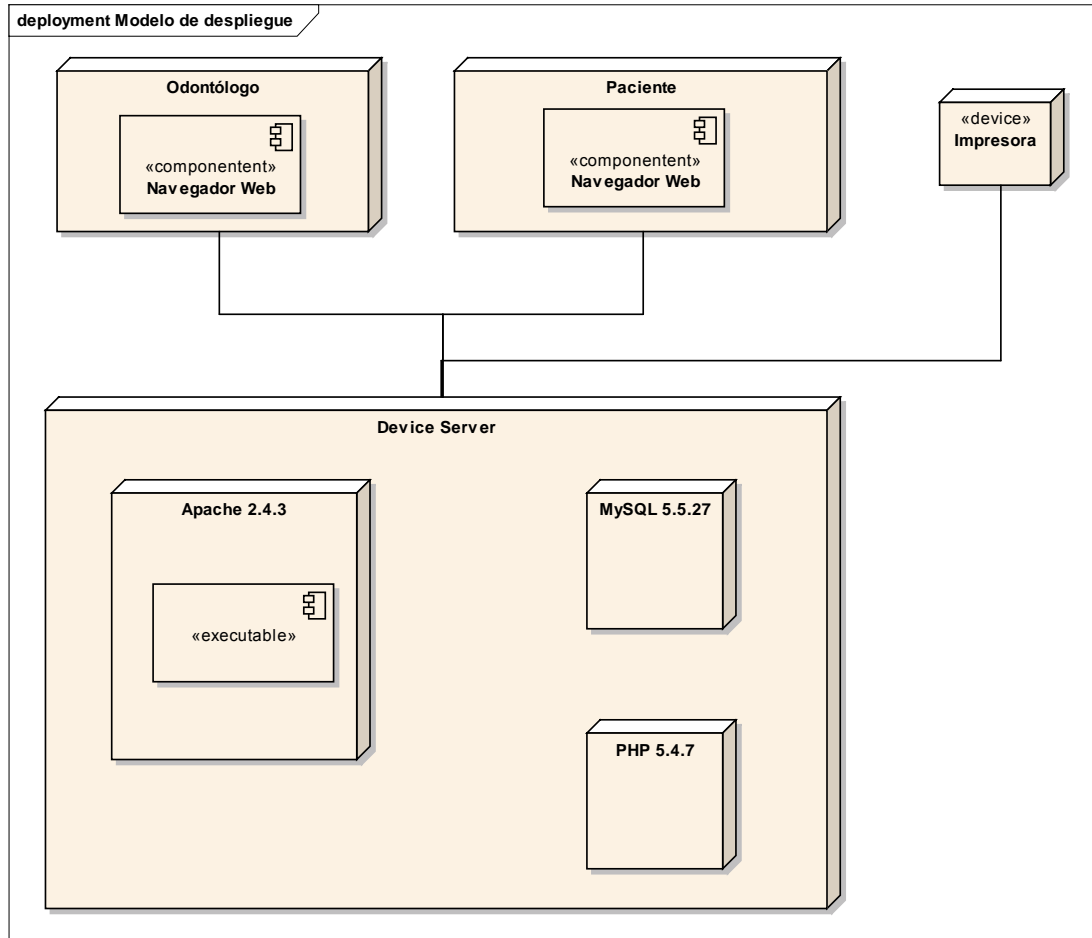


Figura 131 Diagrama de despliegue

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 al proyecto IRACEMA Virtual para el Mejoramiento en el control del historial clínico odontológico 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 clínica dental IRACEMA, 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 63 Hardware

SOFTWARE		
Aplicación	Versión	Descripción
Sistema para el Mejoramiento del control de historiales clínico odontológicos	0.3	Sistema desarrollado por el grupo de trabajo.
Visual Paradigm for UML Community Edition	8.0	Herramienta de Diseño y Desarrollo de Diagramas UML para RUP
MySql	5.1.53	Motor de Base de Datos Relacional.

Tabla 64 Software

OTROS		
Descripción	Versión	Descripción
Documentación	0.3	Documentación de Desarrollo de Software

Tabla 65 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 Informático para el Mejoramiento del control de historiales clínico odontológicos

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 66 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 67 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 68 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 69 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	1. Alfanumérico	3. Vacío
Tamaño: Username	2. $5 \leq \text{Caracteres} \leq 20$	4. Caracteres >50 y Caracteres <5
Tipo: Password	5. Alfanumérico	7. Vacío
Tamaño: Password	6. $5 \leq \text{Caracteres} \leq 20$	8. Caracteres >50 y Caracteres <5

Tabla 70. 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

Interfaz: Formulario ABM Gestión Permisos

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Permiso Tamaño: Permiso	1. Alfabético 2. $1 \leq \text{Caracteres} \leq 30$	3. Vacío 4. Caracteres >30
Tipo: Descripción Tamaño: Descripción	5. Alfabético 6. $0 \leq \text{Caracteres} \leq 30$	7. Caracteres >30

Tabla 71 Caso de Prueba Caja Negra: Gestión Permisos

Casos de Prueba válidas

CP1	
Permiso	Descripción
Técnico Dental	Solo permite realizar seguimiento.

Cubre las clases de equivalencias válidas: 1-3-5-7

Casos de Pruebas Inválidas

CP2	
Permiso	Descripción
	Solo permite realizar seguimiento.

Cubre las clases de equivalencias inválidas: 2

CP3	
Nombre	Descripción
>35	Solo permite realizar seguimiento.

Cubre las clases de equivalencias inválidas: 4

Interfaz: Formulario ABM Gestionar Odontólogo

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: CI Tamaño: CI	1. Numero 2. 7	3. Letra o carácter especial 4. !=7
Tipo: Nombres Tamaño: Nombres	5. Alfabético 6. 1<=Caracteres<=20	7. Vacío 8. Letra>20
Tipo: Apellido Paterno Tamaño: Apellido Paterno	9. Alfabético 10. 0<=letra<=20	11. Vacío 12. Numero o carácter especial 13. Letra >20

Tipo: Apellido Materno Tamaño: Apellido Materno	14. Alfabético 15. $0 \leq \text{letra} \leq 20$	16. Numero o carácter especial 17. Letra >20
Tipo: Fecha de nacimiento	18. Fecha (date)	19. Ninguno
Tipo: Matricula	23. Alfanumérico	25. Ninguno

Tabla 72 Caso de Prueba Caja Negra: Gestión Odontólogo

Casos de Prueba válidas

CP1						
CI	Nombre	Apellido P.	Apellido M.	Fecha	Teléfono	Matricula
71538249	Juan José	López	Araoz	10/07/2012	7123456	I-45

Cubre las clases de equivalencias válidas: 1-2-5-6-9-10-14-15-18-20-23

Casos de Pruebas Inválidas

CP2						
CI	Nombre	Apellido P.	Apellido M.	Fecha	Teléfono	Matricula
71538249	Juan José		Araoz	10/07/2012	7123456	I-45

Cubre las clases de equivalencias inválidas: 11

CP3						
CI	Nombre	Apellido1	Apellido2	Fecha	Teléfono	Correo
1234567834	Juan José	López	Araoz	10/07/2012	7123456	I-45

Cubre las clases de equivalencias inválidas: 4

Interfaz: Formulario ABM Gestión Tratamientos

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Tratamiento	1. Alfabético	3. Vacío
Tamaño: Tratamiento	2. $0 \leq \text{letra} \leq 35$	4. Carácter especial 5. $\text{letra} > 30$
Tipo: Costo	6. Real	7. Vacío
Tipo: Color	8. Alfanumérico	11. Vacío
Tamaño: Color		12. $\text{color} > 10$

Tabla 73 Caso de Prueba Caja Negra: Gestión Tratamientos

Casos de Prueba válidas

CP1		
Tratamiento	Costo	Color
Fisurado nivel 2	30	#001245

Cubre las clases de equivalencias válidas: 1-2-5-6-8

Casos de Pruebas Inválidas

CP2		
Tratamiento	Costo	Color
Fisurado nivel 2		#001245aa1

Cubre las clases de equivalencias inválidas: 7-12

CP3		
Tratamiento	Costo	Color
	30	

Cubre las clases de equivalencias inválidas: 3-11

Interfaz: Formulario ABM Gestión Área

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Área Tamaño: Área	1. Carácter = 1 2. carácter = S o R o D o O o C	3. Vacío
Tipo: Descripción Tamaño: Descripción	3. Alfabético 4. $1 \leq \text{Descripción} \leq 30$	5. descripción > 30 6. Vacío

Tabla 74 Caso de Prueba Caja Negra: Gestión Área

Casos de Prueba Válidas

CP1	
Área	Descripción
O	OPERATORIA DENTAL

Cubre las clases de equivalencias inválidas: 1-2-3-4

Casos de Prueba Inválidas

CP2	
Área	Descripción
O	

Cubre las clases de equivalencias inválidas: 6

Interfaz: Formulario ABM Gestionar Paciente

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: CI Tamaño: CI	1. Numero 2. 7	3. Letra o carácter especial 4. $\neq 7$

Tipo: Nombres Tamaño: Nombres	5. Alfabético 6. $1 \leq \text{Caracteres} \leq 20$	7. Vacío 8. Letra > 20
Tipo: Apellido Paterno Tamaño: Apellido Paterno	9. Alfabético 10. $0 \leq \text{letra} \leq 20$	11. Vacío 12. Numero o carácter especial 13. Letra > 20
Tipo: Apellido Materno Tamaño: Apellido Materno	14. Alfabético 15. $0 \leq \text{letra} \leq 20$	16. Numero o carácter especial 17. Letra > 20
Tipo: Fecha de nacimiento	18. Fecha (date)	19. Ninguno
Tipo: Matricula	23. Alfanumérico	25. Ninguno

Tabla 75 Caso de Prueba Caja Negra: Gestión Paciente

Casos de Prueba válidas

CP1						
CI	Nombre	Apellido P.	Apellido M.	Fecha	Teléfono	Matricula
71538249	Juan José	López	Araoz	10/07/2012	7123456	I-45

Cubre las clases de equivalencias válidas: 1-2-5-6-9-10-14-15-18-20-23

Casos de Pruebas Inválidas

CP2						
CI	Nombre	Apellido P.	Apellido M.	Fecha	Teléfono	Matricula
71538249	Juan José		Araoz	10/07/2012	7123456	I-45

Cubre las clases de equivalencias inválidas: 11

CP3						
CI	Nombre	Apellido1	Apellido2	Fecha	Teléfono	Correo
123456783 4	Juan José	López	Araoz	10/07/2012	7123456	I-45

Cubre las clases de equivalencias inválidas: 4

Interfaz: Formulario ABM Gestionar Usuario

Condición de Entrada	Clases Equivalentes Válidas	Clases Equivalentes Inválidas
Tipo: Password Tamaño: Password	1. Alfabético 2. $5 \leq \text{Caracteres} \leq 20$	3. Vacío 4. $\text{Caracteres} > 20$

Casos de Prueba Válidas

CP1
Password
11111

Cubre las clases de equivalencias válidas: 1-2

Casos de Prueba Inválidas

CP1
Password

Cubre las clases de equivalencias inválidas: 3

CP1
Password

123456789abcdesdfsdsafds

Cubre las clases de equivalencias inválidas: 4

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 de 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
<p>Jefe del Proyecto</p>	<p>Debe establecer un plan donde se debe contemplar los riesgos que sean identificados para el proyecto, adicionalmente dicho plan puede contener las descripciones, análisis, 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>

	Identificar y valorar preliminarmente los riesgos encontrados.
--	--

Tabla 76 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 77 Escala de Ponderación de los Riesgos

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	N°	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 reprobando 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

				conocimiento en el manejo del sistema.		utilizado o solo utilizado o en parte.		cerca de capacitación para realizarse en el presente o futuro. - *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. = *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. = 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	-	*Utilizar herramientas conocidas. - *Asegurar a las personas miembros del equipo de desarrollo	-	Realizar un control y seguimiento a las herramientas con las que se va a trabajar.

						do a 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	-	*Establecer con anticipación las interfaces y el entorno grafico a utilizar.	-	Realizar un control y seguimiento a las interfaces o pantallas que vallan de acuerdo a la facilidad del usuario.

						desagradables.		<p>☐*Hacer uso de herramientas que puedan otorgar una mejor apariencia a las interfaces.</p> <p>☐*Obtener conocimiento de buenas normas para el diseño de interfaces.</p>	
*	*	*	No llegar a la culminación	40%	La empresa queda	-	*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 .		<ul style="list-style-type: none"> ▢*Obtener información actualizada del coste del desarrolló proyectos similares: personal, recursos de hardware y otros. 	
	*		10	La institución no cuente con el Hardware y	20%	El sistema no pueda ser	-	*Aconsejar y plantear equipo de hardware y recursos de software	Realizar un control y seguimiento al software y hardware de la institución.

				Software necesarios.		implementado.		necesario para el funcionamiento del sistema con anticipación.		
*	*		11	El personal está enfermo y no disponible en momentos críticos en el desarrollo del proyecto.	40%	La empresa queda perjudicada y no se presente el proyecto en la fecha acordada	-	-	Buscar temporalmente 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 los	60%	El product o no	-	Revisar y hacer un análisis de	-	Realizar un control y seguimiento a

				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,	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		egresos e ingresos		
--	--	--	--	--	--	---	--	-----------------------	--	--

Tabla 78 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: Eduardo Luis Aguirre Gidiño

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 personal de los pacientes. 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 79 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 “IRACEMA Virtual” Sistema Informático para Mejoramiento del Control del historial Clínico Odontológico que se está desarrollando.

2.1.2.4.1.3 Alcance

Este plan se limita al proyecto de gestión “IRACEMA Virtual” Sistema Informático para Mejoramiento de Control de registro de historial clínico odontológico.

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 “IRACEMA Virtual”. 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 80 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 81 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 82 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 83 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:

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

Donde:

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 84 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:

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

Donde:

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 85 Factor de ponderación

Valores de Ajuste de Complejidad

0	1	2	3	4	5
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

Tabla 86 Valores de ajuste de complejidad

N°	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 87 Parámetros a evaluar

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

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

$$PF = 417.69$$

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 88 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:

$$\text{LDC} = 120 * 418$$

$$\text{LDC} = 50160$$

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:

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

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

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:

Donde: 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 89 Plan de las Fases

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

Descripción	Hito
Fase de Inicio	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

	<p>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>
<p>Fase de Elaboración</p>	<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 unasemana.</p>
<p>Fase de Construcción</p>	<p>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</p>

	<p>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.</p>
<p>Fase de Transición</p>	<p>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.</p>

Tabla 90 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.

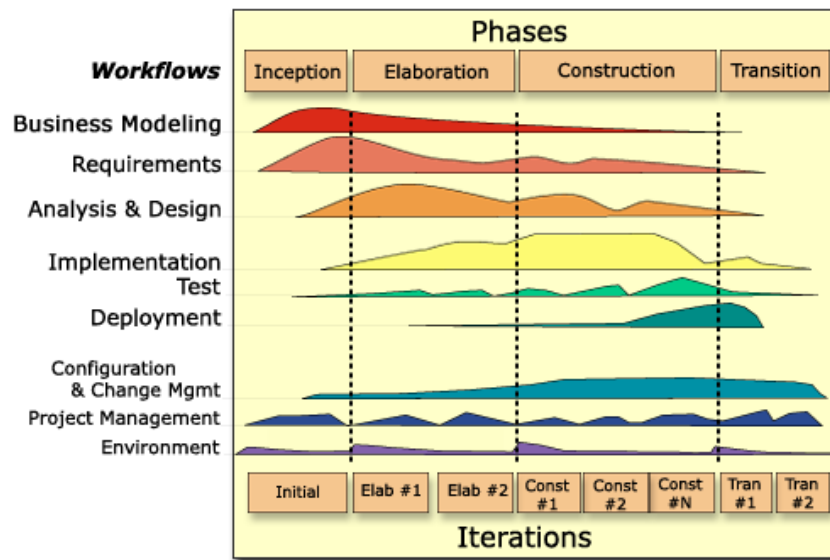


Figura 132 Calendario del proyecto

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 91 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 92 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	12-05-2015	27-06-2015
Elaboración	Primera	30-06-2015	04-07-2015
Construcción	Primera	07-07-2015	03-10-2015
Transición	Primera	06-10-2015	07-11-2015

Tabla 93 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 auditoria 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 auditorias intensivas.

Antes de la entrega del sistema, se realizan una auditoria funcional y una auditoria 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 prepara 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	- 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 94 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	- 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	- Documentación revisada.	Documentos de acuerdo a Estándar, y sin

Tabla 95 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	- Ninguna.	Documentos, plan de desarrollo de Software.
2	Documentación	- Monitorear adherencias de los documentos a los estándares.	Chequear documento desarrollado basado en la metodología RUP.
3	Diseño	- Monitorear adherencias del diseño a los estándares.	Chequear documento UML del sistema.
4	Codificación	- Monitorear adherencias de la codificación a los estándares.	Revisar de acuerdo a Patrones de Diseño MVC.
5	Métricas	- Revisar la métrica definida.	Revisar de acuerdo al estándar Puntos de Función y otros.
6	Criterios de Salida	- Proceso de Documentación revisado. - Proceso de Diseño revisado.	Discrepancias reportadas y solucionadas. Documentos de acuerdo a estándares.

Tabla 96 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 “IRACEMA Virtual” que ayuda a agilizar la centralización de la información de los historiales clínicos.

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 “IRACEMA Virtual” 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 Clínica Dental 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 gerente general y odontóloga Giovana Patricia Iñiguez Aramayo.
- 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 los historiales clínicos (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 ramas de la medicina, 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.