UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE ELECTROTECNIA Y COMPUTACION

TRABAJO MONOGRÁFICO

Sistema Web de Gestión de Pacientes Odontológicos

ODONTOWEB

PRESENTADO POR:

Br. Brayam Ariel Duarte

PARA OPTAR AL TÍTULO DE:

Ingeniero en Computación

TUTORA:

Ing. Flor de María Valle Izaguirre

Abril, 2017

DEDICATORIA

A mi familia

Quienes siempre estuvieron conmigo, ayudándome y animándome en las diversas situaciones que surgieron en el trayecto de mi formación académica. Este trabajo y todos mis logros son fruto de ese apoyo incondicional recibido por parte de ellos.

A mis amigos de la universidad

Que estuvieron ahí dándome ánimos y consejos, los cuales tomé de buena forma y los convertí en el sostén y pilar de perseverancia.

A mi tutora

Por ella llegue hasta aquí, sus buenas enseñanzas son y siempre serán un hito en mi carrera profesional, espero que se sienta orgullosa, porque este también es su logro.

AGRADECIMIENTOS

Agradezco primeramente al Creador

Si no fuera por Él y su ajuste fino, infinitamente preciso para dar vida al universo, nosotros como seres físicos no existiríamos en esta dimensión.

A mi familia le estoy sumamente agradecido

Por haberme apoyado, con las cosas mínimas y grandes. A mi madre, por su cariño y comprensión, siempre ha estado al pendiente de que no me falte nada.

Al Ing. Hubert Cross

Por haberme apoyado y asesorado en la implantación del sistema, además de haberme dado espacio en el centro de datos en su empresa. Gracias por la confianza y el apoyo.

A mi tutora Ing. Flor de María Valle

Por haberme acogido como alumno, me siento como un heredero de su conocimiento. También le agradezco por haberme brindado su amistad y paciencia, este logro es gracias a ella.

RESUMEN DEL TEMA

El presente trabajo monográfico titulado "***Sistema Web de Gestión de Pacientes Odontológicos,* ODONTOWEB**", tiene como principal objetivo facilitar las gestiones primordiales de las clínicas odontológicas, entre ellas: gestión de pacientes, citas, registro de odontogramas, proformas y pagos por servicios, entre otros, por medio de una aplicación web.

ODONTOWEB es una aplicación web orientada al sector de medicina dental nicaragüense, desarrollado en la herramienta Java Enterprise edition de Eclipse, la cual incluye extensiones para la creación de aplicaciones web de manera sencilla. También se utilizó HTML5 y CSS3 para la capa de vista del lado del cliente, utilizando la normativa ISO 9241 como estándar de diseño Web. Posee recargado asíncrono de objetos gracias a la implementación de AJAX utilizando javascript y jquery.

Este sistema fue ideado con la visión de ser integral, de manera que con solo registrarse y realizar algunas configuraciones básicas pueda ser utilizado por los odontólogos, desde sus clínicas o consultorios. Surgió como alternativa a varias herramientas que son ofertadas a precios y condiciones de uso que las hacen prohibitivas al entorno local.

La oportunidad de contar con información actualizada, gráfica – por medio del odontograma – y de acceso inmediato, es muy importante, tanto para los odontólogos como para los pacientes, dado que facilita la revisión del historial odontológico y el progreso en los tratamientos y trabajos dentales realizados.

El presente trabajo se divide en tres grandes apartados: el primero aborda los aspectos introductorios y teóricos, destacando UWE como metodología de desarrollo para aplicaciones web; el segundo apartado describe el análisis y presentación de resultados, documentando desde la especificación de requisitos hasta la implantación del sistema, y, por último, el tercer apartado, puntualiza conclusiones y recomendaciones finales, que se consideran útiles para trabajos futuros.

Tabla de contenido

Índice de Ilustraciones

Índice de Tablas

INTRODUCCIÓN

El presente documento monográfico muestra las etapas para el desarrollo de ODONTOWEB, una aplicación web orientada al sector de medicina dental nicaragüense. Dicha Web App está pensada para aquellos odontólogos interesados en utilizar estratégicamente las tecnologías de información para gestionar la información de sus pacientes, la programación de citas y registro de servicios odontológicos brindados, entre otros.

Se buscó desarrollar una herramienta simple, de fácil acceso y de alta calidad para ser utilizado por cualquier odontólogo perteneciente a una clínica y/o hospital.

La Web App permite el registro de atenciones y servicios para cada paciente, además de mostrar reportes estadísticos de los servicios brindados, el uso del producto software traerá beneficios cuantificables y no cuantificables para los interesados gracias a la facilitación de estrategias basadas en sistemas de información.

Como aspecto importante el sistema permite una agenda electrónica capaz de registrar eventos en una fecha dada para un paciente en específico. El complemento perfecto para un subsistema de agendas es la implementación de notificaciones al usuario con el fin de dar avisos a las actividades próximas a realizarse. El sistema puede enviar notificaciones de citas por medio del correo electrónico del paciente que el doctor haya registrado en el sistema web.

ODONTOWEB puede ser accedido por cualquier computador que se encuentre conectado a INTERNET, incluyendo la técnica “Responsiva” que permitirá una visualización de diseño más agradable.

JUSTIFICACIÓN

La creación de una aplicación web orientada al sector de medicina dental ayudará a extender el uso de los sistemas de información en la comunidad de odontólogos nicaragüenses, facilitando la forma de gestionar actividades propias a su giro de negocio, tales como el registro de pacientes, gestión de servicios odontológicos y la programación de citas, entre otras.

El sistema ODONTOWEB brindará valor agregado a las clínicas dentales, con la posibilidad de bajar los costos en la obtención de software de gestión y mejorar la calidad de sus Atenciones. El sistema es una forma de profesionalizar la gestión de una clínica, ya que con él se puede llevar un mejor control de los expedientes y servicios dentales brindados a sus pacientes.

Con ODONTOWEB, los odontólogos independientes tendrán la oportunidad de centralizar sus registros y agilizar sus actividades, disminuyendo el manejo y costo de almacenamiento, mantenimiento y traslado de expedientes o historiales del paciente (odontogramas), reduciendo así los riesgos de pérdida de documentos físicos. Así también les vendría muy bien a las clínicas que poseen varias sucursales gracias al manejo de información en la nube, estas clínicas podrían acceder a su información en cualquier parte del mundo con una conexión a internet.

ODONTOWEB permite bajar los costos de licencia dada su filosofía de software libre, es decir que los usuarios pueden adquirir el código fuente desde el repositorio GIT oficial del sitio, al igual que el script MySQL de la base de datos. Las clínicas que deseen implantar el sistema tendrán la facilidad de obtener la última versión del sistema y realizarle modificaciones, siempre y cuando se respete la licencia GNU GPL y derechos de autor.

OBJETIVOS

Objetivo general

Desarrollar un sistema web que brinde a los médicos odontólogos una eficiente aplicación para la planificación, programación y monitoreo de los servicios dentales.

Objetivos específicos

Que la aplicación web ODONTOWEB satisfaga los siguientes objetivos:

- ✓ Permitir el acceso al expediente electrónico e información de antecedentes clínicos del paciente al odontólogo desde cualquier dispositivo.
- ✓ Administrar un calendario como herramienta para la programación de citas odontológicas.
- ✓ Brindar informes de los servicios realizados a los pacientes.
- ✓ Proveer un Odontograma web para la esquematización de historial de servicios odontológicos del paciente.

MARCO TEÓRICO

Registros odontológicos

Odontograma como sistema de registro

El Odontograma es la herramienta que sirve como esquema utilizado por los odontólogos para el registro de información de la boca de una persona. En este diagrama el medico registra de forma física (lápiz y papel) la cantidad de piezas dentales permanentes que tiene el paciente, así como también sus estados (restaurados, extraídos, etc.).

De este modo el Odontograma, supone un registro de la historia clínica del paciente. Se trata, por lo tanto, de una herramienta de identificación. El odontólogo, al analizar el Odontograma de un paciente, puede saber qué trabajos se realizaron y establecer comparaciones entre el estado bucal actual y el registrado en la visita anterior.

El esquema del odontograma puede tener diferentes formatos. Hay versiones que identifican los dientes con números, otros con letras mayúsculas e incluso algunos con pares numéricos. Todo depende de la preferencia del odontólogo para elegir uno u otro formato. (Definicion.de, 2015)

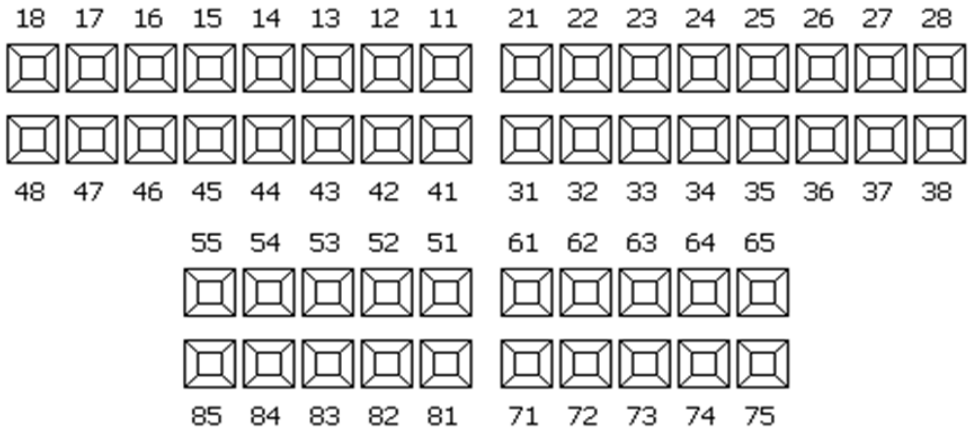


Ilustración 1 Odontograma Numérico

Expediente

Es un instrumento escrito que contiene antecedentes, exámenes, pruebas de laboratorio, diagnostico, pronostico, tratamientos y respuesta a los mismos. Es un

sistema por medio del cual se registran los datos convenientes para conocimiento del equipo de salud.

Este instrumento es de mucha importancia, ya que a partir de él se le da seguimiento a los servicios odontológicos realizados al paciente, es decir, para que un paciente pueda aplicar a un servicio es necesario que esté relacionado con la información contenida en un expediente médico odontólogo.

El expediente odontológico además de registrar información general del paciente, también puede detallar los tratamientos que se le han aplicado. A continuación, algunos de estos tratamientos que pueden estar presentes en el expediente.

- ✓ Endodoncia
- ✓ Periodoncia
- ✓ Ortodoncia
- ✓ Cirugía Maxilofacial

Diagnostico Odontológico

Sitio/Estado

El sitio/estado es utilizado para el diagnóstico y tratamiento de caries dental, según el modelo no invasivo de tratamiento, donde podemos encontrar en una pieza dentaria tres sitios de susceptibilidad a la caries dental y cinco estados de progresión de esta patología. (CAROL ANNE MURDOCH-KINCH, MARY ELLEN McLEAN, 2003)

El sitio/estado fue creado para un diagnóstico rápido de la caries dental, llevando a su respectivo tratamiento. Este nos estrega tres sitios donde se pueden encontrar la caries dental y cinco estados de progresión de la enfermedad. Lo cual nos dará una cifra que nos indicara que tipo de tratamiento debe ser realizado en aquel caso en particular.

Sitios

Sitio 1: Las lesiones cariosas se inician en surcos, fisuras y fosas de la cara oclusal, y superficies vestibulares y linguales/palatinas de todos los dientes; además de otros defectos de las caras libres de las coronas dentarias (por ejemplo: surcos por palatino/lingual en el cíngulo de incisivos).

Sitio 2: Las lesiones cariosas se inician en las superficies proximales de todos los dientes.

Sitio 3: Las lesiones cariosas se inician en la región cervical de todos los dientes, en sus caras palatina/lingual y vestibular.

Progresión de la caries dental

Estado 0: Caries activa sin cavitación. Intervención quirúrgica innecesaria.

Estado 1: Lesiones con alteraciones de la superficie que ha progresado hasta el punto donde la remineralización no es efectiva y un tratamiento restaurador es indicado.

Estado 2: Lesión moderada con cavitación localizada que ha progresado en la dentina sin debilitamiento de las cúspides dentarias.

Estado 3: Lesión que causa una extensa cavitación que ha progresado en la dentina debilitando las cúspides dentarias.

Estado 4: Lesión que ha progresado hasta el punto en el cual una o más cúpsides son destruidas.

Tratamiento

Cada estado de progresión de la caries dental posee un tratamiento específico:

Estado 0: Tratamiento de remineralización y/o sellantes de fosas y fisuras. Monitoreo en el tiempo de la detención o progresión de la enfermedad.

Estado 1: Preparación cavitaria mínimamente invasiva para una obturación adhesiva combinada con un tratamiento profiláctico de las superficies adyacentes.

Estado 2: Preparación mínimamente invasiva para una gran cavidad, combinada con un tratamiento profiláctico de las superficies adyacentes.

Estado 3: Preparación cavitaria para una restauración directa o indirecta que restablezca la función y preservar la resistencia de la unidad diente/restauración.

Estado 4: Extensa preparación cavitaria para la realización de una restauración indirecta que restablezca la función y para preservar la resistencia de la unidad diente/restauración.

Metodología de desarrollo web

“La World Wide Web y la internet que la alimentan son, posiblemente, los desarrollos más importantes en la historia de la computación. Estas tecnologías han llevado a todos (con cientos de millones más que eventualmente seguirán) a la era de la informática; además que se han convertido en la parte integral de la vida diaria en la primera década del siglo XXI.” (Pressman, 2006).

Los sistemas y aplicaciones basados en web ofrecen un completo catálogo de contenido y funcionalidades a la mayor cantidad de usuarios finales. La ingeniería web es el proceso de creación de WebApps de máxima calidad por medio de muchos paradigmas de desarrollo, tales como RUP y UWE, entre otros. Se tomara dentro de este proyecto los conceptos fundamentales de una metodología de desarrollo web para poder cumplir con los objetivos.

UML

El lenguaje unificado de modelado o UML es el sucesor de la oleada de métodos de análisis y diseño orientados a objetos (OOA&D) que surgió a finales de la década de 1980 y principios del siguiente. El UML unifica sobre todo los métodos de Booch, Rumbaugh (OMT) y Jacobson, pero su alcance es mucho más amplio.

UML es un lenguaje de modelado y no un método. La mayor parte de los métodos consiste, al menos en un lenguaje y en un proceso para modelar. El lenguaje de modelado es la notación de que se valen los métodos para expresar los diseños (Martin Fowler, 1999).

UWE

UWE UML (UML-Based Web Engineering) es una herramienta para modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos).

UWE es una propuesta basada en el proceso unificado y UML, pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

La metodología de desarrollo UWE consiste en una notación y en un método. La notación se basa en UML (OMG, 2003): para aplicaciones Web en general y para aplicaciones adaptativas en particular.

Así mismo el método consta de seis modelos mencionados a continuación:

- Modelo de casos de uso para capturar los requisitos del sistema.
- Modelo conceptual para el contenido (modelo del dominio).
- Modelo de usuario: modelo de navegación que incluye modelos estáticos y dinámicos.
- Modelo de estructura de presentación, modelo de flujo de presentación.
- Modelo abstracto de interfaz de usuario y modelo de ciclo de vida del objeto.
- Modelo de adaptación.

Fases de UWE

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas.

Las fases o etapas a utilizar son:

Captura, análisis y especificación de requisitos: En esta fase se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

Trata de diferente forma las necesidades de información, de navegación de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

Diseño del sistema: Se basa en la especificación de requisitos producido por la fase de análisis. El diseño define la estructura que debe darse a la aplicación web para satisfacer estos requisitos.

Codificación del software: Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

Pruebas: Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

La Instalación o Fase de Implantación: es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser utilizados por el usuario final.

Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

El Mantenimiento: es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control. (Galiano, 2012)

Modelo Vista Controlador (MVC)

Aplicación web

“Una aplicación web (web-based aplication) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador web) como el servidor web y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones” (Mora, 2002).

Las aplicaciones web han venido a incursionar en la industria del software, gracias a ellas contamos con acceso remoto a nuestros datos y nos ayuda a realizar funciones a distancia, por lo tanto las Web App son el medio de comunicación, entretenimiento y de negocios que más son explotados hoy en día, y sus capacidades aumentan a como aumentan las necesidades de los usuarios.

El cliente

El cliente web es un programa con el que interaccionan el usuario para solicitar a un servidor web de los recursos que desea obtener mediante HTTP. La parte cliente de las aplicaciones web suele estar formada por el código HTML que forma la página web más algo de código ejecutable realizado en lenguaje de script del navegador (JavaScript o VBScript) o mediante pequeños programas (applets) realizados en

Java. También se suelen emplear plug-ins que permiten visualizar otros contenidos multimedia (como Macromedia Flash), aunque no se encuentran tan extendidos como las tecnologías anteriores y plantean problemas de compatibilidad entre distintas plataformas. Por lo tanto, la misión del cliente web es interpretar las paginas HTML y los diferentes recursos que contienen (imagines, sonidos, etc).” (Mora, 2002)

El servidor

Servidor HTTP

“El servidor web es un programa que está esperando permanentemente las solicitudes de conexión mediante el protocolo HTTP por parte de los clientes web. En los sistemas Unix suele ser un “demonio” y en los sistemas Microsoft Windows un servicio” (Mora, 2002).

El sistema ODONTOWEB utilizará un servidor web basado en apache denominado Tomcat el cual funciona con aplicaciones desarrolladas Java JSP, la versión que se pretende desplegar es la 8.0 por ser la más estable hasta el momento. Existen otros tipos de servidores WEB desarrollados con apache como es el caso de WAMP el cual ejecuta aplicaciones en PHP entre otras.

Servidor MySQL

MySQL es un sistema gestor de base de datos relacional (SGBDR), que permite acceso a datos a velocidades bastantes rápidas. Es multitarea y por medio de consultas al servidor con el lenguaje de consultas SQL se pueden obtener grupos de datos específicos. Este gestor está programado en C++ y soporta muchas API’s como en Python, Java, PHP, etc. La utilidades del cliente y administración utiliza sockets TCP/IP, los sockets UNIX o los canales de con nombre NT (Named Pipes). (THIBAUD, 2006)

Herramientas para el desarrollo de aplicaciones web dinámicas

Java

Java es un lenguaje de programación de alto nivel que utiliza el paradigma de programación orientado a objetos, java está basado en lenguaje C y C++ por eso podemos observar similitudes en sus sintaxis.

La fase de controlador del ODONTOWEB se programará en java con la cual se realizarían las validaciones y se implementarían los Servlets para la gestión de parámetros de contexto (cliente-servidor) que utilizan clases totalmente heredadas de java.

J2EE

“La plataforma J2EE (Java 2 Enterprise Edition), es la propuesta de SUN para el desarrollo y las implementaciones corporativas multinivel.

La plataforma J2EE se apoya por completo en el lenguaje Java beneficiándose, por tanto, de sus características. SUN define al lenguaje Java como sencillo, orientado a objetos, seguro, portable multitarea. Estas características aparecen detalladas en el libro blanco de java, editado por SUN en su sitio Web.

J2EE es, ante todo, una norma que permite a terceras empresas desarrollar su propio servidor de aplicaciones, que puede implementar total o parcialmente bajo las especificaciones de SUN (conforme a dicha norma). (AUMAILLE, 2002)”

Se utilizó esta plataforma para la mejora de funcionalidades en la aplicación ODONTOWEB tanto como diseño además de mejoras en su contenedor de aplicaciones, ya que provee de herramientas ventajosas para el desarrollo de la Web App.

HTML

HTML en un le lenguaje de etiquetas que sirve para la creación de sitios y páginas web de forma esquematizada valiéndose de un estándar propuesto por la organización W3C, todo el contenido observable en una aplicación web está elaborada a base de HTML (texto, imágenes, videos, etc.).

Así mismo HMTL5 la cual podría decirse que no es una nueva versión mejorada de HTML sino que es un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red. (Gauchat, 2012). Esta debe ser integrada por herramientas de diseño y programación del lado del cliente como CSS y JavaScript para poder crear Web App de alta calidad.

CSS

Un archivo de estilos CSS (cascade style Sheet) es un grupo de reglas de formato que ayudaran a cambiar la apariencia de la Web App (por ejemplo, el tamaño y el color de texto). Sin estas reglas, el texto y cualquier otro elemento HTML seria mostrado en pantalla utilizando estilos (diseño) estándar provisto por el navegador. CSS es un lenguaje específico el cual permite modificar las etiquetas de HTML de forma que la Web App pueda ser personalizada en cuestiones de orden, posiciones de objetos en pantalla y colores. La última versión CSS3 también permite la incorporación de animaciones y efectos visuales.

Los estilos son reglas simples que requieren solo unas pocas líneas de código y pueden ser declaradas en el mismo documento. También es posible cargar las reglas CSS desde un documento externo (otro archivo) permitiendo organizar el documento principal, incrementar la velocidad de carga y aprovechar las nuevas características de HTML5. (Gauchat, 2012)

Ajax

Es un conjunto de técnicas para el desarrollo web que utilizan muchas tecnologías en el lado del cliente para crear aplicaciones Web asíncronas. Con Ajax, las aplicaciones web pueden enviar y recuperar datos de un servidor HTTP de forma asíncrona (en el fondo), sin interferir con la visualización y el comportamiento de la página existente, es decir que un componente de la página es actualizado automáticamente o por accionar del usuario, pero sin afectar los demás componentes.

HTML5 Canvas

Canvas o lienzo es un área de la pantalla compuesto por un mapa de bits de modo inmediato que puede ser manipulado con JavaScript, modo inmediato se refiere a la forma en que el canvas pone los píxeles en la pantalla. El canvas HTML vuelve a dibujar por completo la pantalla de mapa de bits en cada cuadro, mediante el uso de llamadas a la API de canvas de JavaScript.

Al utilizar el canvas el programador se encarga de crear la pantalla, antes de que se pinte cada fotograma de manera que sean insertados los píxeles correctos. El canvas básico de HTML5 incluye un contexto 2D, que permite a los programadores dibujar varias formas, renderizar texto y mostrar imágenes directamente en un área definida de la ventana del navegador.

Las opciones disponibles son básicamente aplicar colores; rotaciones; rellenos graduales; transparencias; manipulación de pixeles; y varios tipos de líneas, curvas, cajas y rellenos para aumentar formas, texto, e imágenes en un lugar del canvas (Fulton, 2013).

Estándares de diseño web

Un estándar es un conjunto de reglas normalizadas que describen los requisitos que deben ser cumplidos por un producto, proceso o servicio, con el objetivo de establecer un mecanismo base para permitir que distintos elementos hardware o software que lo utilicen, sean compatibles entre sí. (ISO 9241-151,2008)

Utilización de la normativa ISO 9241

La norma ISO 9241, contempla aspectos de diseño centrados en la comodidad del usuario tales como: universalidad, portabilidad y accesibilidad. Además, se definieron un conjunto de factores a considerar para el diseño y usabilidad de la aplicación ODONTOWEB, estos factores se enumeran a continuación:

- Proceso de diseño y evaluación
- Optimizando la experiencia de usuario
- Accesibilidad
- Navegación

- Encabezados, títulos y etiquetas
- Gráficos, imágenes y multimedia

Control de versiones

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión dando lugar a los llamados sistemas de control de versiones o VCS (del inglés Version Control System). Estos sistemas facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas (por ejemplo, para algún cliente específico). Ejemplos de este tipo de herramientas son entre otros: CVS, Subversion, SourceSafe, ClearCase, Darcs, Bazaar, Plastic SCM, Git, Mercurial, Perforce, Fossil SCM, Team Foundation Server.

Subversión

Subversión es un sistema de control de versiones de código abierto/gratuito (VCS). Es decir, subversión gestiona archivos y directorios, y los cambios Hecho a ellos, con el tiempo. Esto le permite recuperar versiones anteriores de sus datos o examinar el historial de revisiones. En este sentido, muchas personas piensan en un sistema de control de versiones como una especie de "máquina del tiempo".

Subversión puede operar a través de la red, lo que le permite ser utilizado por personas en diferentes equipos. Lo cual fomenta el trabajo en equipo, todos colaborando con ediciones y añadiendo nuevos archivos al repositorio.

El progreso puede ocurrir más rápidamente debido a que el trabajo esta versionado, muchas actualizaciones pueden llegar de diferentes fuentes o conductos sin inconvenientes. Si se llega a hacer un cambio incorrecto solamente es cuestión de regresar a una versión anterior y el cambio será desecho.

Algunos sistemas de control de versiones también son sistemas de gestión de configuración de software (SCM). Estos sistemas se adaptan específicamente para administrar los árboles de código fuente y tener muchas características que son específicas para el desarrollo de software, como la comprensión nativa, lenguajes de programación o herramientas de suministro de software de construcción. Subversión, sin embargo, no es uno de estos sistemas. Es un sistema que se puede utilizar para manejar cualquier colección de archivos. Para ti, esos archivos pueden ser código fuente, para otros, cualquier cosa desde listas de compras de comestibles a mezclas de vídeo digital y más allá. (Ben Collins-Sussman, 2011)

Licencia de software libre GNU GPL

La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License es la licencia más ampliamente usada en el mundo del software y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. Esta licencia fue creada originalmente por Richard Stallman fundador de la Free Software Foundation (FSF) para el proyecto GNU.

La GPL puede ser usada por cualquiera, ya que su finalidad es proteger los derechos de los usuarios finales (usar, compartir, estudiar y modificar), y otorgar a los beneficiarios de un programa de ordenador de los derechos de la definición de software libre. Bajo esta filosofía, la GPL garantiza a los destinatarios de un programa de computador los derechos y libertades reunidos en definición de software libre y usa copyleft para asegurar que el software está protegido cada vez que el trabajo es distribuido, modificado o ampliado. En la forma de distribución (solo pueden ser distribuidos bajo los términos de la misma licencia). (Smith, 2014)

ANÁLISIS Y PRESENTACIÓN DE RESULTADOS

Especificación de requisitos del sistema

Requisitos funcionales del Sistema

La **Tabla 1 Requisitos funcionales** , muestra y detalla cada requisito funcional.

Id Requisito	Nombre del Requisito funcional	Descripción	Nivel de prioridad
RF-1	Registro de Clínica	El usuario podrá crear cuenta de usuario para clínica	5
RF-2	Editar cuenta de usuario	Los usuarios editaran su datos (información personal) registrada en el sitio	3
RF-3	Dar de baja a usuario	El usuario administrador podrá dar de baja a los demás usuarios	3
RF-4	Dar de alta un usuario	El usuario administrador podrá dar de alta a los usuarios con estado inactivo	3
RF-5	Registro de expediente	El sistema presentara una plantilla detallada para el registro de la información del paciente. Posteriormente será en el expediente donde se almacenan los servicios que serán aplicados a los pacientes	5
RF-6	Búsqueda y lista de pacientes	El odontólogo podrá realizar búsquedas de pacientes o podrá listarlos para una mejor gestión	2
RF-7	Dar de baja un expediente	El médico podrá pasar el estado de un paciente a inactivo	2
RF-8	Ver historial de servicios	Los usuarios podrán ver los servicios que se le han aplicado al paciente en todas sus atenciones	3
RF-9	Crear cuenta para usuarios de una clínica	El usuario (Administrador de clínica) podrá generar cuentas de otros usuarios para un odontólogo, o asistente que requiera cuenta en el sistema	3
RF-10	Registrar cita	Se incluirá un calendario como herramienta para que el usuario pueda planificar citas	3
RF-11	Notificación de citas próximas	El sistema brindará notificaciones de citas próximas. Esta notificación podría ser enviada al	2

Id Requisito	Nombre del Requisito funcional	Descripción	Nivel de prioridad
		correo del paciente asociado, del odontólogo, admón. de clínica o asistente; según como lo especifique el usuario.	
RF-12	Registro de catálogos de servicios	El usuario tendrá la opción de registrar sus catálogos de recursos y servicios	3
RF-13	Registro de atención	Al momento de registrar una atención de un paciente el odontólogo podrá disponer de un Odontograma para definir los servicios a realizar	5
RF-14	Registrar cotizaciones	El sistema podrá registrar cotizaciones de un paciente, estas cotizaciones podrán ser cargadas como atenciones futuras	2
RF-15	Generar atención a partir de cotización	El usuario podrá transformar una cotización, en una proforma de atención.	2
RF-16	Crear crédito o financiamiento a paciente	El usuario tendrá la opción de crear créditos a sus pacientes.	4
RF-17	Listar, modificar créditos de pacientes	El usuario podrá ver todos los créditos asignados a paciente, además de poder modificarlos	3
RF-18	Registro abonos	El usuario registrará abonos que afecten el saldo corriente de un paciente con crédito	4
RF-19	Mostrar estados de cuentas y enviarlos por correo	El usuario podrá ver los estados de cuenta de cada paciente, además tendrá la opción de enviárselos por correo	3
RF-20	Generar reportes	El odontólogo podrá generar reportes con respecto a los siguientes componentes: • Expedientes • Servicios brindados • Citas programadas • Estado de cuentas de pacientes • Proforma paciente de paciente • Cotización	3

Tabla 1 Requisitos funcionales

Prioridad	Descripción
1	Cuestionable
2	Útil
3	Importante
4	Critico
5	Obligatorio

Tabla 2 Niveles de prioridad

Requisitos no funcionales

Los requisitos no funcionales están subdivididos en 4 criterios: Seguridad, Diseño, Interoperabilidad y Admón. de base de datos. A continuación, una breve descripción:

Categoría	Id	Nombre	Descripción
Seguridad	RSeg-1	Acceso por login y Password	Los usuarios registrados al sistema accederán a través de un login y Password
	RSeg-2	Cifrar contraseña	El sistema contará con un algoritmo de encriptación para las contraseñas de los usuarios, esto permitirá mayor seguridad en el acceso
	RSeg-3	Solicitud de Registro al sistema	Los usuarios generaran solicitudes de registros, el sistema remitirá un correo al usuario para validar que sea un correo valido y que pueda proseguir con el inicio de sesión. Las solicitudes pasaran a un estado de caducadas si el usuario no la confirma en un plazo de 7 días
Interfaz	RD-1	Guía de Diseño	ODONTOWEB, será diseñado siguiendo los lineamientos de la normativa ISO 9241
	RD-2	Técnica responsiva	El sistema podrá adaptarse para ser mostrado por dispositivos de diferentes tamaños en resolución

Interoperabilidad	RO-1	Implementación operable	El sistema deberá de ser visualizado y operable principalmente en los navegadores google chrome, Opera y firefox
Diseño y Administración de base de datos	RGBD-1	Sistema gestor de Base de Datos	La base de datos del sistema, deberá ser administrada por la herramienta MySQL Server, que es una herramienta Open Source, y de bajo costo para la elaboración y mantenimiento de la base de datos

Tabla 3 Requisitos no funcionales

Especificación de casos de usos

En los casos de uso se pretende plasmar la interacción de cada uno de los usuarios (denominados actores) con los módulos o funcionalidades específicas del sistema. Por consiguiente, la especificación de los casos de uso es de mucha importancia, ya que con ella se puede conseguir trazabilidad con respecto las responsabilidades de un actor y asociarlas con paquetes de requerimientos de la aplicación, además de ayudar en la representación de entregables basados en UML descritos por UWE.

En la siguiente tabla, se detallan los actores principales del sistema:

Actor	Descripción
Usuario no registrado	Es el usuario que tiene acceso al sitio web, pero no ha adquirido una cuenta de acceso. Este usuario es quien envía la solicitud de registro por primera vez al sistema
Administrador	Es el usuario que tiene los privilegios de administrador del sistema, es capaz de gestionar a los usuarios registrados además de utilizar sus funcionalidades
Admón. Clínica	El administrador de la clínica es el encargado de generar otras cuentas de usuarios para una clínica en específico, además de gestionar los catálogos
Odontólogo	Es el usuario catalogado como odontólogo, el cual tendrá acceso a funciones propias de la gestión médica odontológica
Asistente Odontológico	El asistente será creado por el odontólogo. El asistente tendrá la función de ayudar con la gestión al odontólogo, por ende él puede utilizar funciones como la creación de citas y registro de pacientes, entre otras

Actor	Descripción
Paciente	El paciente recibe notificaciones por correo electrónico (recordatorio de citas, estados de cuenta, etc.) generados por el sistema

Tabla 4 Actores del sistema

A continuación, se muestra el nivel de jerarquía de cada uno de los usuarios que tienen interacción con el sistema:

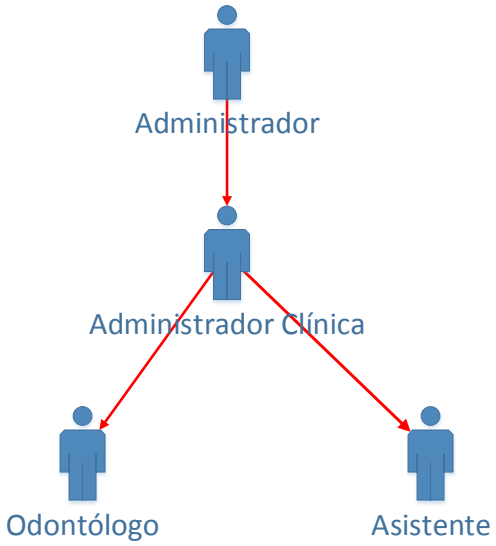


Ilustración 2 Jerarquía de usuarios

Detalle de los Casos de Uso

Una vez definidos quienes serán los actores principales y sus jerarquías en el sistema, a continuación, serán mostrados el detalle y especificación de cada caso de uso utilizando la siguiente tabla para representarlos:

Caso de Uso:	Nombre completo del caso de uso
Id	***Identificador único de cada caso de uso***
Actores	Se especifican los usuarios que tendrán interacción con el caso de uso
Propósito	Se detalla el objetivo para la implementación del caso del uso dentro de la implementación del sistema, su importancia y razón de ser, la cual podría estar ligada a un requerimiento estratégico para el sistema o requerimiento no funcional
Resumen	Se describe el caso de uso, sus componentes importantes e impacto del mismo sobre el sistema
Precondiciones	Son las condiciones para que el caso de uso pueda ser ejecutado

Flujo normal	Lista de tareas necesarias para la realización del caso del uso en el sistema, las cuales pueden surgir a partir del accionar del actor. El símbolo← representa las salidas del sistema y el símbolo → representa las entradas provenientes del actor. El símbolo ← se lee "el sistema" y → se lee "el actor/usuario"
Excepciones	Son las posibles salidas del sistema, las cuales resultan si el usuario decide no seguir con el flujo normal del caso de uso
Post Condiciones	Estado final del sistema después del cumplimiento en forma correcta de un caso de uso

Tabla 5 Plantilla de especificación de casos de usos

Caso de Uso:	***Solicitud de cuenta de usuario***	
Id	***CDU1***	
Actores	Usuario no registrado	
Propósito	Registrarse en ODONTOWEB para obtener cuenta de usuario	
Resumen	Permite al usuario generar una solicitud para una cuenta de usuario, con la cual podrá tener acceso a la gestión odontológica	
Precondiciones	Ingresar al url del sitio desde un navegador	
Flujo normal	1. → Accede a la opción de creación de nueva cuenta 2. ← Despliega el formulario de ingreso 3. → Completa el formulario de ingreso 4. → Envía la solicitud 5. ← Despliega ventana de confirmación 6. ← Envía correo al usuario para confirmar nueva cuenta 7. → Confirma cuenta por medio de correo electrónico	
Excepciones	**Paso**	**Acción**
	3	El usuario no registra datos obligatorios (con *):
		← Despliega mensaje de error en el formulario (E1)
	4	El usuario cancela el formulario:
		E2.El sistema retorna a la página principal
	5	El usuario cliquea en "no enviar" sobre el mensaje de confirmación:
		E3. El sistema retorna al formulario de ingreso
	7	El Usuario ignora el correo del sistema:
		E4. La cuenta queda inactiva
Post Condiciones	1. Cuenta de usuario es activada	

Tabla 6 CU: Solicitud de cuenta de usuario

Caso de Uso	**Dar de baja a usuario registrado**
Id	**CDU2**
Actor(es)	Administrador, Admón. Clínica
Propósito	Dar de baja un usuario registrado para cambiar su estado a inactivo.

Resumen	El usuario tiene la opción de pasar el estado de los demás usuarios a inactivo, esto no borrara la cuenta, simplemente cambiara su estado registrado en la base de datos. La sesión del odontólogo se cierra al cambiar su estado.
Precondiciones	El usuario debe estar registrado al sistema y con estado activo.
Flujo normal	1. → Entrar a los catálogos, listar usuarios 2. →Ingresa a la información de la cuenta 3. ←Despliega la información en controles editables 4. →Selecciona lista de estados de la cuenta y lo coloca como inactivo 5. →Presiona en guardar cambios 6. ←Muestra mensaje de confirmación 7. →Presiona en aceptar dentro del mensaje de confirmación
Excepciones	**Paso** | **Acción** 6 | Usuario presiona en cancelar: E1. El sistema no guarda cambios y retorna a la información del usuario
Post Condiciones	1. El usuario quedara en estado inactivo. 2. Se cerrara la sesión del odontólogo

Tabla 7 CU: Dar de baja a usuario registrado

Caso de Uso	**Dar de alta a usuario inactivo**
Id	**CDU3**
Actor(es)	Administrador, Admón. Clínica
Propósito	Activar cuenta de usuario
Resumen	El usuario tiene la opción de volver a habilitar las cuentas inactivas
Precondiciones	El usuario debe de estar registrado al sistema y con estado inactivo.
Flujo normal	1. →Entrar a catálogos, listar usuarios inactivos 2. →Seleccionar usuario y entrar a modificar información 3. ←El sistema muestra información de usuario 4. →Cambiar estado a activo 5. →Guardar cambios
Excepciones	**Paso** | **Acción** 3 | Usuario cancela la edición de cuentas E1. El sistema retorna a la ventana anterior
Post Condiciones	6. El usuario quedara en estado activo.

Tabla 8 CU: Dar de alta a usuario inactivo

Caso de Uso	**Editar información medico registrado**
Id	**CDU4**
Actor(es)	Admón. clínica, Odontólogo

<table>
<tr><td>Propósito</td><td>Actualizar o corregir información personal del médico odontólogo dentro del sistema</td></tr>
<tr><td>Resumen</td><td>El usuario tiene la posibilidad de corregir cierta información de la cuenta por motivos de actualización tales como: correo, nombre, contraseña, etc.</td></tr>
<tr><td>Precondiciones</td><td>El odontólogo debe estar registrado y activo en el sistema</td></tr>
<tr><td>Flujo normal</td><td>1. →Entrar a la información del perfil
2. ←Muestra información de usuario
3. →Edita información en el formulario
4. →Presiona “Guardar cambios”
5. ← Muestra ventana de confirmación
6. →Presiona aceptar</td></tr>
<tr><td>Excepciones</td><td>
<table>
<tr><th>Paso</th><th>Acción</th></tr>
<tr><td rowspan="2">4</td><td>El odontólogo presiona en cancelar:</td></tr>
<tr><td>E1. El sistema retorna a la página anterior</td></tr>
<tr><td rowspan="2">5</td><td>El odontólogo presiona cancelar:</td></tr>
<tr><td>E2. El sistema retorna a la información del usuario sin guardar cambios</td></tr>
</table>
</td></tr>
<tr><td>Post Condiciones</td><td>1. Nueva información almacenada en el sistema
2. Registro de fecha última modificación</td></tr>
</table>

Tabla 9 CU: Editar información medico registrado

<table>
<tr><th>Caso de Uso</th><th>Registro de expedientes</th></tr>
<tr><td>Id</td><td>CDU5</td></tr>
<tr><td>Actor(es)</td><td>Admón. clínica, Odontologo,Asistente</td></tr>
<tr><td>Propósito</td><td>Crear un expediente de un paciente el cual será registrado por primera vez.</td></tr>
<tr><td>Resumen</td><td>El usuario digita la información personal de un paciente, además de otro tipo de información médica necesaria para aplicar tratamientos como pueden ser alergias u enfermedades crónicas.</td></tr>
<tr><td>Precondiciones</td><td>1. El usuario debe ingresar al registro de pacientes (el cual se puede acceder desde el menú inicial)
2. El paciente no debe existir para ese medico</td></tr>
<tr><td>Flujo normal</td><td>1. →Ingresar al registro de pacientes en el menú principal
2. ←muestra el formulario de registro
3. →Registra la información del nuevo paciente en el formulario
4. →Presiona en guardar</td></tr>
<tr><td>Excepciones</td><td>
<table>
<tr><th>Paso</th><th>Acción</th></tr>
<tr><td rowspan="2">4</td><td>El usuario presiona en cancelar:</td></tr>
<tr><td>E1. El sistema regresa al menú principal.</td></tr>
</table>
</td></tr>
<tr><td>Post Condiciones</td><td>Expediente creado</td></tr>
</table>

Tabla 10 CU: Registro de expedientes

Caso de Uso	Actualizar información de paciente	
Id	**CDU6**	
Actor(es)	Admón. Clínica, Odontólogo, Asistente	
Propósito	Actualizar la información registrada en el sistema para un paciente en especifico	
Resumen	El usuario tiene la posibilidad de actualizar cierta información del paciente tales como: correo, nombre, dirección, etc.	
Precondiciones	El paciente debe estar registrado en el sistema	
Flujo normal	1. →Ingresa en la lista de pacientes y seleccionar actualizar información 2. ←Mostrará la información del paciente en controles editables 3. →Registra cambios en el formulario 4. →Guarda cambios 5. ←Muestra mensaje de confirmación 6. →Presiona en aceptar	
Excepciones	**Paso**	**Acción**
	4	El usuario presionó en cancelar:
		E1. El sistema retornara a la lista de pacientes
	6	El odontólogo presiona en cancelar:
		E2. No se guardan cambios
Post Condiciones	1. Información del sistema actualizada 2. Registro de fecha última modificación	

Tabla 11 CU: Actualizar información de paciente

Caso de Uso	Crear crédito a paciente	
Id	**CDU7**	
Actor(es)	Admón. Clínica, Asistente	
Propósito	El usuario puede generar cuenta de crédito, para los pacientes	
Resumen	El usuario puede crear una cuenta de crédito para los pacientes que desean ser financiados, los créditos formaran parte del estado de cuenta de un paciente	
Precondiciones	1. El paciente debe estar registrado en el sistema	
Flujo normal	1. →Entra al menú de Atención 2. →Seleccionar crear crédito 3. ←muestra formulario de nuevo crédito 4. →selecciona cliente asignado al crédito 5. ←Ingresar información del crédito 6. → da clic en guardar 7. ← muestra mensaje de confirmación 8. →Acepta confirmación	
Excepciones	**Paso**	**Acción**
	8	El usuario selecciona cancelar
		E1. El sistema no genera cuenta de crédito

Post Condiciones	Ninguna

Tabla 12 CU: Crear crédito a paciente

Caso de Uso	**Registrar Abono**	
Id	**CDU7**	
Actor(es)	Admón. Clínica, Asistente	
Propósito	Registrar los abonos de un paciente a su estado de cuenta.	
Resumen	El usuario puede registrar abonos para saldar el monto de su crédito hasta pagarlo completamente.	
Precondiciones	1. El paciente debe estar registrado 2. El paciente debe poseer una crédito con saldo vigente	
Flujo normal	1. →Entra al menú de Atención 2. →Seleccionar Registrar Abono- crédito 3. ←muestra formulario de registro Abono 4. →selecciona cliente asignado y el crédito 5. ←Ingresar información del Abono 6. → da clic en guardar 7. ← muestra mensaje de confirmación 8. →Acepta confirmación	
Excepciones	**Paso**	**Acción**
	8	El usuario selecciona cancelar
		E1. El sistema no genera registro de abono
Post Condiciones	Saldo de cuenta corriente es actualizado	

Tabla 13 CU: Registrar Abono

Caso de Uso	**Ver estado de cuenta paciente**	
Id	**CDU7**	
Actor(es)	Admón. Clínica, Asistente	
Propósito	Ver información de estados de cuenta	
Resumen	El usuario puede consultar los movimientos que se realizaron en el estado de cuenta de un paciente	
Precondiciones	1. El paciente debe tener créditos asociados	
Flujo normal	1. →Entra al menú de Atención 2. →Seleccionar ver estado de cuenta paciente 3. ->Seleccionar paciente y crédito 4. →clic en visualizar	
Excepciones	**Paso**	**Acción**
	8	El usuario seleccionado no posee cuenta de crédito
		E1. El sistema mostrara mensaje de error
Post Condiciones	Ninguna	

Tabla 14 CU: Ver estado de cuenta paciente

Caso de Uso	Ver historial del paciente	
Id	**CDU8**	
Actor(es)	Admón. Clínica, Odontólogo,Asistente	
Propósito	Ver los servicios que se le han brindado a un paciente	
Resumen	El usuario del sistema podrá ver el historial de un paciente en cualquier momento que lo desea siempre que busque o liste a los pacientes anteriormente	
Precondiciones	1. El paciente debe estar registrado en el sistema 2. El historial debe contener información de atenciones realizadas	
Flujo normal	1. →Entra a la lista de pacientes 2. ←Se muestra la lista de pacientes 3. →Dar clic sobre ver historial 4. →Selecciona rango de fechas 5. ←Muestra los servicios efectuados en lapso de tiempo definido por el usuario	
Excepciones	**Paso**	**Acción**
	4	El usuario selecciona una fecha invalida
		E1. El sistema muestra mensaje de error
	3	El usuario selecciona un paciente sin historial
		E2. El sistema muestra mensaje de error
Post Condiciones	Ninguna	

Tabla 15 CU: Ver historial del paciente

Caso de Uso	Cambiar estado a paciente	
Id	**CDU9**	
Actor(es)	Administrador, Admón. Clínica, Asistente	
Propósito	Cambiar el estado de paciente registrado, para cambiar su estado a inactivo.	
Resumen	El usuario tiene la opción de pasar el estado de un paciente a inactivo o activo dentro del sistema, esto no borrara su expediente.	
Precondiciones	El paciente debe estar registrado en el sistema	
Flujo normal	1. →Entra a la lista de pacientes y dar clic sobre actualizar información 2. ←Mostrará la información del paciente y el estado en una lista desplegable 3. →Modifica estado a Activo/Inactivo 4. → Presiona en guardar 5. ← muestra una ventana de confirmación 6. →Clica en Aceptar	
Excepciones	**Paso**	**Acción**
	4	El usuario presiona en cancelar:
		E1. El sistema retorna a la página anterior
	6	El usuario presiona en cancelar:
		E2. El sistema no guarda cambios
Post Condiciones	1. Estado de paciente cambia a activo/inactivo	

	2. Si el paciente está inactivo sus citas y tratamientos quedan inactivas de la misma manera

Tabla 16 CU: Cambiar estado a paciente

Caso de Uso	**Registrar cita**	
Id	**CDU10**	
Actor(es)	Administrador, Admón. Clínica, Odontólogo	
Propósito	Programar una cita para un paciente	
Resumen	El usuario podrá registrar una nueva cita con un odontólogo de acuerdo a su horario de atención	
Precondiciones	1. El paciente debe existir en la base de datos 2. No debe existir una cita el mismo día y a la misma hora (incluyendo tiempo de duración de la atención odontológica)	
Flujo normal	1. →Ingresa en gestión de citas del menú principal 2. ←Despliega el formulario de registro 3. →Registra la hora y fecha de la cita 4. →Registra comentario breve 5. →Clica en Guardar	
Excepciones	**Paso**	**Acción**
	5	El usuario presiona cancelar:
		E1. La cita no es registrada E2. Retorna al menú principal
Post Condiciones	La cita será registrada en el sistema como programada	

Tabla 17 CU: Registrar cita

Caso de Uso	**Reprogramar cita**	
Id	**CDU11**	
Actor(es)	Administrador, Admón. Clínica, Odontólogo	
Propósito	Reprogramar una cita para un paciente	
Resumen	El usuario tiene la opción de reprogramar una cita activa, para cambiar la hora y fecha de atención del paciente	
Precondiciones	1. La cita debe existir 2. La reprogramación no debe afectar otras citas programadas 3. No se puede reprogramar citas canceladas	
Flujo normal	1. →Ingresar a la de gestión de citas en el menú principal 2. →Lista las citas programas 3. ←Muestra el calendario de citas 4. →Selecciona reprogramación de cita 5. →Registra la nueva fecha y hora 6. →Da en guardar 7. ←Mostrará ventana de confirmación 8. →Da clic en aceptar	
Excepción	**Paso**	**Acción**
	6	El usuario cancela la reprogramación:

		E1. El sistema retorna a la página principal
	8	El usuario presiona cancelar
		E2. No se guarda la reprogramación
Post Condiciones	La fecha y hora de la cita es actualizada en el sistema	

Tabla 18 CU: Reprogramar cita

Caso de Uso	**Cancelar cita**	
Id	**CDU12**	
Actor(es)	Administrador, Admón. Clínica, Odontólogo	
Propósito	cancelar una cita programada para un paciente	
Resumen	El usuario tiene la opción de cancelar una cita activa.	
Precondiciones	La cita debe estar en estado programada	
Flujo normal	1. →Ingresar a la gestión de citas 2. →Lista citas 3. ←Muestra listado de citas programadas 4. →Clica en cancelar cita 5. ←Muestra ventana de confirmación 6. →Acepta la confirmación de cancelación	
Excepciones	**Paso**	**Acción**
	6	El usuario presiona cancelar:
		E1. El sistema retorna al listado de citas
Post Condiciones	La cita quedara cancelada en el sistema	

Tabla 19 CU: Cancelar cita

Caso de Uso	**Notificación de cita**	
Id	**CDU13**	
Actor(es)	Odontólogo, Paciente	
Propósito	Notificar a los usuarios el cambio de estado de una cita	
Resumen	El sistema notificara vía correo electrónico citas programadas, reprogramadas y canceladas a los usuarios registrados (odontólogo, paciente)	
Precondiciones	Modificación del estado de una cita	
Flujo normal	1. →Entra a la gestión de citas del menú principal 2. →Crea cita o modificar el estado de una cita existente 3. ←Generará un correo el cual llegara al usuario indicando la cita	
excepciones	**Paso**	**Acción**
	3	El correo no llega al usuario:
		E1. El usuario perdió acceso a su cuenta de correo electrónico E2. El correo electrónico de un usuario está registrado incorrectamente en la base de datos
Post Condiciones	Los usuarios recibirán correos del cambio efectuado	

Tabla 20 CU: Notificación de cita

<table>
<tr><th>Caso de Uso</th><th>Registro de servicios</th></tr>
<tr><td>Id</td><td>CDU14</td></tr>
<tr><td>Actor(es)</td><td>Admón. Clínica</td></tr>
<tr><td>Propósito</td><td>Registrar el catálogo de servicios para una clínica</td></tr>
<tr><td>Resumen</td><td>El usuario podrá registrar su propio catálogos de servicios personalizados, precios y descripciones</td></tr>
<tr><td>Precondiciones</td><td>1. Tener cuenta de usuario
2. El usuario requiere los permisos de administradores de clínica</td></tr>
<tr><td>Flujo normal</td><td>1. →Entra a la gestión de catálogos de servicios y recursos en el menú principal
2. →Crea un nuevo catálogo de servicios
3. ←Cargara los servicios existentes
4. →Selecciona servicios o crear nuevos con estado activos
5. →Ingresa precios de cada servicio
6. →Clica en guardar cambios</td></tr>
<tr><td>Excepciones</td><td>
<table>
<tr><th>Paso</th><th>Acción</th></tr>
<tr><td rowspan="2">6</td><td>El usuario presiona cancelar:</td></tr>
<tr><td>E1. El sistema retorna al menú principal</td></tr>
</table>
</td></tr>
<tr><td>Post Condiciones</td><td>El catalogo para el medico ha sido creado</td></tr>
</table>

Tabla 21 CU: Registro de servicios

<table>
<tr><th>Caso de Uso</th><th>Remitir paciente para atención</th></tr>
<tr><td>Id</td><td>CDU15</td></tr>
<tr><td>Actor(es)</td><td>Admón. Clínica, Asistente</td></tr>
<tr><td>Propósito</td><td>Remitir pacientes desde la recepción hasta el odontólogo especifico</td></tr>
<tr><td>Resumen</td><td>El asistente podrá registrar al paciente en la agenda inmediata (cola de atención) del odontólogo para que este pueda atenderlo, esto también funciona para validar las citas programadas del día.</td></tr>
<tr><td>Precondiciones</td><td>El asistente debe tener una cuenta activa</td></tr>
<tr><td>Flujo normal</td><td>1. → Ingresar a la búsqueda de expediente
2. →Si el usuario no posee expediente proceder a crear (Caso de uso crear expediente)
3. →Una vez creado, el asistente creara una nueva cita para el paciente (Caso de uso registrar cita)
4. ← Mostrará formulario de registro de cita
5. → selecciona el odontólogo existente en el circulo
6. → Registrar fecha, hora y descripción de cita
1. → Clic en guardar</td></tr>
<tr><td>Excepciones</td><td>
<table>
<tr><th>Paso</th><th>Acción</th></tr>
<tr><td rowspan="2">4</td><td>El usuario presiona cancelar:</td></tr>
<tr><td>E2. Se mostrara ventana de confirmación para la acción</td></tr>
<tr><td rowspan="2">7</td><td>El odontólogo presiona en cancelar</td></tr>
<tr><td>E3. No se guardan los cambios y retorna al menú</td></tr>
</table>
</td></tr>
</table>

Post Condiciones	1. Se actualizara la agenda del odontólogo

Tabla 22 CU: Remitir paciente para atención

Caso de Uso	**Registrar atención odontológica**
Id	**CDU16**
Actor(es)	Admón. Clínica, Odontólogo
Propósito	Registrar las atenciones y servicios prestados a un paciente por medio de un Odontograma web
Resumen	El usuario podrá registrar la atención y servicios odontológicos que se le han aplicado a un paciente y al mismo tiempo dichos servicios pasaran a ser parte del historial del paciente
Precondiciones	El paciente debe estar registrado y activo
Flujo normal	1. → Entra a la opción registro de atención a pacientes 2. ← Muestra las opciones del menú 3. → Selecciona Mostrar Agenda 4. ← Muestra citas en cola 5. →Seleccionar cita y clic en ejecutar 6. ← Despliega el Odontograma 7. → Registra servicios a realizar en el Odontograma 8. → Genera cotización de la atención 9. → Seleccionar si cargará a plan de pago por cita 10. → Presiona la opción de “registrar Atención” 11. ←Se mostrará ventana de confirmación 12. →Dar clic en aceptar
Excepciones	**Paso** / **Acción** **6** / El odontólogo presiona cancelar: E2. Se mostrará ventana de confirmación para la acción **12** / El odontólogo presiona en cancelar E3. No se guardan los cambios y retorna al Odontograma
Post Condiciones	2. Se guardará en la base de datos todos los servicios aplicados en el Odontograma para el paciente, de manera histórica clínica. 3. Si el usuario seleccionar la opción de cargar a plan de pago se generara un crédito para el paciente

Tabla 23 CU: Registrar atención odontológica

Caso de Uso	**Generar reportes**
Id	**CDU17**
Actor(es)	Administrador, Admón. Clínica, Odontólogo, Asistentes, Paciente
Propósito	Generar reportes sobre todos los módulos del sistema

<table>
<tr><td>Resumen</td><td>El odontólogo puede generar reportes acerca su gestión de pacientes, tales como: servicios, atenciones, tratamientos, citas.</td></tr>
<tr><td>Precondiciones</td><td>Ingresar en el sistema</td></tr>
<tr><td>Flujo normal</td><td>1. →El usuario debe de entrar al módulo de reportes del sistema
2. ←El sistema mostrara lista de reportes existentes
3. →El odontólogo debe presionar la opción de “generar reporte” para visualizarlo</td></tr>
<tr><td>Excepciones</td><td><table><tr><th>Paso</th><th>Acción</th></tr><tr><td>3</td><td>Los parámetros del reporte no son válidos:</td></tr><tr><td></td><td>E1. El reporte es cargado sin datos
E2.</td></tr></table></td></tr>
<tr><td>Post Condiciones</td><td>Generación de reporte</td></tr>
</table>

Tabla 24 CU: Generar reportes

Diagramas de caso de uso

Una vez especificado los casos de uso es importante denotar la información con diagramas UML, según como lo detalla la metodología UWE en su etapa de modelado del análisis de requisitos. En los siguientes diagramas podremos observar la interacción de los actores del sistema denotados por nivel de usuario, además de la jerarquía de actividades que se deben de cumplir para cada caso de uso.

A partir de los requerimientos funcionales y no funcionales se crean paquetes encapsulando los casos de uso en módulos que serán integrados en el sistema.

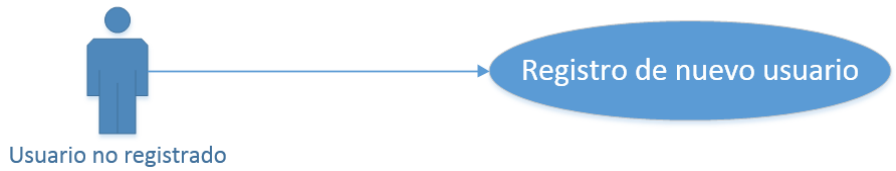


Ilustración 3 Caso de Uso usuario no registrado

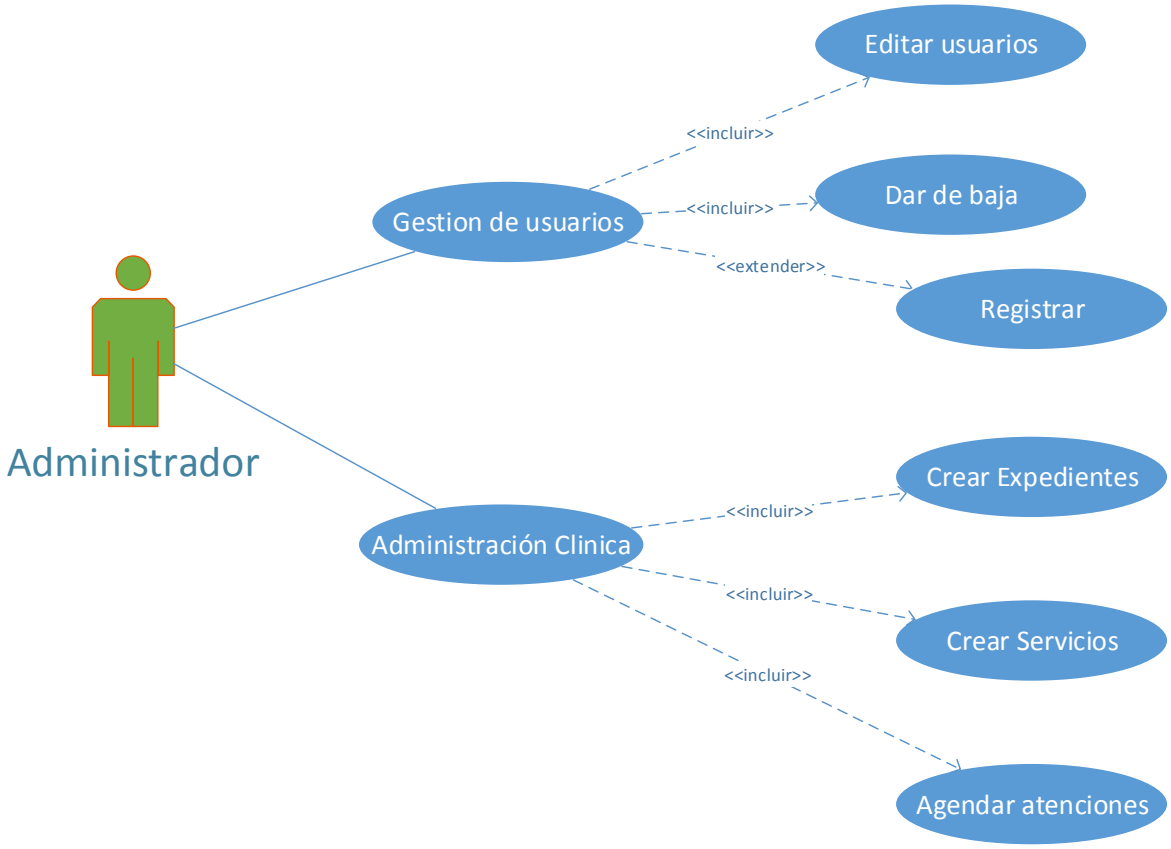


Ilustración 4 Casos de Uso administrador

Los usuarios no registrados tendrán la opción de acceder a la página principal del sitio y generar una solicitud de registro en el sistema, por otro lado el actor administrador está encargado de la gestión de los usuarios, además de tener la posibilidad de interactuar con la gestión odontológica de un usuario en específico, esto es mostrado en la relación “administrador – gestión odontológica” el cual resume todos los casos de uso del odontólogo y por ende de la misma manera podrían formar parte de la responsabilidad del administrador.

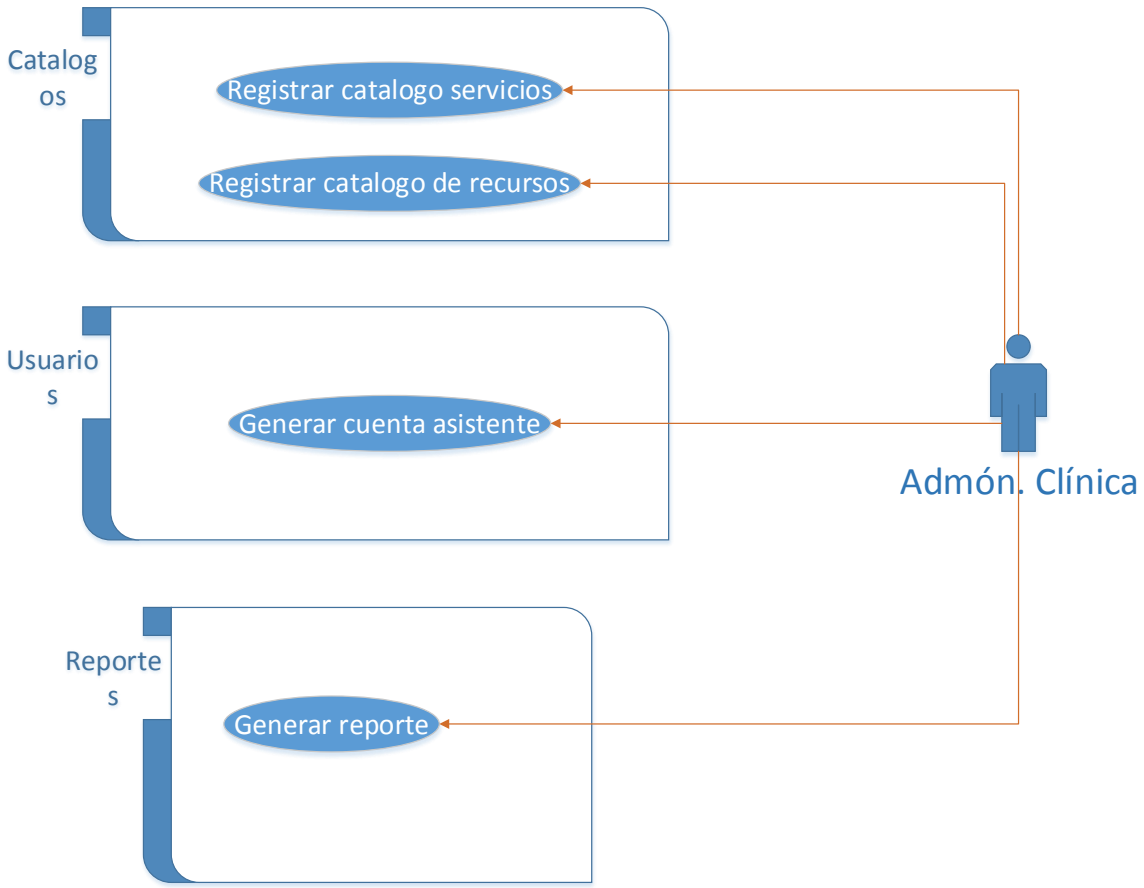


Ilustración 5 Casos de uso Admón. Clínica

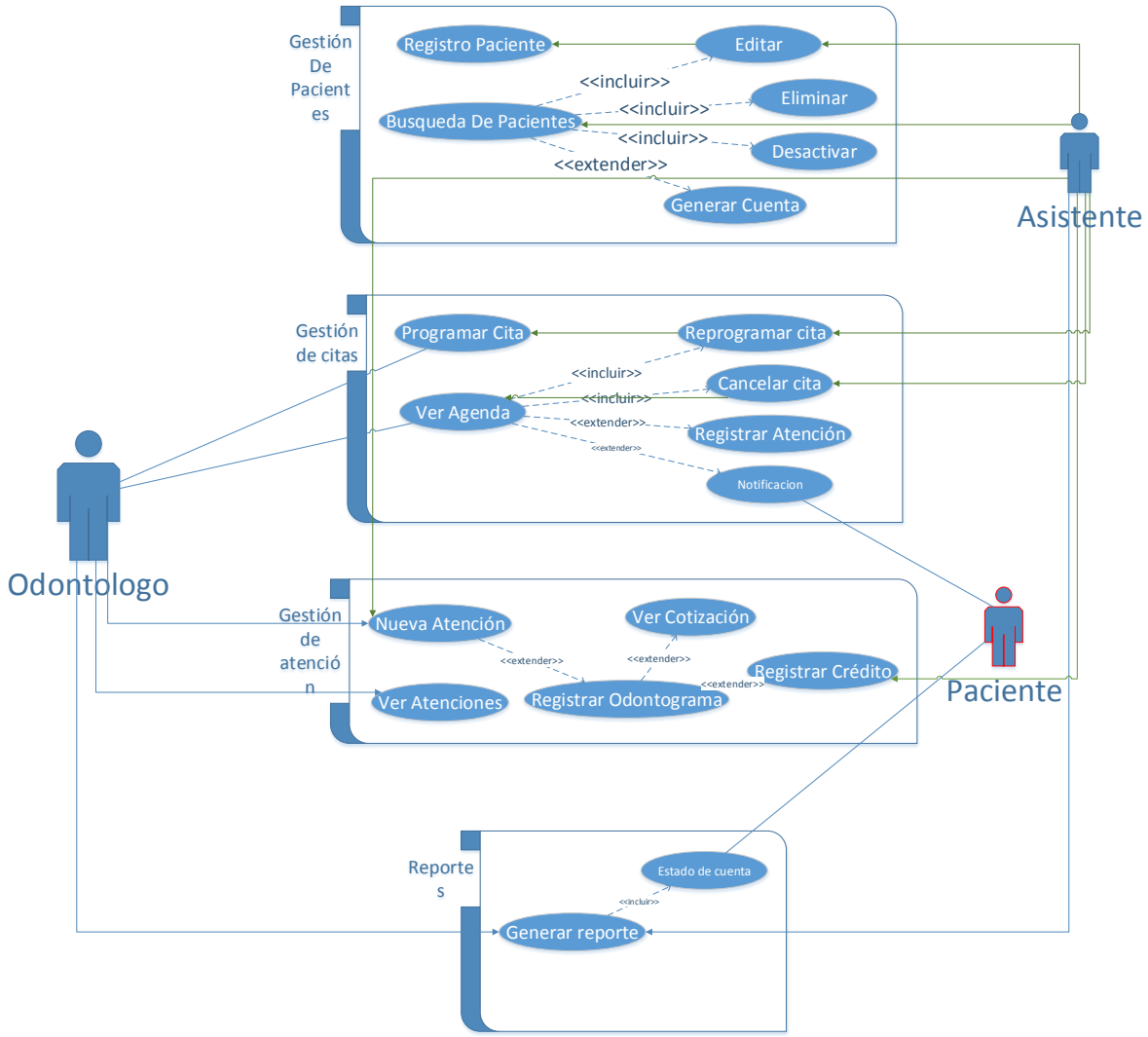


Ilustración 6 Casos de Uso personal clínica

Diagramas de actividades

El flujo normal de los casos de uso para los procesos principales del sistema será representado en diagramas de actividades, con el fin de aplicar buenas prácticas de UML para una obtener una mejor comprensión de la lógica con la cual ha sido diseñado y desarrollado el sistema. Ya habiéndose definido los casos de uso, se puede lograr un avance en la implementación de requerimientos, de manera que puedan ser trazados el flujo normal requerido en los casos de uso con los procesos de negocio desarrollados en el software.

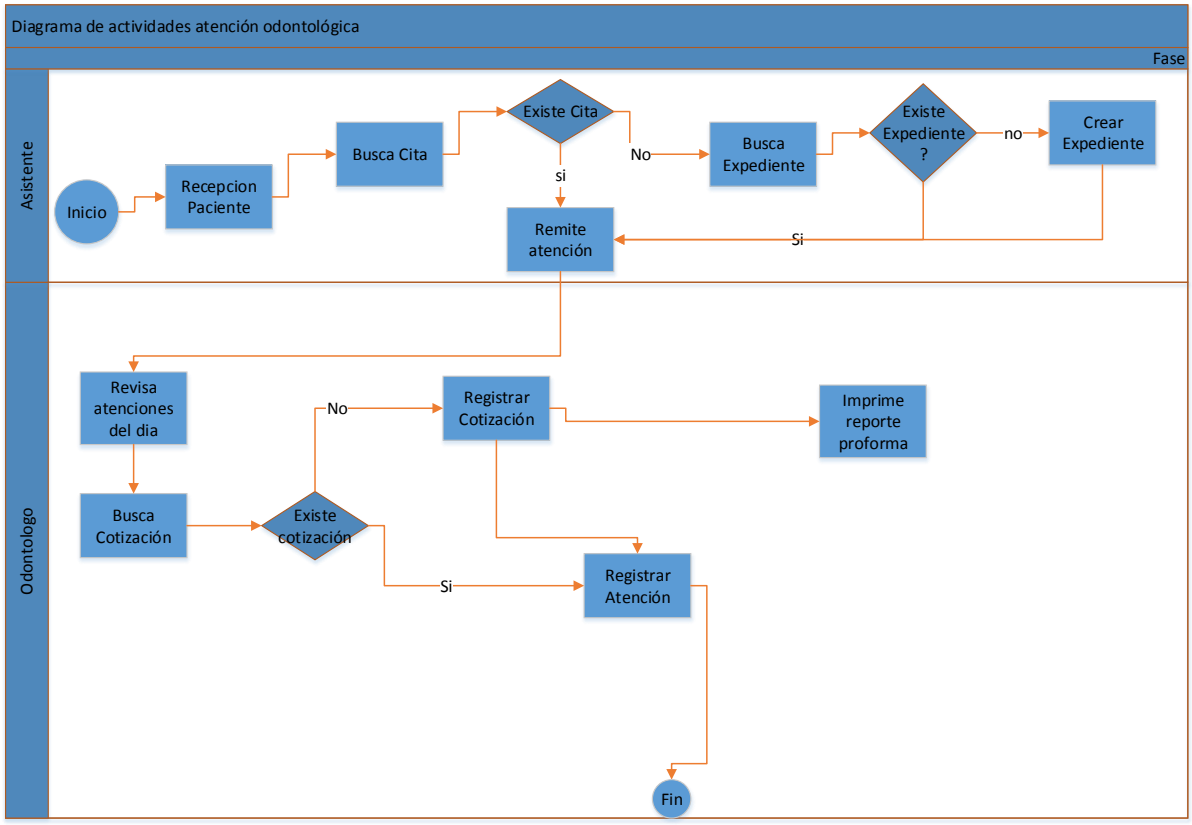


Ilustración 7 Diagrama de actividades (con marco de responsabilidad)

Modelo de dominio

Se usará el modelo de dominio como mecanismo de línea base, para la creación del diseño de los objetos de software que concebirá el modelo de base de datos y de clases. Se mostrará las clases conceptuales significativas para el dominio del problema, de la forma en que se muestran en la vida real (en contrastes con las clases en software).

El diagrama de dominio ayudara a comprender mejor los términos asociados al giro de negocios, además de relacionar estos términos de manera en que deberían representarse tal y como seria en un diagrama de clases, con la excepción de que no es necesario incluirle las operaciones destinadas a modificar los atributos de las entidades, ya que por ende el diagrama de dominio se aleja completamente del desarrollo de la aplicación en sí, es decir que la comprensión es basada simplemente

en como las palabras propias del ambiente de negocio serán tomadas en futuros análisis de implementación de clases y objetos de datos.

Por consiguiente, para la implementación del sistema tomando como referencia el diagrama de dominio, promete una buena comprensión del modelo de la aplicación.

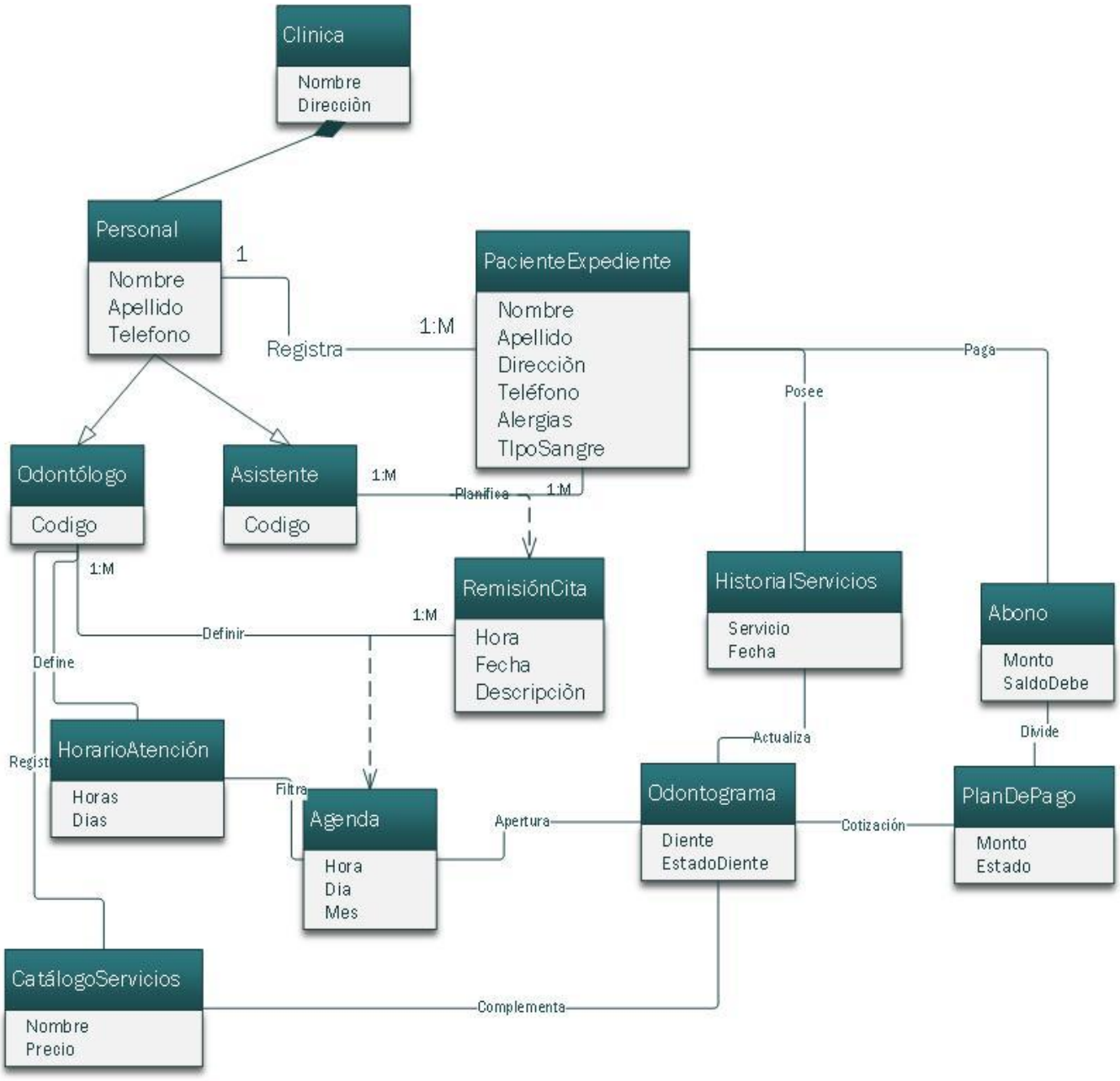


Ilustración 8 Modelo de dominio

Diseño del sistema

Capa conceptual

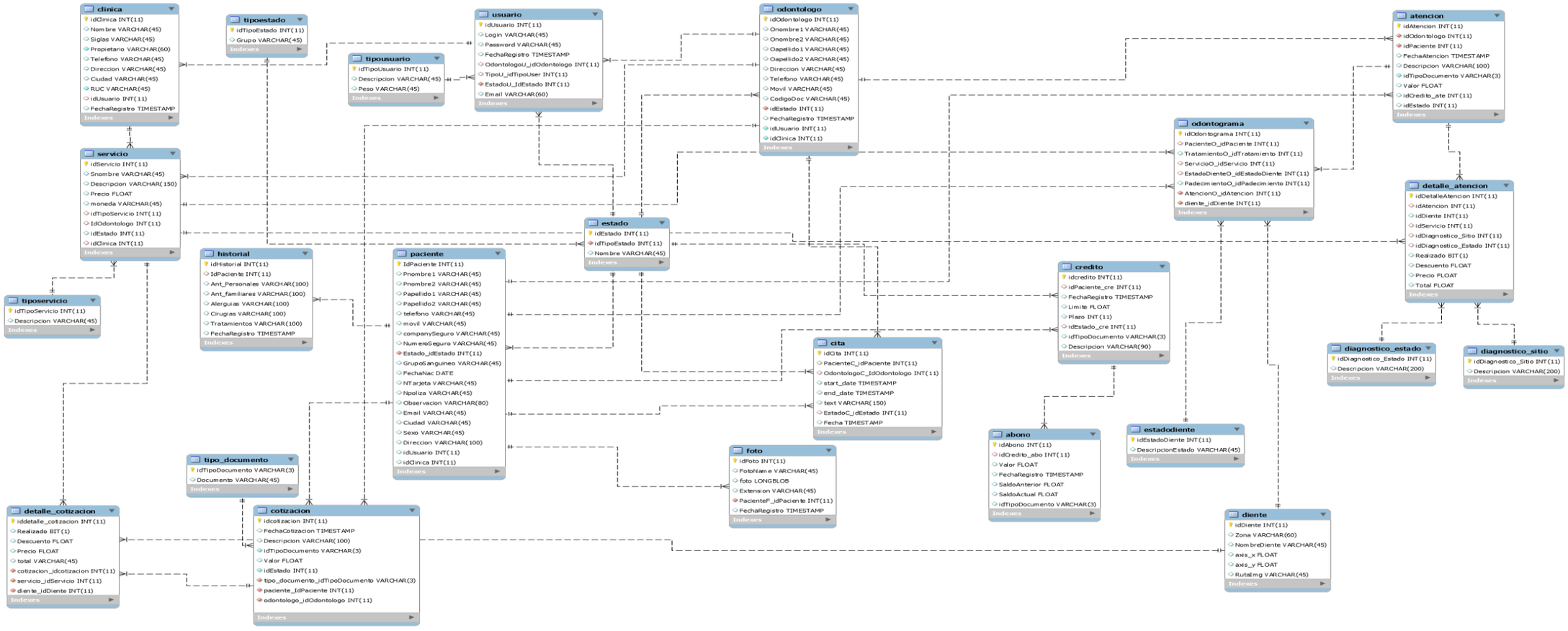


Ilustración 9: Modelo de datos

Como primer acápite de la etapa de diseño se muestra el modelo de datos relacional. La herramienta de diseño utilizada es el gestor de esquemas propio de MySQL Workbench, en el cual se puede generar los objetos de la base de datos, con la exportación de un script .SQL que contiene la estructura de la base de datos diseñada.

El modelo de datos está basado en el modelo de dominio el cual facilitó una mejor comprensión de los objetos afines al giro de negocio.

Diseño Arquitectónico

En el diseño arquitectónico se buscó representar las relaciones de módulos y componentes del sistema. Para llegar a este modelo se debió diseñar el modelo de datos como se vio anteriormente, se analizaron alternativas de estilos y de esta forma asegurar una estructura lo más adecuada posible para los requerimientos de los usuarios. Una vez seleccionada la alternativa, se procedió a elaborar la arquitectura utilizando un método de diseño basado en paquetes UML.

El modelo describe un primer contexto del sistema dentro del género arquitectónico orientado a servicios odontólogos, se diseñó con una estructura de implementación, de esta forma empacar las funciones de manera abstracta.

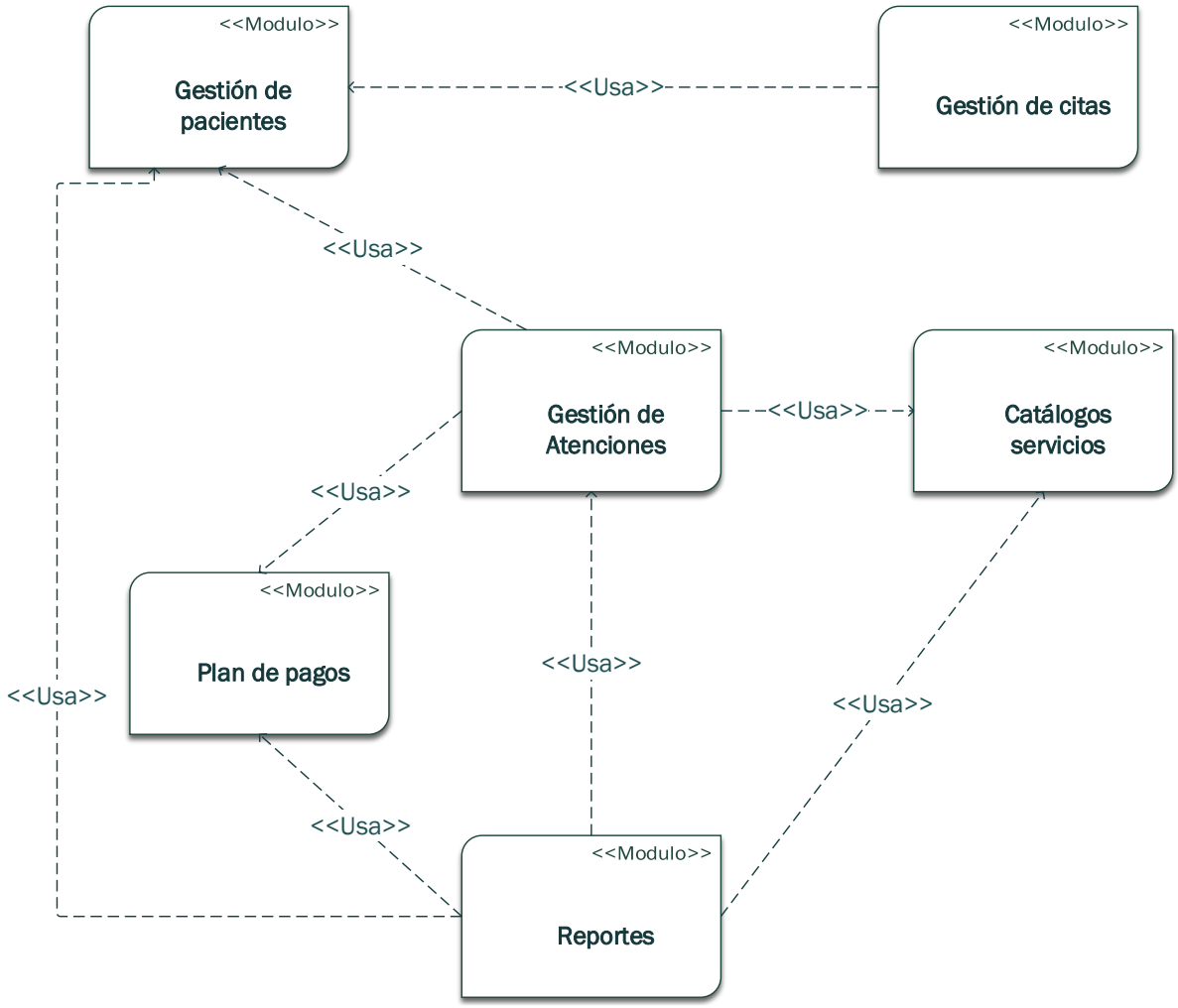


Ilustración 10 Diagrama Paquetes

Diseño Web

Modelo de navegación

Es un modelo jerárquico donde se esquematizan las actividades por niveles de usuarios que presenta una aplicación, es decir, es el orden de interacción de cada usuario. En este caso se mostrará el modelo de navegación para los niveles de usuario Administrador y Odontólogo.

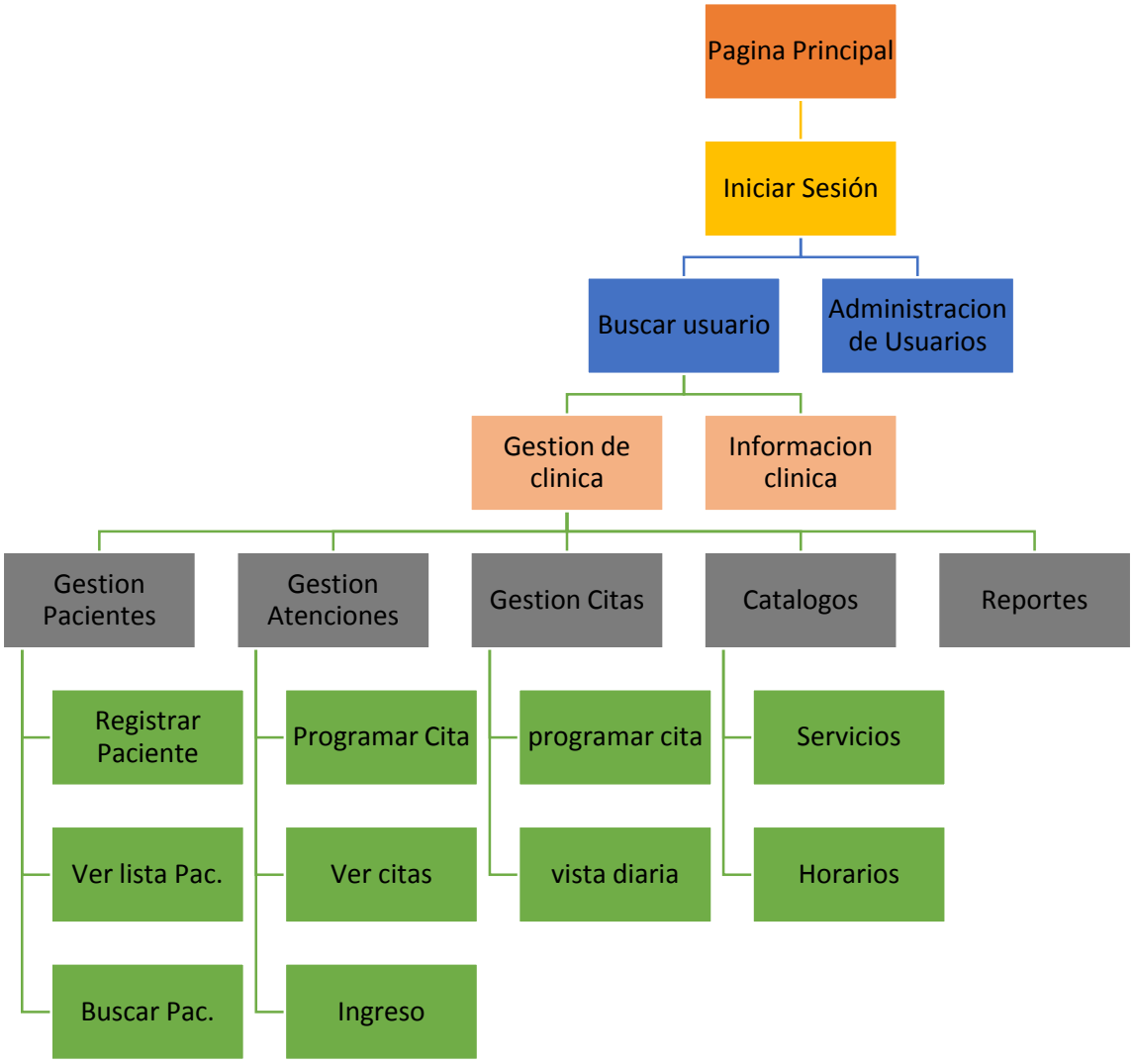


Ilustración 11 Modelo Navegación Admin

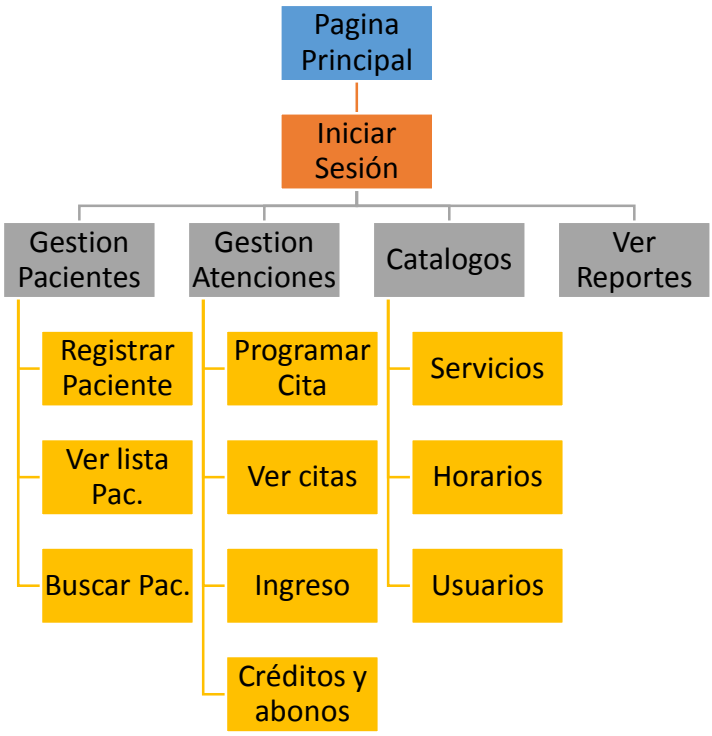


Ilustración 12 Modelo Navegación Admón. Clínica

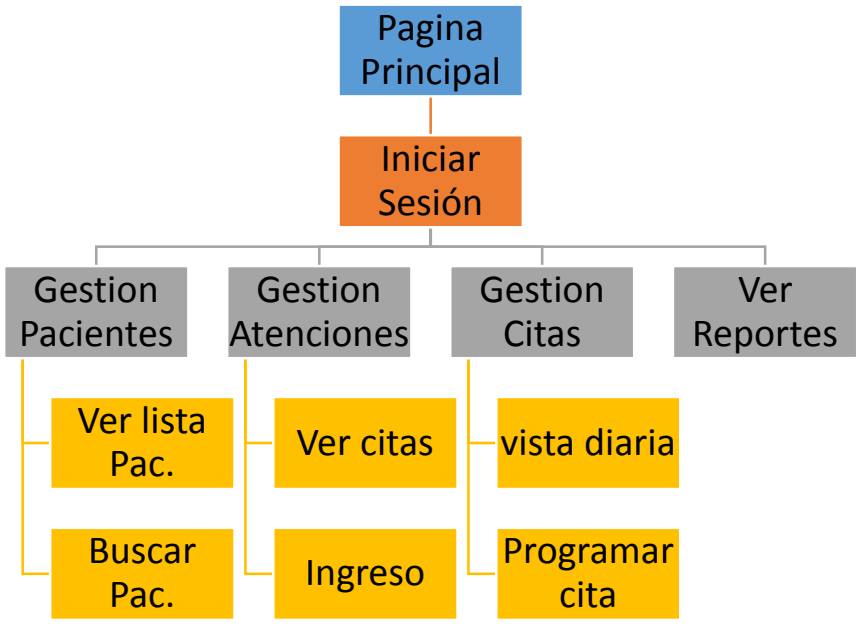


Ilustración 13: Modelo de navegación de Odontólogo

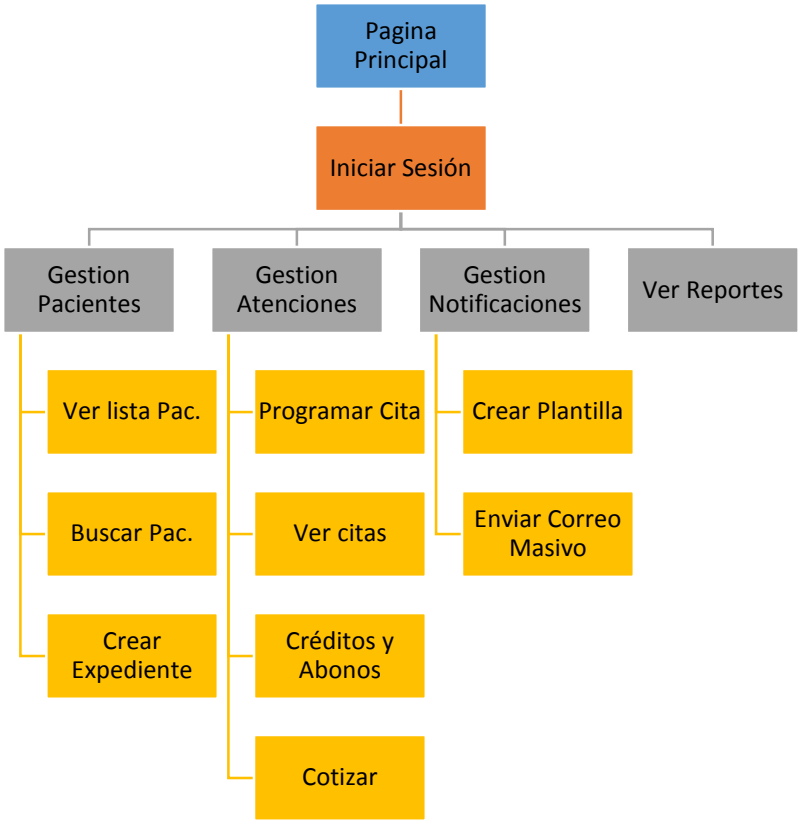


Ilustración 14 Modelo Navegación Asistente

Modelo de estructura de presentación

Se utilizará un diagrama de presentación para mostrar las clases de navegación y de proceso que pertenecen a una página web ya que el diagrama de navegación no muestra los objetos que estarán presentes en la interface. Se describirá dónde y cómo los objetos de navegación y accesos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario.

Se utilizará la siguiente simbología para definir los objetos a representar en los diagramas:

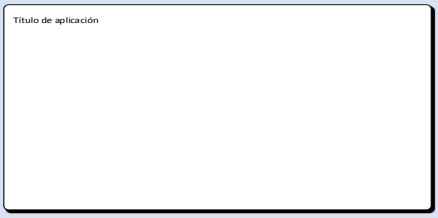
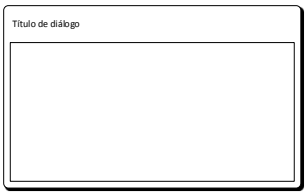
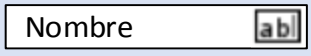
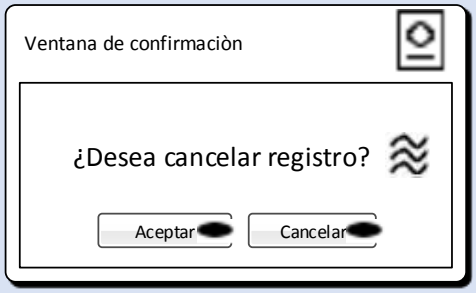
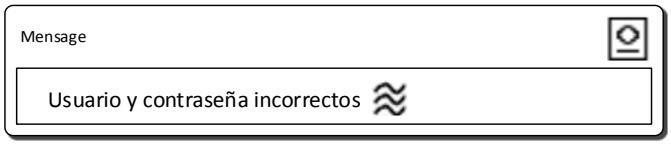
Nombre objeto	Gráfico	Descripción objeto
Ventana		Contendor de aplicación, puede representar una o un conjunto de clases
Formulario de aplicación		Ejecutor de las acciones del programa, también contiene los parámetros de entrada o salida
Caja de texto editable		Objeto que sirve como entrada de texto en un formulario
Botón		Ejecutor de la clase de aplicación, envía parámetros entre los objetos de navegación
Ventana emergente		Dialogo que no forma parte de la ventana principal, puede ser accionada desde una excepción al flujo de navegación
Etiqueta emergente		Dialogo que se muestra oculto dentro de la ventana principal, esta es activada por errores en parámetros del formulario

Tabla 25: Componentes del modelo estructura de presentación

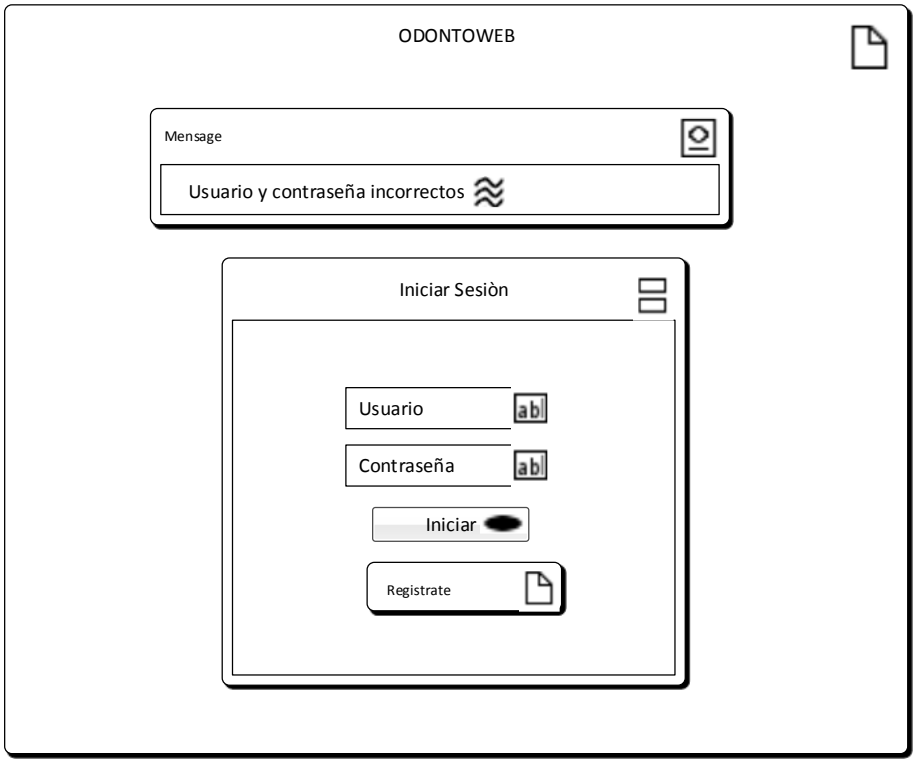


Ilustración 15 Diagrama de presentación inicio de sesión

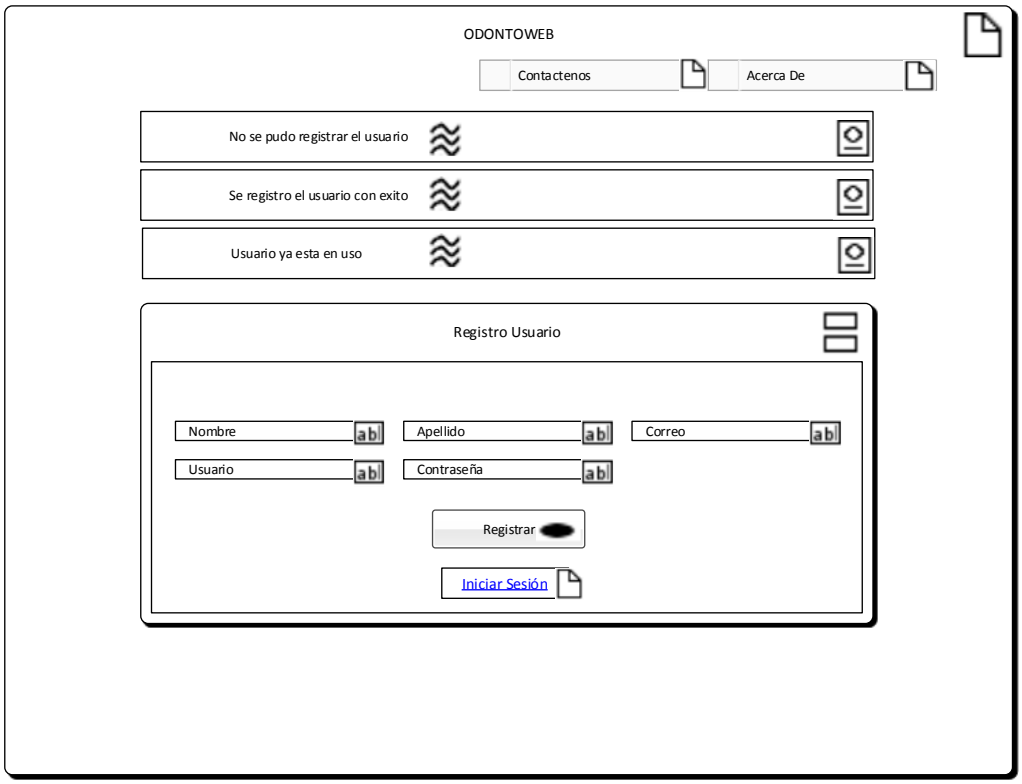


Ilustración 16 Registro de usuarios

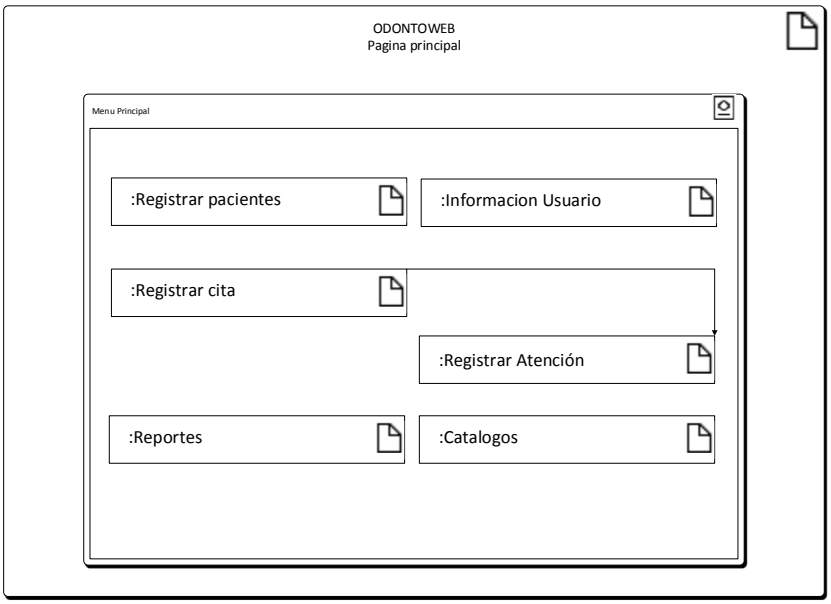


Ilustración 17 Menú principal

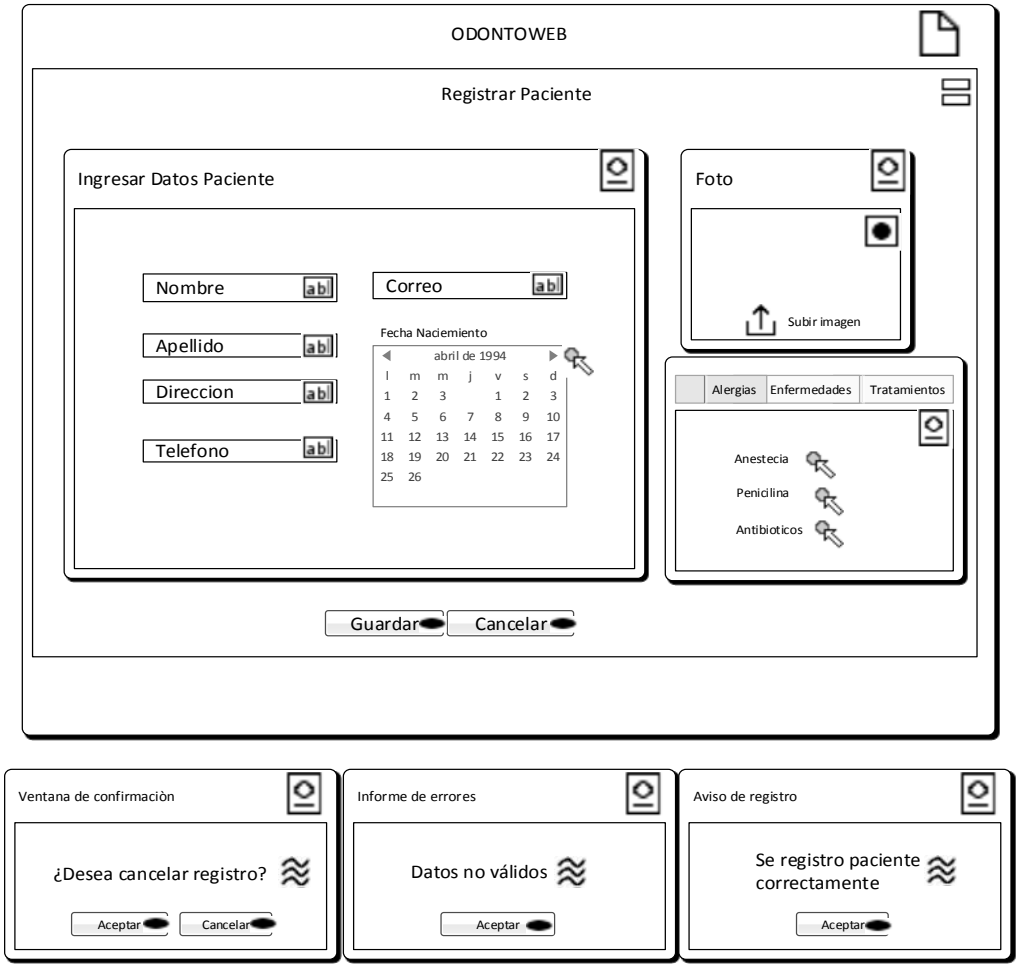


Ilustración 18 Registro de Paciente

Modelo abstracto de la interface

EL diseño de todas las interfaces deberá estar basado en un esquema de modelo abstracto, el cual servirá para regir la distribución del contenido de la aplicación como un plano para la edificación de una estructura, esto agiliza el diseño ya que estará predefinido el orden de los objetos de contenido. A continuación, se muestran los diseños bases de interfaces las cuales fueron creadas con la herramienta **Balsamiq Mockups**:

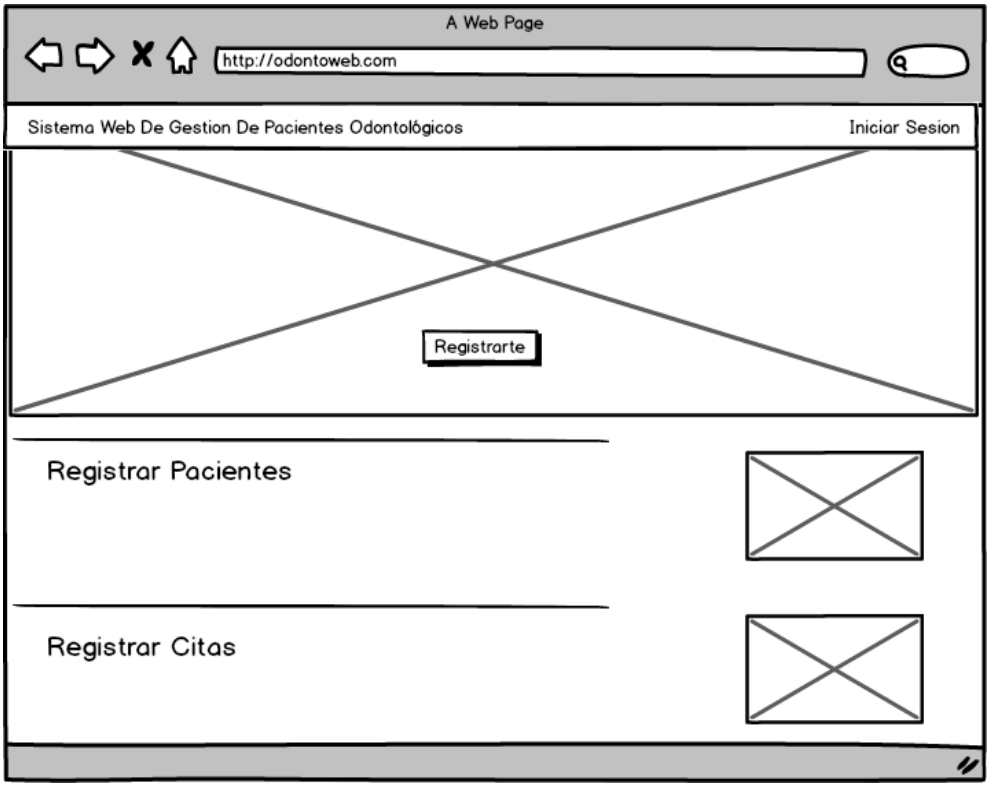


Ilustración 19: Modelo abstracto interface principal

Esta es la página principal del sistema (índex page), en ella podremos observar la presentación inicial, aquí los usuarios deciden iniciar sesión o registrarse en el sitio web.

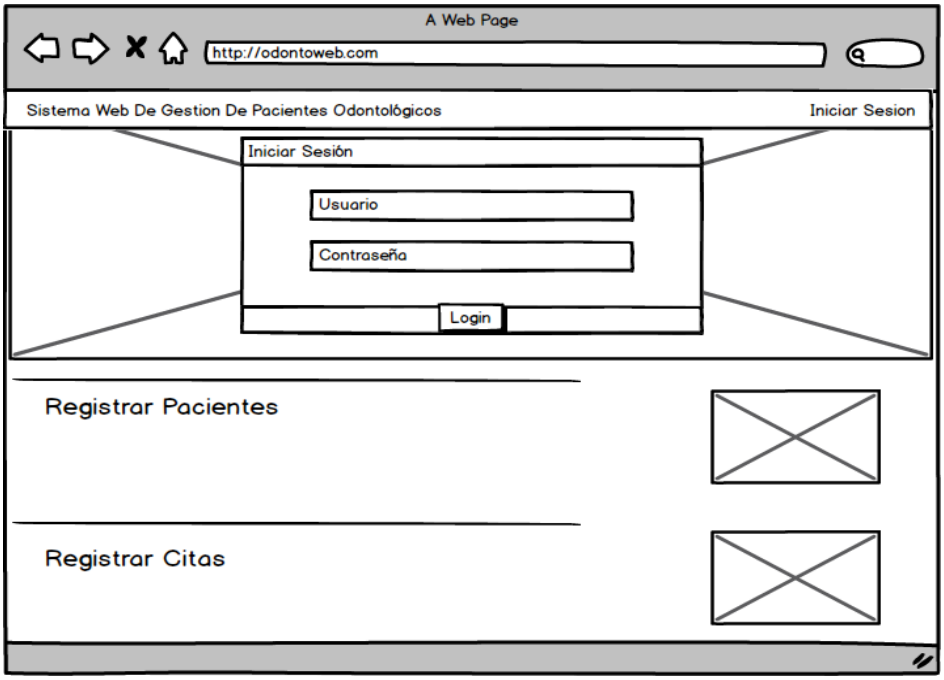


Ilustración 20: Modelo abstracto de Inicio de sesión

El inicio de sesión aparece después de haber presionado el botón de la barra de menú **inicio de sesión,** aparecerá una ventana de forma emergente.

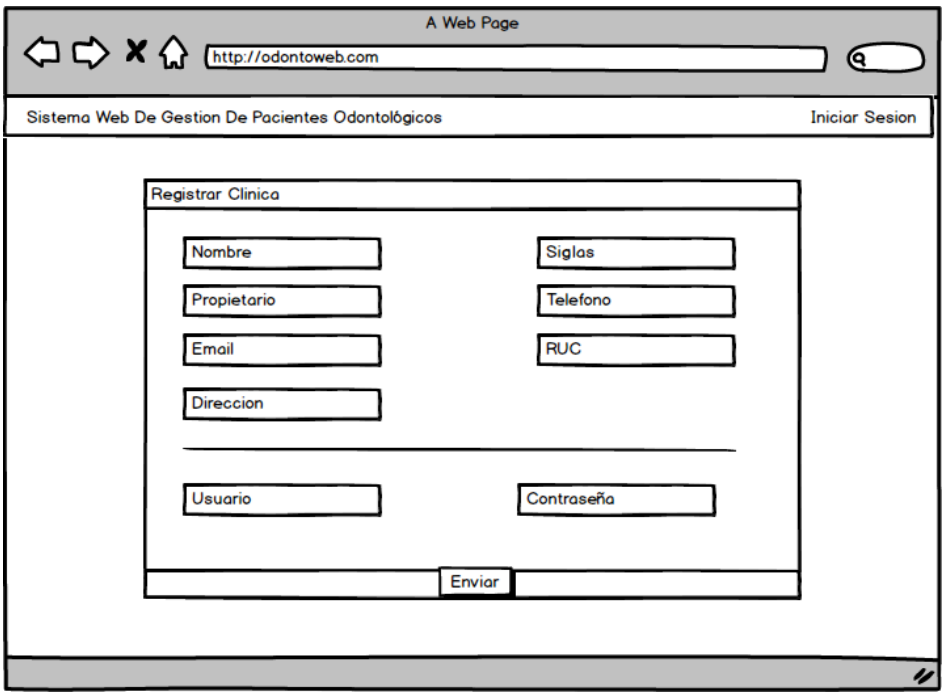


Ilustración 21: Modelo abstracto de registro de usuario

Si el usuario no tiene una cuenta, deberá registrarse para generar una solicitud de registro para nueva clínica, seguidamente el mismo usuario deberá confirmar la cuenta entrando a su correo afiliado en el sistema.

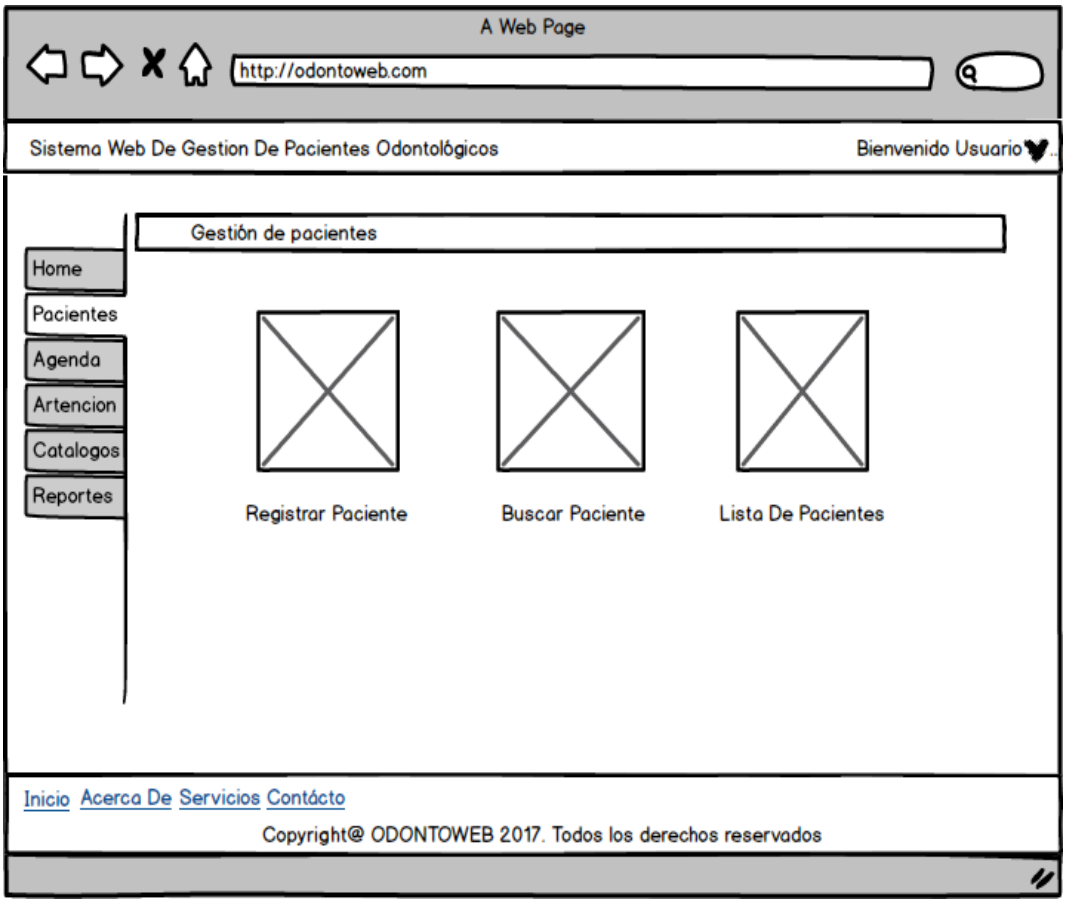


Ilustración 22: Modelo abstracto de opciones de menú

La pantalla de inicio aparece después de que un usuario inicio sesión, en ella se encuentran los accesos a módulos del sistema, como ejemplo tenemos registro de pacientes, atenciones, citas, etc.

.

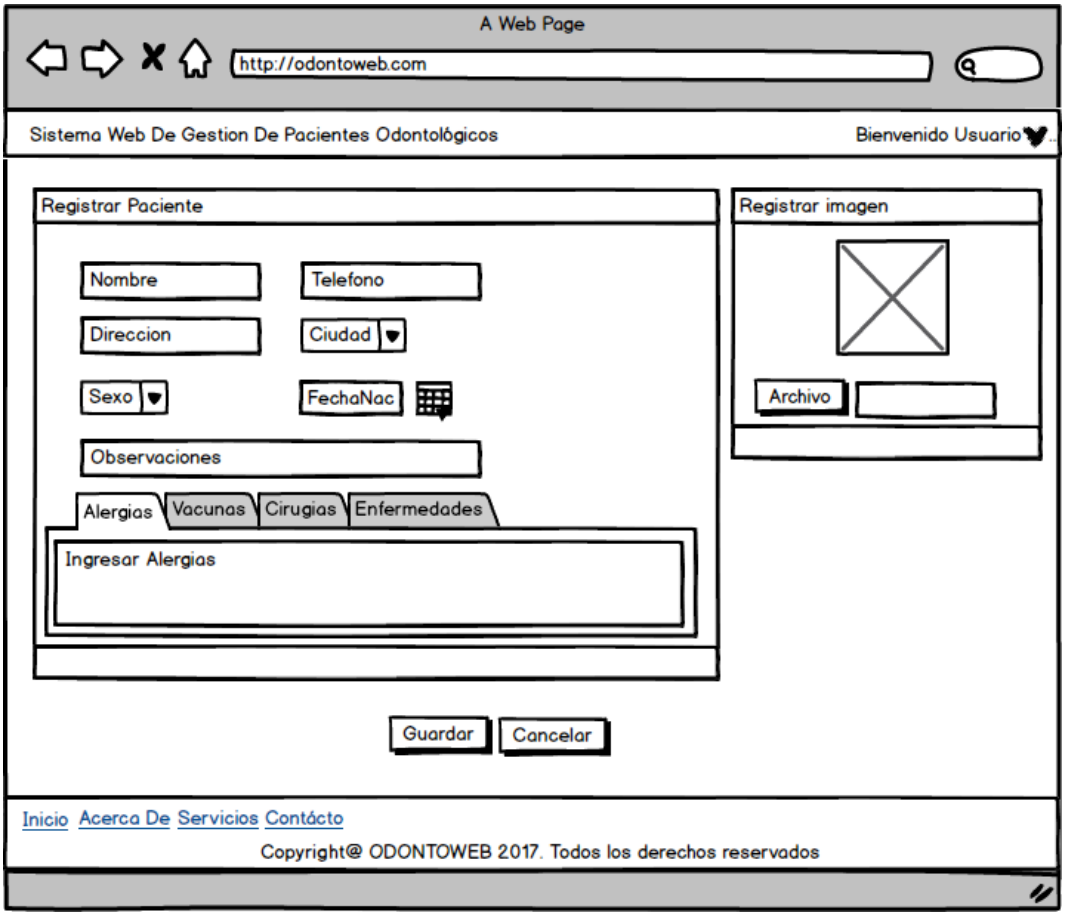


Ilustración 23: Modelo abstracto de registro de datos

Aplicación de los estándares para el desarrollo del proyecto web

Proceso de diseño y evaluación

Entendimiento y expectativas del usuario

El uso de paginación y navegación familiar en el diseño web, hace más fácil para los usuarios aprender y recordar el layout del sitio. Teniendo en consideración el argumento planteado anteriormente, se diseñó el ODONTOWEB, de tal manera que se asegure que aun para usuarios que no visiten el sitio frecuentemente, aprenderán a usar el sitio de manera fácil y rápida. Ver ilustración 8.

Ilustración 24 página de inicio ODONTOWEB

El sitio será encontrado fácilmente

Con el objetivo de tener una alta probabilidad de ser accedido, el sistema se diseñó, de tal forma que pueda asegurar que cualquier motor de búsqueda en la web encuentre el sitio entre los primeros en la lista.

Para esto se hizo uso de etiquetas <meta>, detallando explícitamente el nombre, contenido, tema, título y palabras claves para acceder a ellos

Optimizar la experiencia de usuario

Estandarizar secuencias de tareas

Se diseñaron componentes que estandariza la funcionalidad dentro de la aplicación, esto con el propósito de permitir a los usuarios realizar tareas que les permitan aprender ciertas secuencias y comportamientos al momento de interactuar con el sistema. Ver ilustración 9

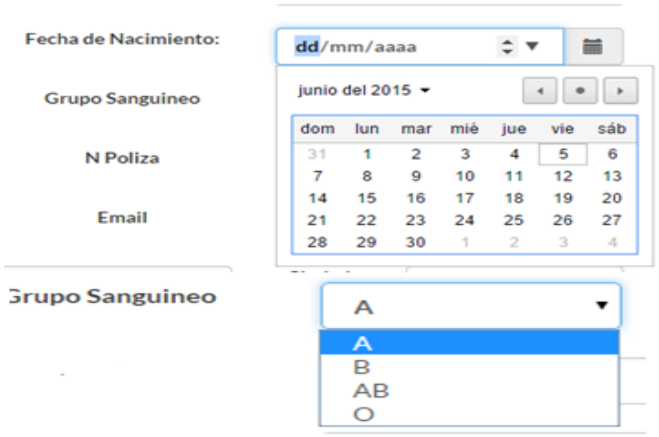


Ilustración 25 Estandarización de componentes del ODONTOWEB

Formatear información para leer e imprimir

En esta sección se preparó el sitio, de tal forma que la información quede estructurada de manera que los usuarios decidan leer el documento o imprimir la información si así lo desea. **Ver ilustración 24**

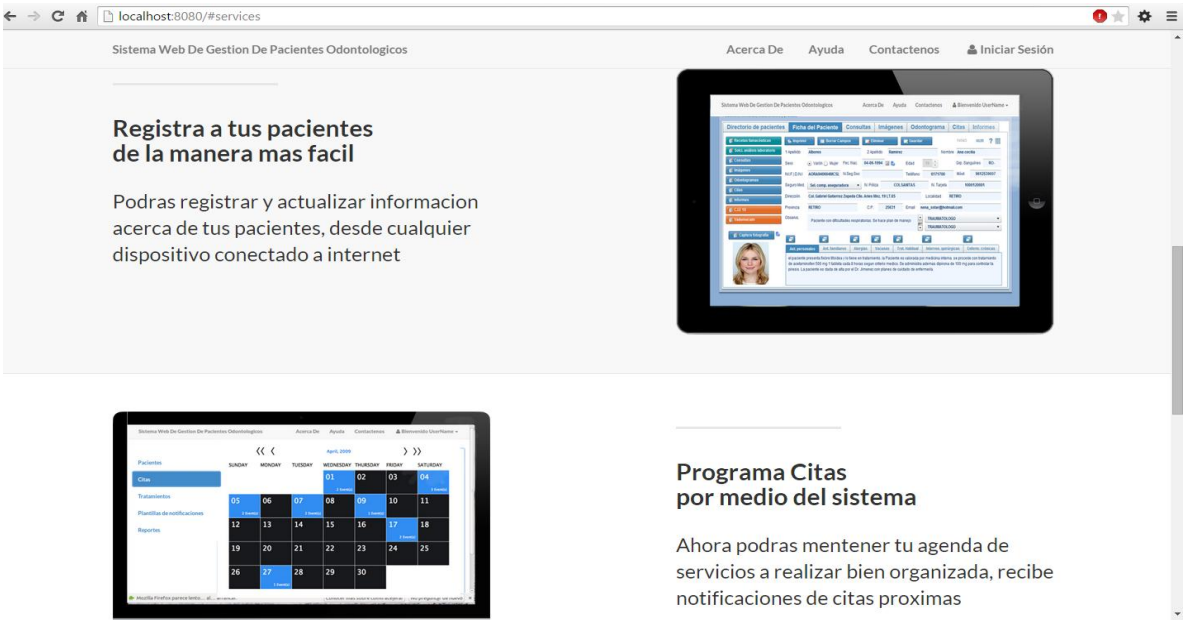


Ilustración 26 Diseño estructurado para comprensión del usuario

Accesibilidad

Diseño para exploradores comunes

Se diseñó, desarrolló y probó las funcionalidades, en diferentes exploradores web, a continuación la ilustración 11 muestra el sitio web ejecutado y probado en el explorador web Internet Explorer, así mismo la **Ver ilustración 25**, muestra el sitio ejecutado y probado en el explorador web Google Chrome.

Ilustración 27 ODONTOWEB en internet Explorer

Ilustración 28 ODONTOWEB en google chrome

Habilitar acceso a la página de inicio

El objetivo de esta sección de la guía de diseño, es permitir varios medios de acceso al usuario, por lo tanto, se diseñó el sitio, de tal modo que permitiese al usuario retornar a la página de inicio desde cualquier sitio dentro de la aplicación. **Ver ilustración 27**, muestra el acceso al home page, desde la página de registro de médico.

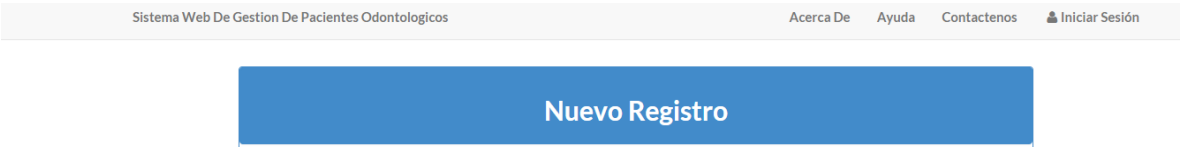


Ilustración 29 acceso al home page desde la página registro

Navegación

Agrupamiento de elementos de navegación

El ODONTOWEB, se diseñó usando un esquema de navegaciones localizadas en tabs, encabezados, listas y búsquedas, esto aplica para cada página dentro del sitio. La parte superior de la **Ver ilustración 28**, muestra los tags de navegación agrupados en lista que permitirá el acceso en todo el sitio.

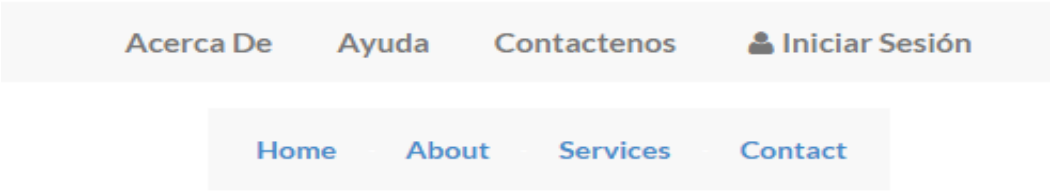


Ilustración 30 navegación en ODONTOWEB

Presentación de tags

Los tags de navegación, se diseñaron para evitar confusión del uso al usuario; estos están agrupados de tal forma que puedan diferenciarse, y ser remarcables en el sitio. **Ver ilustración 29**

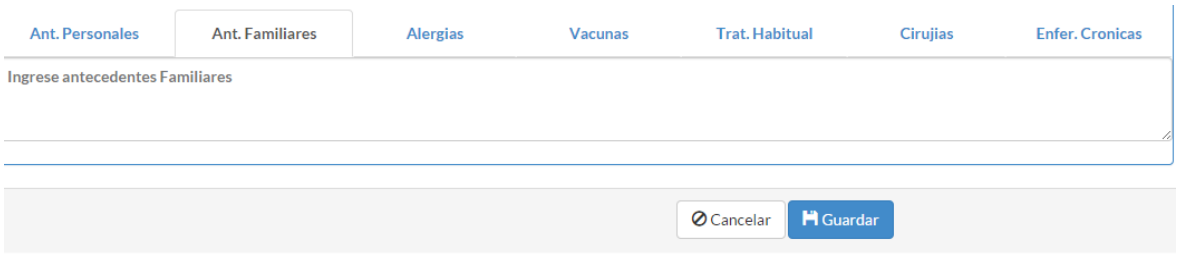


Ilustración 31 presentación de tags

Encabezados, títulos y etiquetas

Uso de un mismo formato de encabezado en cada página

El diseño se estableció con el fin de asegurar encabezados descriptivos y relacionados al contenido que se está mostrando. **Ver ilustración 30 y 31**

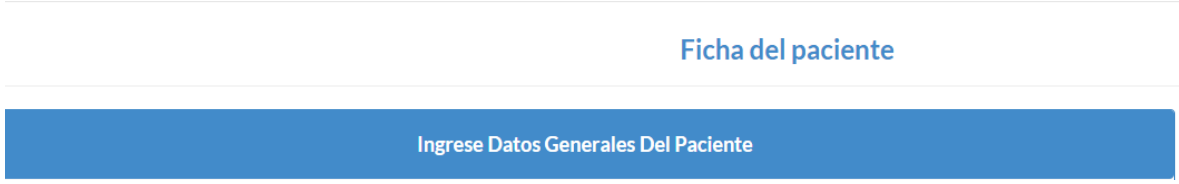


Ilustración 32 encabezado ficha del paciente

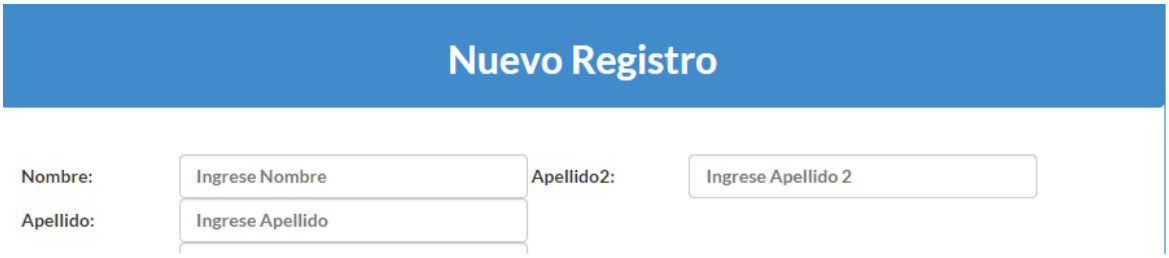


Ilustración 33 encabezado nuevo registro

Uso de encabezados en orden apropiado

Usando apropiadamente las etiquetas HTML de encabezado, ayuda a los usuarios a entender la jerarquía de información en el sitio, La siguiente imagen, muestra un segmento del código en el que se implementó uso el correcto de encabezados HTML.

```
<div class="page-header" align="center">
        <h3 class="text text-primary">Ficha del paciente</h3>
</div>

                <form id="formRP" class="form">
                <div class="row">

                        <div class="col-sm-9">
                        <div class="panel panel-primary" align="Center">
                                <div class="panel panel-heading">
                                        <h4>Ingrese Datos Generales Del Paciente</h4>
                                </div>
```

Ilustración 34: Uso apropiado de encabezados HTML

Gráficos, imágenes y multimedia

Uso de imágenes de fondo simples

Las imágenes de fondo pueden dificultar a los usuarios leer e incluso entender el contenido mostrado en el sitio, por ende, se estipuló una imagen de fondo simple, adaptable, flexible y agradable para la vista del usuario, considerando que el objetivo de esta aplicación web es no comercial.

Aspectos de funcionalidad

Tiempo de respuesta

En el desarrollo del presente sistema se consideró el tiempo de repuesta de las peticiones y consultas hechas por el usuario, y se tomaron en cuenta los siguientes aspectos:

- Tiempo de respuesta al realizar consultas de definición de datos
- Tiempo de respuesta al realizar consultas de Manipulación de datos

Para cumplir y reducir estos factores que minimizan la óptima interacción del usuario con el sistema, se proponen las siguientes técnicas de desarrollo de funcionalidad:

- Buenas prácticas de codificación: Esta técnica evita código redundante que retrasan el tiempo de respuesta, eliminando líneas de códigos que están de más en la clase objeto.
- Implementación de modelo MVC para optimizar la ejecución e integridad de los datos, esta metodología de desarrollo de codificación permitió que se optimice y actualice de manera más amigable. Ver ilustración #19

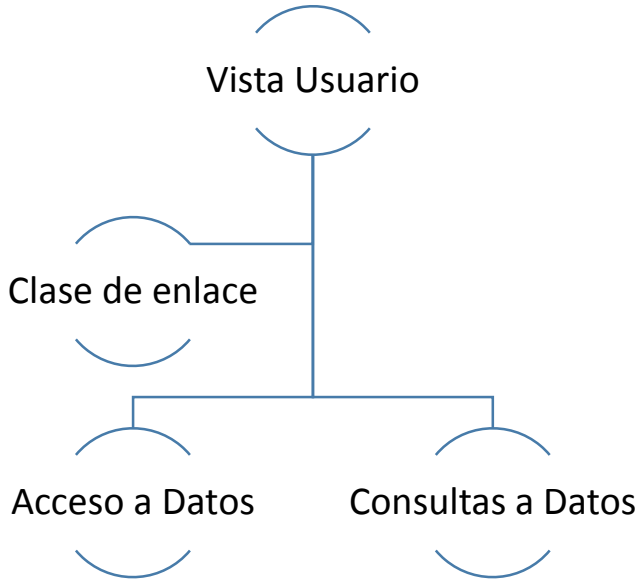


Ilustración 35 Modelo MVC

Notificación de error

Cuando el usuario haya cometido algún error de tipo lógico, el sistema notificará de forma automática el tipo de error y el motivo de error. La siguiente notificación muestra un ejemplo de notificación de error. Ver ilustración 14

Los posibles errores considerados se enumeran a continuación:

- Errores de acceso al sistema
- Errores de registro de datos
- Errores por pérdida de conexión.

Para cada uno de estos errores o excepciones se proveen de mecanismos intuitivos para la comprensión del error y lo que se debe de hacer para remediar por parte del usuario.

Ilustración 36 Notificación de error

Fase de desarrollo del sistema

En el siguiente acápite se detallará la información técnica del desarrollo del sistema, se documenta la descripción de las clases, procedimientos y métodos, que conforman el software, las herramientas necesarias para la implementación de ODONTOWEB y un plan de pruebas pensado para medir la funcionalidad de la aplicación conforme los resultados esperados.

Entorno de desarrollo

ODONTOWEB está codificado desde su base o capa de controlador con el lenguaje java versión 1.7 (implementación de servlets y JSP's), se desarrolló en la herramienta Java Enterprise edition de Eclipse la cual incluye extensiones para la creación de aplicaciones web de manera sencilla. También se utilizó HTML5 y CSS3

para la capa de vista del lado del cliente. Posee recargado asíncrono de objetos gracias a la implementación de AJAX utilizando javascript y jquery.

La aplicación utiliza el servidor web Apache tomcat versión 7 como contenedor de servlets especializado en esta tarea, y el servidor de bases de datos mysql versión 5.6 permite múltiples conexiones y peticiones con gran rendimiento. Las ventajas de estas herramientas además de su gran capacidad es que son completamente open source.

Para el control de versiones del código fuente, se utilizó un servidor subversión gratuito provisto por el sitio web Assembla.com, el cual permite la creación de múltiples repositorios. Se integró un plugin de subversión en Eclipse para un manejo más cómodo del repositorio.

Implementación de ODONTOWEB

En el siguiente acápite, se detallará y se explicara parte del código fuente que componen los módulos del sistema, los cuales son de mucha importancia, además de puntualizar como se implementó el modelo MVC en el código.

Estructura del código fuente

Los proyectos de tipo JAVA EE, tienen una estructura que separa de algún modo el source (código fuente) de la aplicación con todos los componentes web que puedan ser creados. Es decir que por defecto los proyectos web de java están segmentados por una capa encargada a las interfaces de cliente y otra de servidor dentro de la misma carpeta a como vemos en la siguiente imagen:

Ilustración 38 Estructura proyecto Java EE

En el proyecto se crean diferentes carpetas automáticamente definidas por la perspectiva Java EE. En la ***carpeta Java Resources*** se almacenan cada una de las clases, servlets y por ende los métodos que conforman la capa de aplicación del servidor, en la ***carpeta WebContent*** se crean todos los archivos web (JSP, HTML, xml, Javascript, CSS, Jason, etc...) que son desplegados por el usuario desde un navegador web, los cuales funcionan como la interface de comunicación entre el usuario y el listener de peticiones del servidor (Servlets).

Un proyecto web de java por si solo implementa un sistema de capas con una lógica de comunicación efectiva, para ODONTOWEB se aprovechó esta arquitectura de desarrollo provista por el lenguaje, pero también se implementaron modelos de programación MVC, para poder segmentar el código fuente y separarlo de manera más conveniente.

Para dar solución a la implementación de capas en el sistema ODONTOWEB se crearon paquetes para poder encapsular cada una de las clases según nivel en el modelo MVC. El paquete Dao es el encargado de contener las clases que realizan conexiones a la base de datos, por otra parte, el paquete Common contiene todas las clases que controlan las peticiones y hacen las llamadas a la capa de datos. Para las vistas la última capa del modelo, no había otra solución más que utilizar el WebContent.

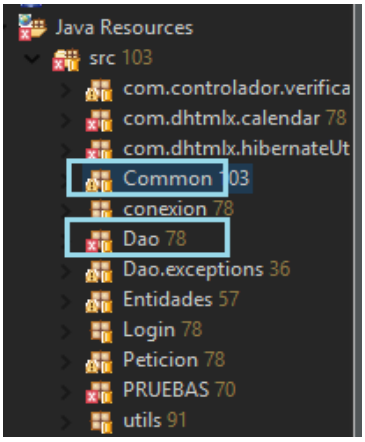


Ilustración 39 Paquetes ODONTOWEB

Redefiniendo el modelo MVC según la solución planteada de ODONTOWEB quedaría de esta manera:

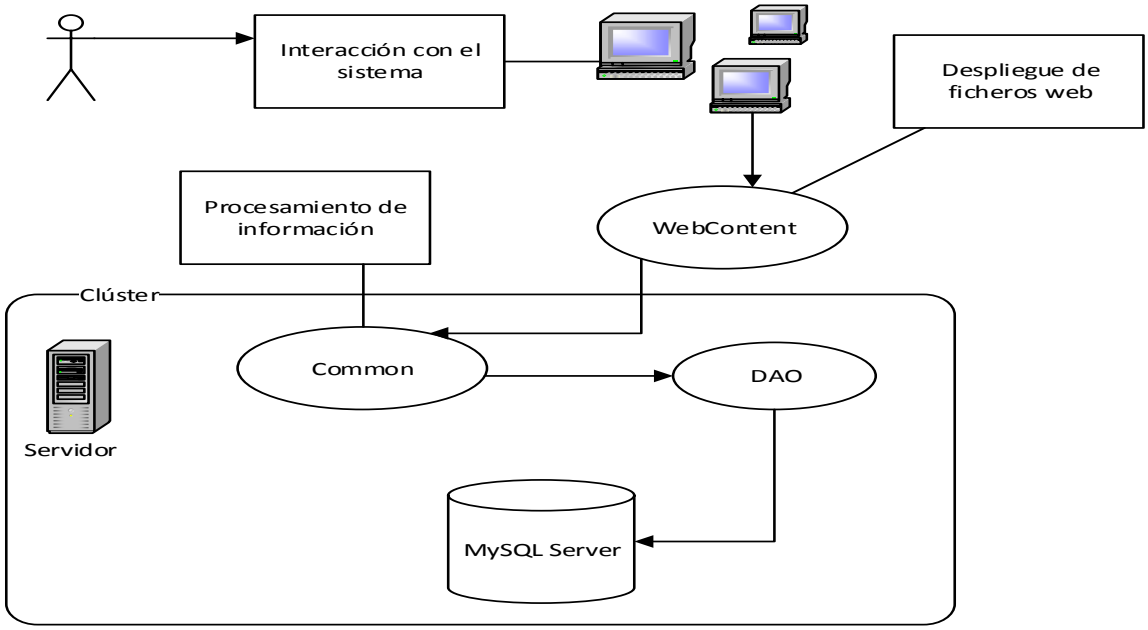


Ilustración 40 Modelo interconexión ODONTOWEB

En la carpeta principal del código (src) existen otros paquetes complementarios que funcionan para hacer algunas validaciones, las más importantes son los controladores de peticiones web las cuales están en el paquete Peticiones.

Peticiones Web

La clase padre de las peticiones se llama BasePeticiones.java, esta contiene métodos set y get para poder obtener el context, request y response de la aplicación. Las clases de Common deben heredar de BasePeticiones ya que necesitan obtener el contexto de la aplicación para recibir las peticiones y procesarlas o validarlas antes de enviar la petición hacia las clases Dao.

```java
package Peticion;

import java.io.IOException;

public class BasePeticiones {
  private HttpServletRequest request;
  private HttpServletResponse response;

  public HttpServletRequest getRequest() {
    return request;
  }

  public void setRequest(HttpServletRequest request) {
    this.request = request;
  }

  public HttpServletResponse getResponse() {
    return response;
  }

  public void setResponse(HttpServletResponse response) {
    this.response = response;
  }

  public String getContextPath(){
      return request.getServletContext().getRealPath("/");
  }

  public void enviarRespuestaHtml(String respuestaHtml) throws IOException{
    HttpServletResponse respuestaHttp = this.response;
    respuestaHttp.setContentType("text/html;charset=UTF-8");
    respuestaHttp.getWriter().write(respuestaHtml);
  }
}
```

Ilustración 41 BasePeticiones.java

Dentro de las peticiones existe un servlet muy importante llamada ManajarPeticiones.java, este se encarga de redireccionar las peticiones que lo han llamado desde un Action(propiedad de <form> HTML) hacia el controlador en Common especificado por los parámetros method y class.

```java
package Peticion;
import java.io.IOException;

@WebServlet("/ManejarPeticiones")
public class ManejarPeticiones extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        doPost(request, response);
    }
    protected void doPost(  HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        //Le mandamos un parametro que haga match con una clase especifica y un metodo especifico
        //Se utiliza instroneccion para crear una instancia de la clase que viene en el request como parametro.
        try{
            request.setCharacterEncoding("iso-8859-1");
            //Obtenes la clase que vamos a instanciar
            Class<?> action = Class.forName(request.getParameter("class").toString());
            Object actionObject = action.newInstance();

            //Setean parametros de request y response en el objeto instanciado esto es inyeccion de dependencias.
            Class<?> requestClass = HttpServletRequest.class;
            Object requestParam = request;
            Class<?> respondeClass = HttpServletResponse.class;
            Object responseParam = response;
            Method setRequest = action.getMethod("setRequest", requestClass);
            setRequest.invoke(actionObject, requestParam);
            Method setReponse = action.getMethod("setResponse", respondeClass);
            setReponse.invoke(actionObject, responseParam);

            //Metodo de inicio
            Method startPoint = action.getMethod(request.getParameter("method").toString());
            startPoint.invoke(actionObject);
        } catch(ClassNotFoundException notFoundClass){
            //Si no encontramos el metodo entonces la operacion es invalida.
            //deberiamos de poder encontrarlo porque sabemos lo que estamos buscando.
            response.sendRedirect("invalido.jsp");
            notFoundClass.printStackTrace();
        } catch (InstantiationException instantationException) {
            instantationException.printStackTrace();
        } catch (IllegalAccessException illegalAccessException) {
            illegalAccessException.printStackTrace();
        } catch (NoSuchMethodException noMethodException) {
            //Si no encontramos el metodo entonces la operacion es invalida.
            //deberiamos de poder encontrarlo porque sabemos lo que estamos buscando.
            response.sendRedirect("invalido.jsp");
            noMethodException.printStackTrace();
        } catch (SecurityException securityException) {
            securityException.printStackTrace();
        } catch (IllegalArgumentException argumentException) {
            argumentException.printStackTrace();
        } catch (InvocationTargetException invocationException) {
            invocationException.printStackTrace();
        }
    }
```

Ilustración 42 ManejarPeticiones.java

```html
<form  action="ManejarPeticiones" method="post" enctype="application/x-www-form-urlencoded">
      <input type="hidden" name="class" id="class" value="Common.ClinicaCOM">
      <input type="hidden" name="method" id="method" value="insertarClinica">
```

Ilustración 43 envió de los parámetros class, method para ManejarPeticiones

Controladores Common

Siguiendo el proceso lógico de la comunicación entre capas, el mensaje o petición proveniente desde la capa de vista debió pasar por una redirección hacia su destino, según como se ha especificado en los parámetros. Haciendo una analogía esto es

parecido al proceso de un sistema de entrega de cartas, La carta en este caso la petición debe pasar por una central de correos la cual identifica el destinatario impreso en su contenido, y así poder ser ubicado en el sitio correspondiente al que debe llegar el mensaje. De igual forma La petición paso por el manejador de peticiones y se ha ubicado en una clase y método encapsulados por Common los cuales procesaran la información requerida.

Partiendo del modelo de dominio en la fase de análisis se implementó la clase controladora ClinicaCOM.java la cual contiene todos los métodos que afectaran al objeto clínica. Dentro de ella se encuentra el método insertarClinica el cual obtiene los parámetros incluidos en el request activo de la aplicación, haciendo uso de la herencia de la súper clase BasePeticiones.java y las mandas a la capa de datos.

La relación entra la capa de control y la de datos estará definida mayormente por la instanciación de las mismas.

```java
package Common;

import java.io.IOException;

public class ClinicaCOM extends BasePeticiones{

    public void insertarClinica(){
        String nom= super.getRequest().getParameter("txtNom1");
        String siglas= super.getRequest().getParameter("txtSiglas");
        String dir= super.getRequest().getParameter("txtDir");
        String tel= super.getRequest().getParameter("txtTel");
        String ruc= super.getRequest().getParameter("txtRuc");
        String user= super.getRequest().getParameter("txtLogin");
        String pass= super.getRequest().getParameter("txtpass");
        String email= super.getRequest().getParameter("txtcorreo");
        ClinicaDAO clinicaDAO=new ClinicaDAO();

        if(clinicaDAO.RegistrarClinica(nom, siglas, dir, tel, ruc, pass, user,email)){
            try {
                super.getResponse().sendRedirect("RegistrarClinica.jsp?ok=1");
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        } else
            try {
                super.getResponse().sendRedirect("RegistrarClinica.jsp?ok=0");
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

    }

}
```

Ilustración 44 ClinicaCOM.java

En la capa de control deben estar las llamadas a los parámetros que provienen del request del manejador de peticiones (ManejarPeticiones.java), además se debe realizar el procesamiento de la información, como por ejemplo validaciones de tipo. En dicha capa se deben definir todos los métodos que generaran la respuesta a la vista, el request podría ser enviado ya sea por medio de Strings (buffers) de información, formatos de textos como JSON o XML, entro otros.

Capa DAO

La capa DAO se encarga de obtener la información (parámetros) ya procesados en el Common y gestionar la conexión con la base de datos para persistir sobre las tablas relacionales, es importante la implementación de mecanismo para cumplir con la integridad de información, en el caso de ODONTOWEB se emplearon transacciones para cumplir con inserción de datos de forma atómica.

ClinicaDAO.java

```
package Dao;

import java.sql.ResultSet;


public class ClinicaDAO {
    Conexion Conexion=new Conexion();
    private java.sql.Connection con=null;

    public boolean RegistrarClinica(String nom, String siglas, String dir, String tel, String ruc, String pass, String user,String prop,String email,String uri) {
        java.sql.PreparedStatement stmt1=null;
        java.sql.PreparedStatement stmt2=null;
        con=Conexion.getconexion();
        boolean retorno=false;

        try {
            con.setAutoCommit(false);
            Statement s=con.createStatement();

            String sql ="Insert into usuario (Login,Password,TipoU_idTipoUser,EstadoU_IdEstado,Email)"
                    + " values(?,?,?,?,?)";
            stmt1 =con.prepareStatement(sql);
            stmt1.setString(1, user);
            stmt1.setString(2, pass);
            stmt1.setInt(3, 4);
            stmt1.setInt(4, 2);
            stmt1.setString(5, email);
            stmt1.executeUpdate();
            //stmt1.close();
            int idLastUser=-1;
            ResultSet re=s.executeQuery("select auto_increment from `information_schema`.tables where TABLE_SCHEMA = 'swgpo' and TABLE_NAME = 'usuario'");
             if(re.next())
                 idLastUser=re.getInt("auto_increment")-1;

            sql="Insert into clinica (Nombre,Siglas,Telefono,Direccion,RUC,idUsuario,Propietario)"
                    + " values(?,?,?,?,?,?,?)";
            stmt2 =con.prepareStatement(sql);
            stmt2.setString(1, nom);
            stmt2.setString(2, siglas);
            stmt2.setString(3, dir);
            stmt2.setString(4, tel);
            stmt2.setString(5,ruc);
            stmt2.setInt(6, idLastUser);
            stmt2.setString(7, prop);
            stmt2.executeUpdate();

            //re=s.executeQuery("select auto_increment from `information_schema`.tables where TABLE_SCHEMA = 'swgpo' and TABLE_NAME = 'clinica'");
             //if(re.next())
                // idLastUser=re.getInt("auto_increment")-1;

             Email e = new Email();
             boolean resultado =e.enviarCorreoRegistroOdontologo(email,user,idLastUser,uri);

```

Ilustración 45 ClinicaDAO.java

En el fragmento de código pasado, pudimos observar la clase ClinicaDAO.java la cual implementa la inserción de una nueva clínica en la base de datos, el método **RegistrarClinica** obtiene los parametros del controlador luego realiza la conexión con la base de datos mientras activa la transacción haciendo uso de los métodos propios de la librería **Java.Sql**, las consultas están definidas en el código y en caso de un error en la inserción, la transacción genera un **rollback** para devolver la integridad de datos a las tablas afectadas, en caso contrario se genera un **commit** para actualizar la información de manera correcta.

```
    if(resultado){
        con.commit(); //si se envia el correo se registrara la transaccion con exito
        retorno = true;
    }else{
        if (con != null) {
                System.out.println("Rollback");
                try {
                    // deshace todos los cambios realizados en los datos
                    retorno = false;
                    con.rollback(); //si sucede una exencion se genera un rollback

                } catch (SQLException ex1) {
                    System.err.println("No se pudo deshacer" + ex1.getMessage());
                }
            }
    }

} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    if (con != null) {
        System.out.println("Rollback");
        try {
            // deshace todos los cambios realizados en los datos
            retorno = false;
            con.rollback();

        } catch (SQLException ex1) {
            System.err
                    .println("No se pudo deshacer" + ex1.getMessage());
```

Ilustración 46 implementación de transacciones

Capa de vista

Para el desarrollo de la capa de vistas, Java EE proporciona herramientas que facilitan el trabajo para la codificación de tags HTML con la cual se crean las

interfaces WEB. En una aplicación Web generalmente se combinan el lenguaje de etiquetas HTML con JavaScript ya que este ayuda a manejar de una mejor forma las validaciones y eventos del lado del cliente, es la forma más versátil para desarrollar sitios web dinámicos.

Con HTML se crea la estructura de la vista utilizando etiquetas adecuadas para el despliegue de formularios, y controles que sirven como entrada de información para los usuarios, mientras que en JavaScript es utilizado para dinamizar la interacción del usuario al lado del cliente, con eventos precargados desde la descarga de los archivos del sitio.

A continuación, se ve un ejemplo de vista del lado del cliente implementando el registro de un usuario administrador de clínica.

```
<script type="text/javascript">
        function validarDatosPost(){
            var verificar = true;

            if (!document.frmRegistroOdontologo.txtNom1.value){
                alert("Este campo es requerido");
                document.frmRegistroOdontologo.txtNom1.focus();
                verificar=false;

            }
```

Ilustración 47 Validación de campo vacío JavaScript

```
<div class="row" align="center">
    <form id="frmRegistroOdontologo" name="frmRegistroOdontologo" class="form" action="ManejarPeticiones"
                  method="post" enctype="application/x-www-form-urlencoded">
              <input type="hidden" name="class" id="class" value="Common.ClinicaCOM">
              <input type="hidden" name="method" id="method" value="insertarClinica">
              <div class="span6">
                  <div class="panel panel-primary" >
                      <div class="panel panel-heading">
                          <h3>Ingrese datos de la clínica odontológica </h3>
                      </div>

                      <div class="panel panel-body" align="left">
                              <div class="row">
                                  <div class="col-sm-1">
                                  <label>Nombre:</label>
                                  </div>
                                  <div class="col-sm-6">
                                      <input type="text"class="form-control" name="txtNom1" id="txtNom1" placeholder="Ingrese nombre de la clínica" value="">
                                  </div>
                                  <div class="col-sm-1">
                                  <label>Siglas:</label>
                                  </div>
                                  <div class="col-sm-3">
                                  <input type="text"class="form-control" name="txtSiglas" id="txtSiglas"placeholder="Ingrese siglas">
                                  </div>
                          </div><br>
                          <div class="row">
                                  <div class="col-sm-1">
                                  <label>Propietario:</label>
                                  </div>
                                  <div class="col-sm-6">
                                  <input type="text" class="form-control" name="txtPropietario" id="txtPropietario" placeholder="Ingrese nombre del propietar
                                  </div>
                                  <div class="col-sm-1">
                                      <label>Telefono:</label>
                                  </div>
                                  <div class="col-sm-3">
                                      <input type="text" class="form-control" name="txtTel" id="txtTel"placeholder="Ingrese teléfono">
                                  </div>
                          </div><br>
                                  <div class="row">
                                      <div class="col-sm-1">
                                          <label>Email:</label>
                                      </div>
                                      <div class="col-sm-6">
                                          <input type="email" class="form-control" name="txtcorreo"
                                              id="txtcorreo"
                                              placeholder="Ingrese la dirección de correo electrónico">
                                      </div>
                                  </div>
                                  <br>
                                  <div class="row">
```

Ilustración 48 Estructura HTML formulario ODONTOWEB

Pruebas del software

Para las pruebas de ODONTOWEB se siguió la propuesta de estrategias de software para webApp de la documentación Pressman, el cual sugiere validar el programa a partir de los requerimientos de usuario, además de probar entradas de valores y resultados generados por estos de forma controlada. El objetivo es llegar a una correcta trazabilidad de los requerimientos con el producto final utilizando un formato de casos de pruebas.

Con respecto a la creación del escenario de pruebas se crearon 4 usuarios correspondientes a cada nivel de usuario (administrador, administrador clínico, odontólogo y asistente) para realizar las pruebas, estos fueron los actores implicados en los eventos de cada caso de prueba.

Documento diseño de pruebas

Casos de prueba

Caso de prueba	***Nombre significativo del caso de prueba que permita identificar el propósito de la prueba.***
Identificador caso de prueba	Identificador único del caso de prueba. La nomenclatura del nombre: CPNNNN_NombreCasoDePrueba. Donde CP corresponde a las siglas de casos de prueba, NNNN corresponde a la numeración única del caso de prueba y el NombeCasoDePrueba corresponde al nombre significativo asignado en el campo caso de prueba.
Función probar	Definir el modulo, servicio o función que probara con el caso de prueba.
Objetivo	Describir que funcionalidad que será probada con el caso de prueba.
Descripción	Describir y explicar el propósito el caso de prueba.
Criterios de éxito	Definir los criterios de aceptación, que permiten determinar que el caso de prueba ejecutado es exitoso

Criterios de falla	Definir los criterios que permiten determinar que el caso de prueba ejecutado es fallido		
Precondiciones	Describir las condiciones y el estado en las que se debe encontrar el sistema para la ejecución del caso de prueba, en caso de ser necesario incluir los casos de pruebas que se deben ejecutar previo al caso de prueba.		
Perfil del usuario	Perfil del usuario en el sistema con el que se ejecutara la prueba.		
Necesidades para el caso de prueba	Definir las necesidades para la ejecución de los casos de pruebas, como por ejemplo los datos de pruebas, las condiciones adicionales a tener en cuenta, configuración de la prueba.		
Autor	Nombre de la persona que diseña el caso de prueba		
Fecha de creación	Fecha en la que se diseña el caso de prueba		
Flujo del caso de prueba	**No paso**	**Usuario del sistema**	**Sistema**
	Orden en el que se ejecuta el paso	Acción del usuario en el sistema, definir las entradas requeridas en el paso y que realiza el usuario durante el paso, en caso que presente entradas, describir que hace el usuario con las entradas.	Respuesta del sistema a la acción realizada por el usuario
Post condiciones	Describir el estado del sistema luego de la ejecución de caso de prueba.		

Tabla 26: Plantilla de Casos de Prueba

La matriz de trazabilidad es un esquema que busca relacionar los distintos casos de pruebas que se pudieron haber realizado con los requerimientos planteados en el análisis del sistema.

ID Caso De Prueba	
ID Requerimiento	X:conector trazabilidad

Tabla 9: Esquema matriz de trazabilidad de pruebas

Caso de prueba Identificador caso de prueba	Registro de Clínica **CP0001**		
Función probar	RegistroClinica.jsp		
Objetivo	Encontrar errores en el registro de nuevas clínicas		
Descripción	Se registró clínicas de prueba en el formulario		
Criterios de éxito	Mensaje de correo electrónico enviado correctamente		
Criterios de falla	Mensaje de error		
Precondiciones	1. Entrar al formulario de registro de clínica 2. Registrar la información requerida por el formulario		
Perfil del usuario	sin usuario		
Necesidades para el caso de prueba	Ingresar la siguiente información necesaria: 1. Nombre de la clínica 2. Nombre propietario 3. Usuario 4. contraseña		
Autor	Brayam Duarte		
Fecha de creación	18/02/2017		
Flujo del caso de prueba	**No paso**	**Usuario del sistema**	**Sistema**
	1	Persona ingresa al sitio web	carga la información de pantalla principal
	2	Presionar el botón de “registrarse”	Carga formulario de registro
	3	Registra información del formulario	Valida los patrones de los campos

Caso de prueba Identificador caso de prueba	Registro de Clínica **CP0001**		
	4	Dar clic en enviar	Muestra mensaje según resultado
Post condiciones	Se registró el nuevo usuario en modo inactivo (hasta activar por medio del correo)		

Tabla 10: Caso de Prueba CP0001

Caso de prueba Identificador caso de prueba	Activación de cuentas **CP0002**		
Función probar	validaruser.jsp		
Objetivo	Encontrar errores en la activaciones de cuentas de usuarios		
Descripción	Se registró clínicas de prueba en el formulario, y se activaron las cuentas desde el link enviado a los correos asociados a las cuentas nuevas		
Criterios de éxito	Mensaje de cuenta activada correctamente		
Criterios de falla	Mensaje de error ¡cuenta no activada!		
Precondiciones	1. Entrar al correo del usuario después de haber registrado la cuenta 2. Entrar al link provisto por el sistema		
Perfil del usuario	sin usuario		
Necesidades para el caso de prueba	1. Correo ingresado correctamente en el formulario de registro		
Autor	Brayam Duarte		
Fecha de creación	18/02/2017		
Flujo del caso de prueba	**No paso**	**Usuario del sistema**	**Sistema**
	1	Usuario registra cuenta de acceso al sistema	Muestra información de correo enviado
	2	Se Dirige al correo	

Caso de prueba Identificador caso de prueba	Activación de cuentas **CP0002**		
	3	Clickea en el link provisto por el sistema dentro del cuerpo del correo	
	4	Es redireccionado a la página temporal de activación de cuenta	Muestra mensaje de cuenta activada
Post condiciones	La cuenta de usuario está habilitada para inicio de sesión en el sistema		

Tabla 27 Caso de Prueba CP0002

Caso de prueba Identificador caso de prueba	Registro de paciente **CP0003**		
Función probar	RegistrarPaciente.jsp		
Objetivo	Encontrar errores en el registro de expedientes para pacientes		
Descripción	Se registró paciente de prueba en el formulario, se ingresaron valores de pruebas		
Criterios de éxito	Mensaje de paciente registrado correctamente		
Criterios de falla	Mensaje de error ¡paciente no registrado!		
Precondiciones	1. Iniciar sesión en el sistema 2. Entrar al registro de pacientes		
Perfil del usuario	Admón. Clínica, odontólogo		
Necesidades para el caso de prueba	1. Ingresar correctamente los parámetros requeridos por el formulario		
Autor	Brayam Duarte		
Fecha de creación	18/02/2017		
Flujo del caso de prueba	**No paso**	**Usuario del sistema**	**Sistema**
	1	Usuario inicia sesión el sistema	Despliega pagina principal

Caso de prueba Identificador caso de prueba	Registro de paciente **CP0003**		
	2	Seleccionar modulo pacientes, registrar paciente	Muestra formulario de registro
	3	Registrar información	Muestra campos requeridos
	4	Presiona en guardar información	Muestra mensaje de registro exitoso
Post condiciones	Nuevo paciente es registrado en la base de datos del sistema		

Tabla 28 Caso de pruebas CP0003

Caso de prueba Identificador caso de prueba	Registrar cita **CP0004**		
Función probar	Calendar.jsp		
Objetivo	Mitigar errores en el registro de citas		
Descripción	Se registró citas de pruebas para validar el correcto registro de la información en las tablas correspondientes		
Criterios de éxito	La cita es visualizada en el calendario		
Criterios de falla	Mensaje de error ¡cita no programada!		
Precondiciones	1. Entrar en el calendario del sistema		
Perfil del usuario	Admón., odontólogo		
Necesidades para el caso de prueba	1. Información de la cita registrada correctamente		
Autor	Brayam Duarte		
Fecha de creación	18/02/2017		
Flujo del caso de prueba	**No paso**	**Usuario del sistema**	**Sistema**
	1	Usuario entra a calendario	Muestra calendario de citas
	2	Da doble click en un bloque del calendario	Muestra formulario de

Caso de prueba Identificador caso de prueba	Registrar cita **CP0004**		
			registro de nueva cita
	3	Introduce la información del formulario	
	4	Presiona el botón guardar	Muestra nueva cita registrada
Post condiciones	La cita es almacenada con estado programada		

Tabla 29 Caso de prueba CP0004

Caso de prueba Identificador caso de prueba	Registrar atención **CP0005**		
Función probar	odonto.jsp		
Objetivo	Mitigar errores en el registro servicios de una atención medica		
Descripción	Se registraron atenciones para varios pacientes		
Criterios de éxito	Se registra el detalle de la atención sin ningún error de fidelidad de datos		
Criterios de falla	Mensaje de error ¡Atencion registrada!		
Precondiciones	2. Entrar en el registro de atenciones del sistema		
Perfil del usuario	Admón., odontólogo		
Necesidades para el caso de prueba	2. Selección de las piezas dentales 3. Registro de los servicios realizados de la atención medica		
Autor	Brayam Duarte		
Fecha de creación	18/02/2017		
Flujo del caso de prueba	**No paso**	**Usuario del sistema**	**Sistema**
	1	Usuario entra a atenciones	Muestra odontograma y formulario de atenciones

Caso de prueba Identificador caso de prueba	Registrar atención **CP0005**		
	2	Selecciona pieza dental	Muestra formulario de servicios y descuentos
	3	Da click en agregar servicio	Agrega fila al detalle de atenciones
	4	Guardar la atención	Manda mensaje de atención almacenada
Post condiciones	La atención queda registrada en el sistema		

Tabla 30 Caso de prueba CP0005

	ID Caso De Prueba				
ID Requerimiento	CP0001	CP0002	CP0003	CP0004	CP0005
RF-1	X	X			
RF-5			X		
RF-10				X	
RF-13					X

Tabla 31 Matriz de trazabilidad casos de pruebas

Implantación del sistema ODONTOWEB

Configuración del servidor

En este acápite se denotarán los pasos que se requirieron para montar y desplegar el sistema en un ambiente de producción. Además del resultado final después de la configuración de los servicios.

Como primer paso se creó un servidor virtual dedicado utilizando Debían 8 como sistema operativo. Se mapeo la interfaz de red utilizando una IP pública con la cual los clientes se conectarán atreves de la internet.

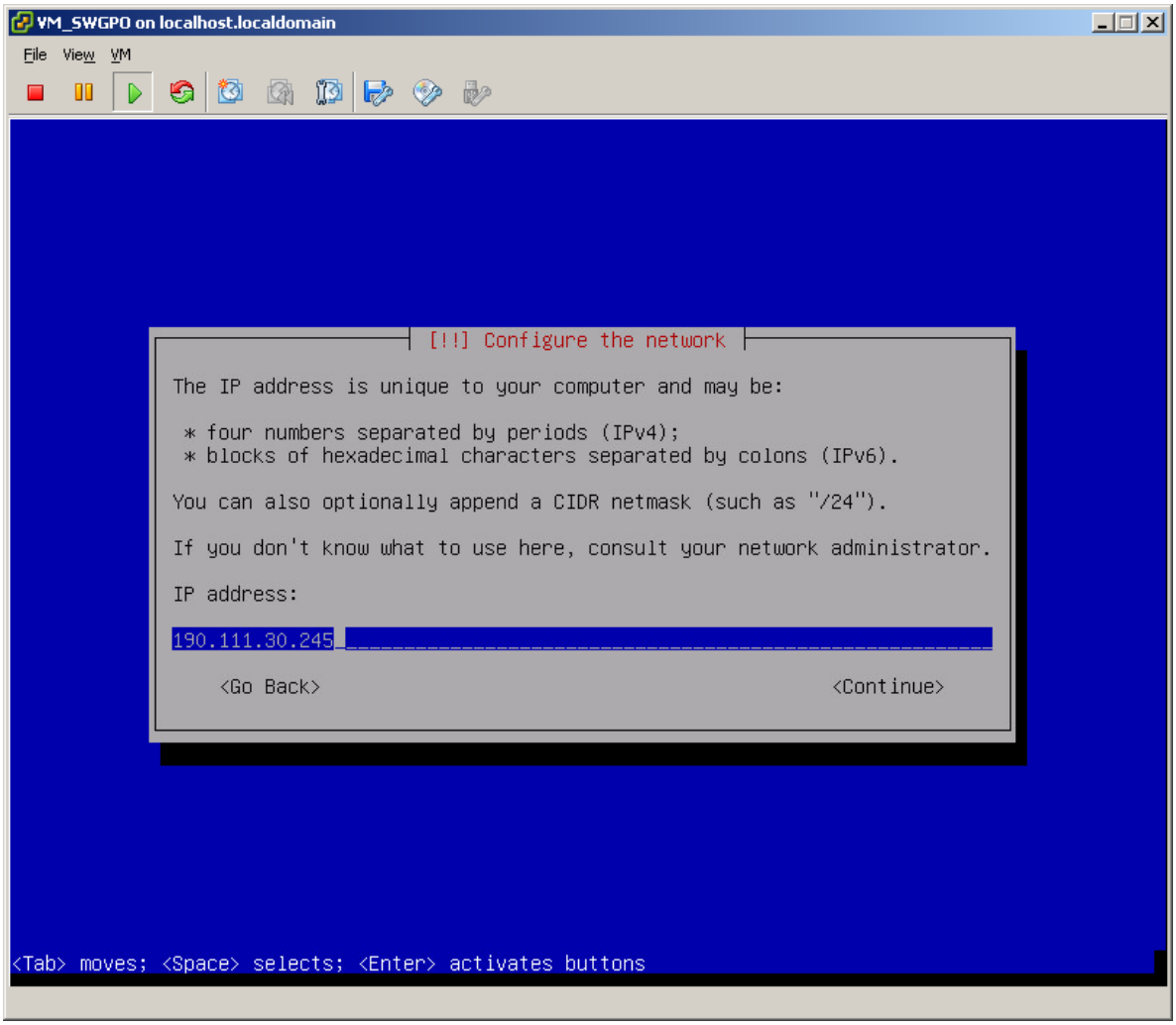


Ilustración 49 IP publica ODONTOWEB

En otro paso muy importante, ya habiendo instalado el sistema operativo y configurado la red, se debe instalar el servidor http en este caso tomcat7.

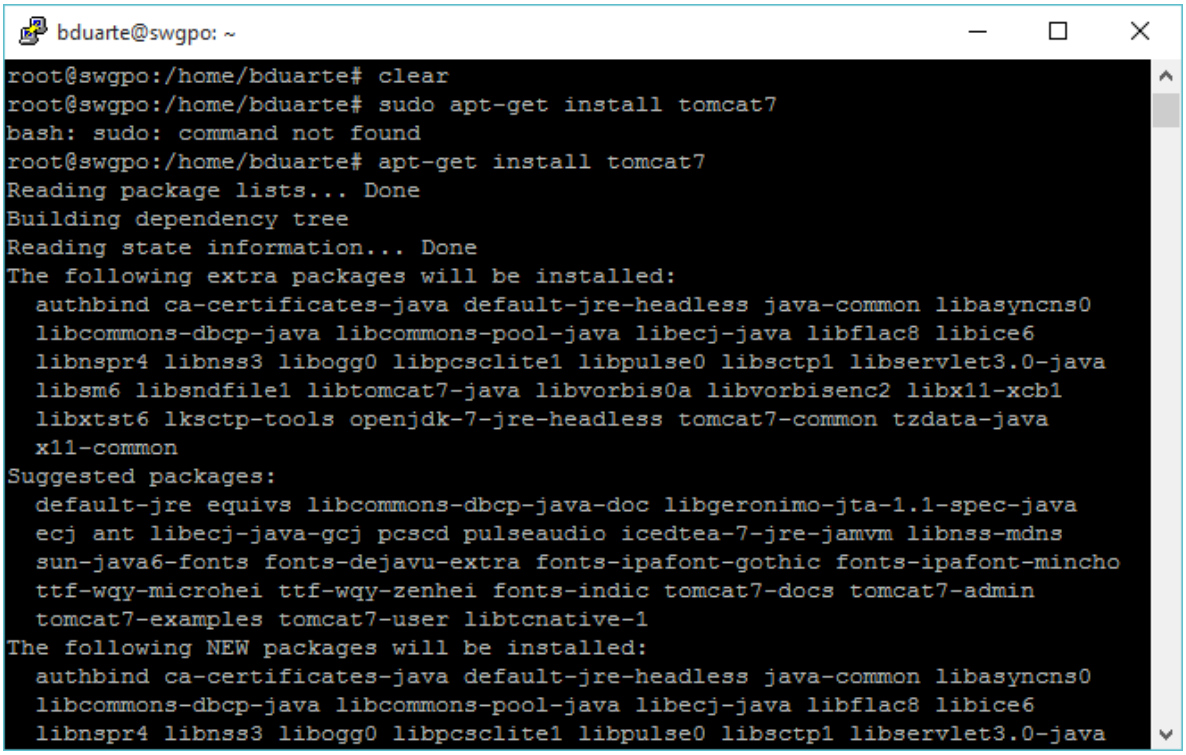


Ilustración 50 Instalación de Tomcat

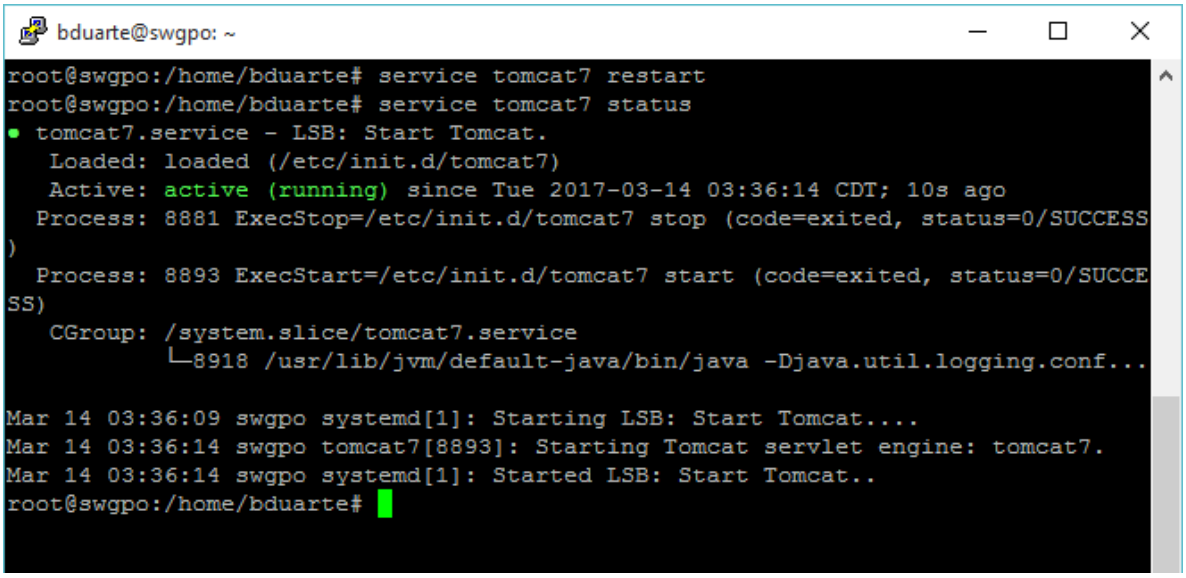


Ilustración 51 Servicio de tomcat corriendo

Por otro lado, uno de los servicios más importes es la base de datos, se instaló, el servidor de Mysql en el cual se creará el schema de la base de datos y seguido de esto se montará el script generado anterior mente con las estructuras de las tablas.

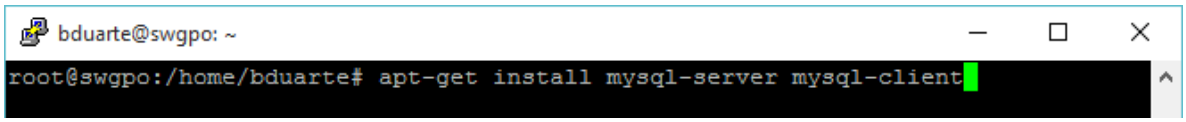


Ilustración 52 Instalación del servidor MySQL

Se creó un usuario y se le dieron todos los permisos para poder gestionar la base de datos del servidor.

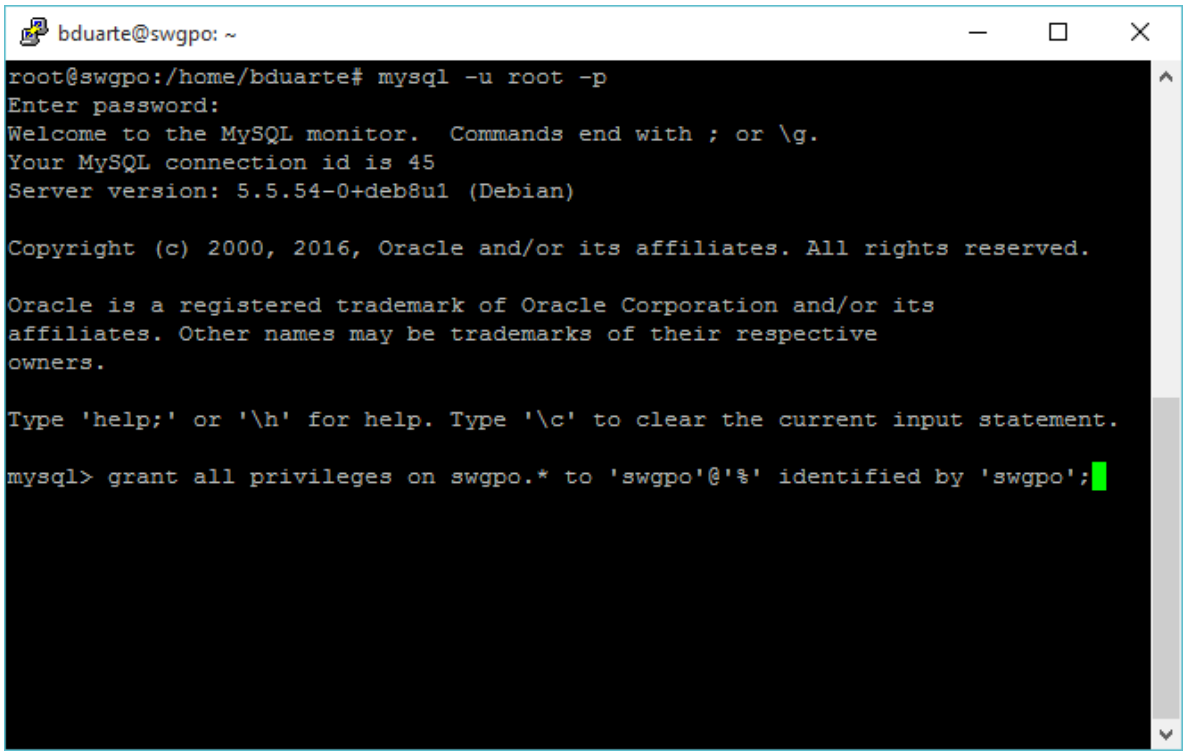


Ilustración 53: otorga privilegios a la base de datos del sistema

Anterior mente se debio crear un script con la estructura de la base de datos, por consihuiente este respaldo es montado en la base de datos creada en el servidor de producción.

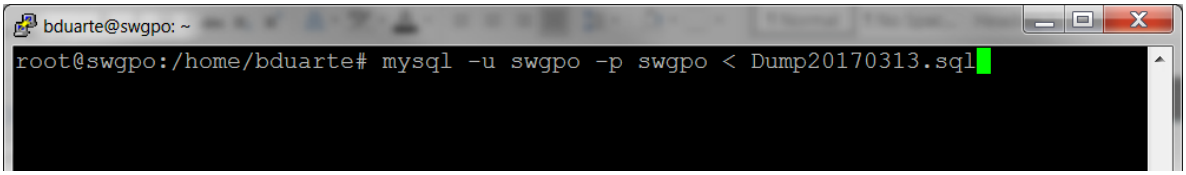


Ilustración 54 Montar estructura de base de datos

Para finalizar debemos pasar el archivo del proyecto al directorio que nos ofrece tomcat para la publicación de aplicaciones /tomcat/webapps

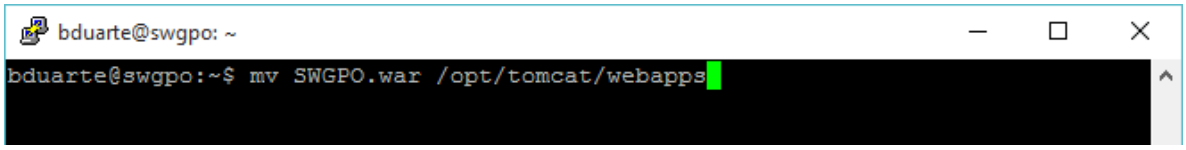


Ilustración 55 copiar archivo del sistema en directorio tomcat

Lluego reiniciamos el servidor tomcat para que ejecute los cambios.

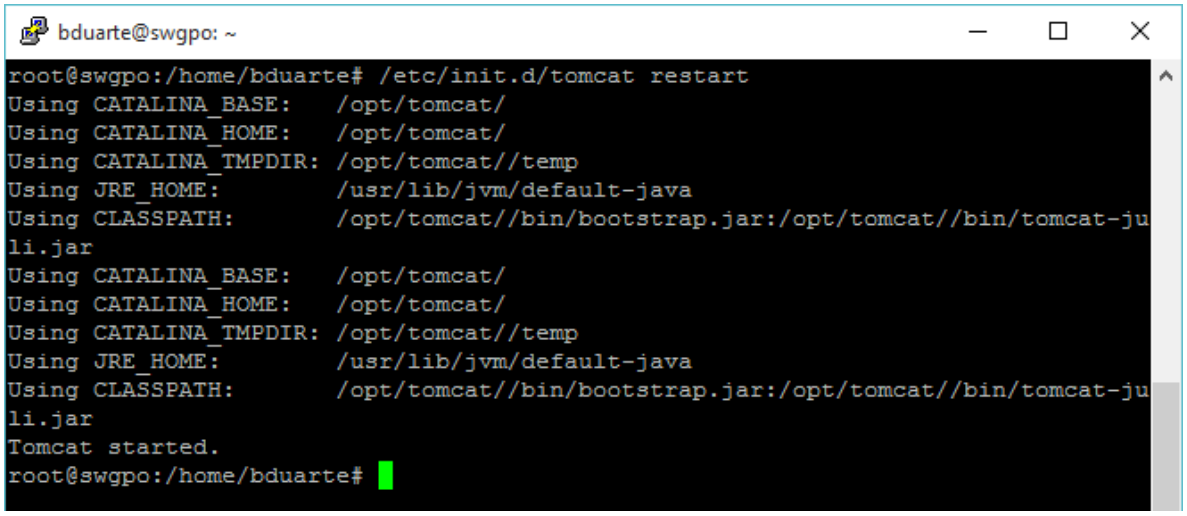


Ilustración 56 Reiniciamos el servidor

En la siguiente imagen podemos observar el sistema corriendo en un servidor virtual con una interfaz pública capaz de conectarse a la WEB. Solamente digitando el siguiente link: 190.111.30.245:8080/SWGPO

Ilustración 57 ODONTOWEB en producción

Manejo de la seguridad ODONTOWEB

Con respecto a la seguridad del ambiente del software se tomaron las siguientes medidas, las cuales denotan aspectos propios de la seguridad informática:

Confidencialidad

A nivel de sitio web, todos los usuarios utilizan contraseñas cifradas, esto con el objetivo de preveer ataques de fuerza bruta, Phishing (suplantación de identidad) o de MitM (man in the middle).

A nivel de base de datos, las contraseñas tienen alta complejidad y solo se utiliza un usuario que posee los permisos necesarios para ejecutar consultas al servidor. En cambio, para proteger el servidor fue necesario utilizar iptables y firewalls que imposibilitan el acceso a personas no autorizadas desde el protocolo ssh. También

se bloquearon los puertos. Para personas que deseen instalar la aplicación en un servidor local es recomendable configurar muy bien el acceso de usuarios al servidor de alojamiento del software, por ejemplo, implementar bloqueos de puertos, como buena práctica de seguridad.

Integridad

Para este aspecto se tomó en cuenta la validez de la información ingresada por los usuarios desde el sitio, validándose desde los formularios de ingreso de datos de acuerdo al dominio especificado en las tablas de la base de datos, además de la implementación de transacciones para la inserción de información en procesos que incluyen más de una tabla, esto evita errores de inconsistencias en los datos.

Disponibilidad

La publicación Demo del sistema tiene una disponibilidad de 24/7, gracias a su alojamiento en un Data Center que brinda los servicios de respaldo de información y restauración inmediata, ancho de banda suficiente para recibir numerosas peticiones web al mismo tiempo, sin riesgos de caídas por agentes ambientales o falta energética.

En el caso de adquisición del software e instalación local, deben ellos mismo velar por la seguridad y disponibilidad de sus datos, asumiendo los costos que esto representa.

ESTIMACIÓN DE COSTOS

El cálculo del costo del sistema presentado a continuación, hace uso de la métrica de Puntos de Función, la cual permite traducir en un número el tamaño de la funcionalidad que brinda el sistema desde el punto de vista del usuario, a través de una suma ponderada de las características del mismo.

Elementos de Función (EF)		Descripción
EI	Entrada externa	Interfaz donde el usuario ingresa datos
EO	Salida externa	Informes, Gráficos, listados de datos
EQ	Consulta	Recuperación de datos del sistema (Búsquedas)
ILF	Archivo lógico interno	Archivos de almacenamiento de datos
EIF	Archivo de interfaz externo	Datos referenciados desde otros sistemas

Tabla 32 Detalle de EF

En la siguiente tabla se presenta la clasificación de cada requerimiento funcional del sistema según el EF correspondiente:

Código requerimiento funcional	Clasificación EF	Complejidad (Alta, Media, Baja)
RF-1	EI	M
RF-2	EI	M

Código requerimiento funcional	Clasificación EF	Complejidad (Alta, Media, Baja)
RF-3	EI	M
RF-4	EI	M
RF-5	EI	A
RF-6	EQ	M
RF-7	EI	M
RF-8	EO	A
RF-9	EI	M
RF-10	EI	M
RF-11	EO	A
RF-12	EI	M
RF-13	EI	A
RF-14	EI	A
RF-15	EO	A
RF-16	EI	M
RF-17	EO	M
RF-18	EI	A
RF-19	EO	A
RF-20	EO	A
Totales: EI=13, EQ=1, EO=6		

Tabla 33 Clasificación R. F. según EF

El valor ILF correspondería según el estándar, al número de tablas de la base de datos que utiliza el sistema las cuales serían 34. El valor de los EIF sería de 0 ya que el sistema no se comunica o no utiliza información de otra aplicación externa. A continuación; teniendo los datos de entrada se deben calcular los puntos de función sin ajustar (PFSA), los cuales son determinados al someter los EF a un estándar impuesto por International Function Point Users Group (IFPUG).

Tipo / Complejidad	Baja	Media	Alta
(EI) Entrada externa	3 PF	4 PF	6 PF
(EO) Salida externa	4 PF	5 PF	7 PF
(EQ) Consulta externa	3 PF	4 PF	6 PF
(ILF) Archivo lógico interno	7 PF	10 PF	15 PF
(EIF) Archivo de interfaz externo	5 PF	7 PF	10 PF

Tabla 34 Estándar IFPUG para calcular (PFSA)

Tipo / Complejidad	Baja	Media	Alta	Totales
(EI) Entrada externa	0x3 PF	9x4 PF	4x6 PF	60 PF
(EO) Salida externa	0X4 PF	1X5 PF	5X7 PF	40 PF
(EQ) Consulta	0X3 PF	1X4 PF	0X6 PF	4 PF
(ILF) Archivo lógico interno	0X7 PF	43X10 PF	0X15 PF	430 PF
(EIF) Archivo de interfaz externo	0X5 PF	0X7 PF	0X10 PF	0
			Total:	**534 PF**

Tabla 35 Calculo de PFSA

Habiendo calculado los PFSA se deben de llegar a los PFA (Puntos de función ajustados) utilizando un factor de ajuste para poder realizar el cálculo final.

Factor de ajuste	Puntaje
Comunicación de Datos	3
Procesamiento Distribuido	4
Objetivos de rendimiento	0
Configuración del equipamiento	1
Tasa de transacciones	0
Entrada de datos en línea	5
Interface con el usuario	4
Actualizaciones en línea	3
Procesamiento complejo	2
Reusabilidad del código	1
Facilidad de implementación	1
Facilidad de operación	0
Instalaciones múltiples	0
Facilidad de cambios	2
Factor de Ajuste:	**26**

Tabla 36 Factor de ajuste PF

Para calcular los PFA se debe resolver la siguiente ecuación:

$$PFA = PFSA * [0.65 + (0.01 * Factor\ Ajuste)]$$

Ingresando los valores, entonces la expresión quedaría de la siguiente manera:

$$PFA = 534 * [0.65 + (0.01 * 26)]$$

$$PFA = 485.94 \rightarrow 486$$

Para la estimación del esfuerzo se utiliza una tabla estándar que brinda un promedio de horas trabajadas según el tipo de lenguaje de implementación. En el caso de ODONTOWEB fue desarrollado en JAVA un lenguaje de 4ta generación.

Lenguaje	Horas PF promedio	LDC por PF
Ensamblador	25	300
COBOL	15	100
Lenguajes 4ta Generación	**8**	**20**

Tabla 37 Horas PF promedio por lenguaje de programación

La ecuación de esfuerzo es la siguiente:

$$\frac{H}{H} = PFA * Horas\ PF\ Promedio$$

Rellenando las variables el resultado sería:

$\frac{H}{H} = 486 * 8 = 3{,}888$ Horas hombre

$$\frac{H}{H} = 3{,}888\ Horas\ Hombre \rightarrow\ 5{,}33\ Meses\ Hombres$$

Para obtener el costo de desarrollo del software se escaló el tiempo a 6 meses, para incluir el esfuerzo en instalación y configuración de las herramientas de desarrollo.

Costo de Desarrollo

Costos de desarrollo				
Recursos	**Rol**	**Salario Mensual $US**	**Cantidad de meses**	**Salario Total $US**
1	Desarrollador	600.00	6	3,600.00
4	Tester	0.00	1	0.00
Total				**$3,600.00**

Tabla 38 Costos de Desarrollo

Otros Costos

Costos de inversión inicial - Hardware			
Cant.	**Descripción**	**Depreciación $US**	**Precio total $US**
1	Laptop Dell Inspiron	250.00	250.00
1	Pc Intel dh55hc	200.00	200.00
1	Servidor Dell	0.00	0.00
Total			**450.00**

Tabla 39: Inversión Inicial - HW

Costos de inversión inicial - Software			
Cant.	**Descripción**	**Precio unitario $US**	**Precio total $US**
1	Eclipse	0.00	0.00
1	MySql Server	0.00	0.00
1	Jquery	0.00	0.00
1	Bootstrap3	0.00	0.00
1	Github	0.00	0.00
Total			**$0.00**

Tabla 40 Costo de Inversión Inicial – SW

Costo de inversión inicial - Comunicaciones			
Cant.	**Descripción**	**Precio unitario $US**	**Precio total $US**
1	IP pública	0.00	0.00
Total			**$0.00**

Tabla 41 Costo de Inversión Inicial – Comunicaciones

Costos complementarios

Costos complementarios			
Cant.	**Descripción**	**Precio unitario $US**	**Precio total $US**
2	Tinta impresora tinta continua	10.00	20.00
1	Resma de papel tamaño carta	5.00	5.00
Total			**$25.00**

Tabla 42: Costo complementarios

Costo del software

Costo del proyecto	
Descripción	**Total $US**
Costos de inversión inicial	450.00
Costos de desarrollo	3,600.00
Costos complementarios	25.00
Total	**$4,075.00**

Tabla 43: Costo del software

El costo de instalación y soporte del software se proyecta a $1,100.00, partiendo de la premisa de garantizar un mínimo de 4 clientes. A partir del quinto cliente se obtendrán utilidades.

Para los usuarios que deseen adquirir el software y decidan implantarlo en un servidor local, se deberá sumar sus costos de inversión en equipos y servicios de configuración, los cuales no están incluidos en el cálculo anterior.

MODELO DE NEGOCIO

ODONTOWEB estará desplegado como un servicio gratuito en el internet, pero como un demo de registros limitados, permitiendo así mostrar el potencial del sistema. En esta versión pública del sistema, los interesados podrán registrarte, probar la funcionalidad antes de decidir sobre la adquisición del software. En la web se mostrarán los datos del contacto para adquirir el software, así como también el catálogo de servicios.

El producto software y código fuente se puede obtener desde un repositorio en git (Presente en el demo de la web) totalmente gratuito, gracias a la implementación de licencias GNU. La obtención de soporte, instalación de la web app en algún ambiente local y el desarrollo de nuevos módulos, serán tratados como servicios al cliente (o interesado), por medio de contratos.

Se plantea el siguiente esquema para la distribución de ODONTOWEB como servicio:

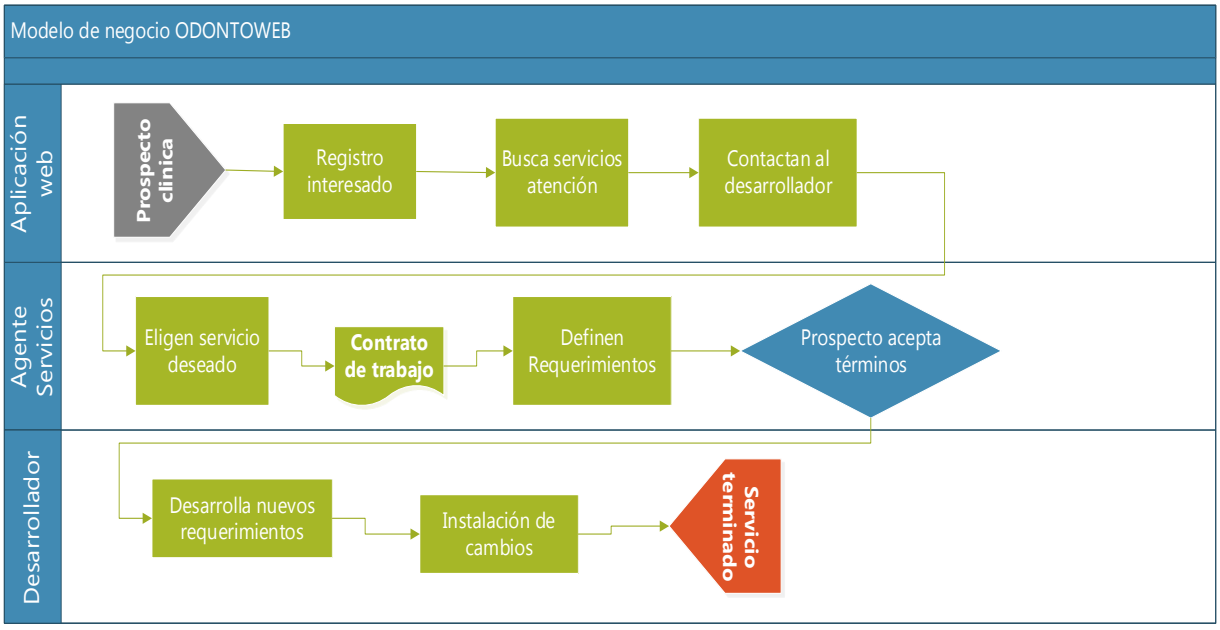


Ilustración 58 Modelo de negocio ODONTOWEB

CONCLUSIONES Y RECOMENDACIONES

ODONTOWEB es un sistema web que funciona como herramienta para el registro de expedientes y gestión de citas de pacientes odontológicos, lo que da por satisfecho el objetivo general de este trabajo monográfico.

El software también posee un sistema de notificaciones de cambio y recordatorios de citas por medio de correo electrónico. Asimismo, incluye reportes de mucha utilidad para la administración de la clínica.

ODONTOWEB incluye odontogramas que muestran de forma gráfica la historia clínica de la dentadura del paciente, facilitándole al odontólogo un rápido análisis del estado de la dentadura y comparaciones entre el estado bucal actual y el registrado en la visita anterior. Las diversas patologías y tratamientos, así como las ausencias, se señalan en el odontograma, mediante símbolos y códigos de colores totalmente familiar para los profesionales de la salud bucal (ver Simbología del Odontograma).

La metodología UWE y el UML fueron de mucha utilidad para la definición y elaboración de los entregables necesarios para construir el sistema desde cero, por lo que se considera que – tanto funcionalmente como metodológicamente – el producto de este trabajo monográfico cumple con los objetivos académicos y funcionales planteados.

Como recomendaciones para futuros desarrolladores que quieran implementar nuevos módulos al sistema sería magnífico que incluyan un odontograma con modelado 3D utilizando herramientas como three.js. También se recomienda incorporar el control de recursos y materiales, lo que ayudaría mucho a la clínica en cuanto al abastecimiento de los mismos.

BIBLIOGRAFÍA

AUMAILLE, B. (2002). *J2EE Desarrollo de aplicaciones web.* Barcelona: P° Ferrocarriles Catalanes.

Ben Collins-Sussman, B. W. (2011). *Version Control with Subversion.* California.

CAROL ANNE MURDOCH-KINCH, MARY ELLEN McLEAN. (2003). Minimally invasive. *PRACTICAL SCIENCE*, 95.

Definicion.de. (2015). *Definicion.de*. Obtenido de Definicion.de: http://definicion.de/odontograma/

Fulton, S. F. (2013). *HTML5 Canvas.* Gravenstein Highway North: O'Reilly Media, Inc.

Galiano, L. (2012). *INFORME DE LA METODOLOGÍA .* Bolivar.

Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y javascript.* Barcelona: MARCOMBO, S.A.

Martin Fowler, K. S. (1999). *UML gota a gota.* Pearson Educación, 1999.

Mora, S. L. (2002). *Programacion de aplicaciones web: historia, principios basicos y clientes web.* San Vicente(Alicante): Club Universitario.

Pressman, P. R. (2006). *Ingenieria del software un enfoque práctico.* Connecticut: McGrawHill.

Smith, B. (2014). *A Quick Guide to GPLv3*. Obtenido de www.gnu.org: https://www.gnu.org/licenses/licenses.es.html#GPL

THIBAUD, C. (2006). *MYSQL 5.* Barcelona: Ediciones ENI.

ANEXOS

Diccionario de datos

Tabla: Abono

Columna	Tipo	Null	Llave	Descripción
idAbono	int(11)	NO	PRI	identificador tabla abono
idCredito_abo	int(11)	YES	MUL	identificador del crédito donde se realizará el abono
Valor	float	YES		Valor del abono
FechaRegistro	timestamp	YES		Fecha de registro del abono
SaldoAnterior	float	YES		Saldo anterior al abono
SaldoActual	float	YES		saldo actual junto al abono
idTipoDocumento	varchar(3)	YES	MUL	identificador del tipo de documento

Tabla: Atencion

Columna	Tipo	Null	Llave	Descripción
idAtencion	int(11)	NO	PRI	Identificador atención
idOdontologo	int(11)	NO	MUL	Identificador del odontólogo
idPaciente	int(11)	NO	MUL	Identificador del paciente
FechaAtencion	timestamp	YES		Fecha de registro
Descripcion	varchar(100)	YES		Descripción de la atención
idTipoDocumento	varchar(3)	NO	MUL	Identificador del tipo de documento
Valor	float	YES		Valor total de la atencion

Tabla: Cita

Columna	Tipo	Null	Llave	Descripción
idCita	int(11)	NO	PRI	Identificador cita
PacienteC_idPaciente	int(11)	YES	MUL	Identificador paciente
OdontologoC_IdOdontologo	int(11)	YES	MUL	Identificador del odontólogo
start_date	timestamp	YES		Fecha y hora de comienzo de cita

Columna	Tipo	Null	Llave	Descripción
end_date	timestamp	YES		Fecha y hora de conclusión
text	varchar(150)	YES		Descripción de la cita
EstadoC_idEstado	int(11)	YES	MUL	Estado de la cita

Tabla: Clinica

Columna	Tipo	Null	Llave	Descripción
idClinica	int(11)	NO	PRI	Identificador de la clínica
Nombre	varchar(45)	YES		Nombre de la clínica
Siglas	varchar(45)	YES		Siglas de la clínica
Propietario	varchar(60)	NO		Propietario de la clínica
Telefono	varchar(45)	YES		Teléfono
Direccion	varchar(45)	YES		Dirección
Ciudad	varchar(45)	YES		Ciudad
RUC	varchar(45)	NO		Numero RUC
idUsuario	int(11)	YES	MUL	Identificador del usuario
FechaRegistro	timestamp	YES		Fecha de registro

Tabla: Cotizacion

Columna	Tipo	Null	Llave	Descripción
idcotizacion	int(11)	NO	PRI	Identificador de la cotización
idOdontologo	int(11)	NO	MUL	Identificador del odontólogo
idPaciente	int(11)	NO	MUL	Identificador del paciente
FechaCotizacion	timestamp	YES		Fecha de la cotización
Descripcion	varchar(100)	YES		Descripción de la cotización
idTipoDocumento	varchar(3)	NO	MUL	Identificador del tipo de documento
Valor	float	YES		Valor de la cotización
idEstado	int(11)	NO		Estado de la cotización

Tabla: Credito

Columna	Tipo	Null	Llave	Descripción
idcredito	int(11)	NO	PRI	Identificador del crédito
idPaciente_cre	int(11)	YES	MUL	Identificador del paciente
FechaRegistro	timestamp	YES		Fecha de registro del crédito
Limite	float	YES		Monto del límite de crédito
Plazo	int(11)	YES		Días de plazo para que el crédito este vencido
idEstado_cre	int(11)	YES	MUL	Estado del crédito
idTipoDocumento	varchar(3)	YES	MUL	Identificador tipo documento
Descripcion	varchar(90)	YES		Descripción del crédito

Tabla: detalle_atencion

Columna	Tipo	Null	Llave	Descripción
idDetalleAtencion	int(11)	NO	PRI	Id del detalle atención
idAtencion	int(11)	YES		Id de la atención
idDiente	int(11)	YES	MUL	Id del diente
idServicio	int(11)	YES	MUL	Id del servicio
idDiagnostico_Sitio	int(11)	YES	MUL	Control de diagnóstico sitio
idDiagnostico_Estado	int(11)	YES	MUL	Control de diagnóstico estado
Realizado	bit(1)	YES		El servicio fue realizado
Descuento	float	YES		Descuento del usuario
Precio	float	YES		Precio del servicio
Total	float	YES		Total del detalle

Tabla: detalle_cotizacion

Columna	Tipo	Null	Llave	Descripción
iddetalle_cotizacion	int(11)	NO	PRI	Id del detalle atención
idCotizacion	int(11)	YES	MUL	Id de la atención
idDiente	int(11)	YES	MUL	Id del diente
idServicio	int(11)	YES	MUL	Id del servicio
idDiagnostico_Sitio	int(11)	YES	MUL	Control de diagnóstico sitio

Columna	Tipo	Null	Llave	Descripción
idDiagnostico_Estado	int(11)	YES	MUL	Control de diagnóstico estado
Realizado	bit(1)	YES		El servicio fue realizado
Descuento	float	YES		Descuento del usuario
Precio	float	YES		Precio del servicio
total	varchar(45)	YES		Total del detalle

Tabla: Diente

Columna	Tipo	Null	Llave	Descripción
idDiente	int(11)	NO	PRI	Identificador pieza dental
Zona	varchar(60)	YES		Zona de la pieza dental
NombreDiente	varchar(45)	YES		Nombre del diente
axis_x	float	YES		Coordenada x del canvas
axis_y	float	YES		Coordenada y del canvas
RutaImg	varchar(45)	YES		Ruta donde está alojada la imagen

Tabla: foto

Columna	Tipo	Null	Llave	Descripción
idFoto	int(11)	NO	PRI	Identificador foto
FotoName	varchar(45)	YES		Nombre del archivo
foto	longblob	YES		Binarios del archivo
Extension	varchar(45)	YES		Tipo de archivo
PacienteF_idPaciente	int(11)	NO	MUL	Código del paciente
FechaRegistro	timestamp	YES		Fecha de registro

Tabla: Odontograma

Columna	Tipo	Null	Llave	Descripción
idOdontograma	int(11)	NO	PRI	Identificador del odontograma
PacienteO_idPaciente	int(11)	YES	MUL	Identificador del paciente
FechaRegistro	timestamp	YES		Fecha de registro

Tabla: Detalle_Odontograma

Columna	Tipo	Null	Llave	Descripción
idDetalleOdontograma	int(11)	NO	PRI	Identificador del detalle odontograma
idOdontograma	int(11)	NO	MUL	Identificador del odontograma
idDiente	Int(11)	NO	MUL	Identificadode de la pieza dental
idEstado	Int(11)	No	Mul	Identificador del estado de la pieza dental

Tabla: odontologo

Columna	Tipo	Null	Llave	Descripción
idOdontologo	int(11)	NO	PRI	Identificador odontólogo
Onombre1	varchar(45)	YES		Primer nombre odontolofo
Onombre2	varchar(45)	YES		Segundo nombre odontólogo
Oapellido1	varchar(45)	YES		Primer apellido odontólogo
Oapellido2	varchar(45)	YES		Segundo apellido odontólogo
Direccion	varchar(45)	YES		Dirección del odontólogo
Telefono	varchar(45)	YES		Teléfono
Movil	varchar(45)	YES		Móvil
CodigoDoc	varchar(45)	YES		
idEstado	int(11)	NO	MUL	Estado del odontólogo
FechaRegistro	timestamp	YES		Fecha de registro
idUsuario	int(11)	NO	MUL	Identificador del usuario
idClinica	int(11)	NO	MUL	Identificado de la clínica ala que el pertenece

Tabla: Paciente

Columna	Tipo	Null	Llave	Descripción
IdPaciente	int(11)	NO	PRI	Identificador del paciente
Pnombre1	varchar(45)	YES		Primer nombre del paciente
Pnombre2	varchar(45)	YES		Segundo nombre del paciente

Columna	Tipo	Null	Llave	Descripción
Papellido1	varchar(45)	YES		Primer apellido del paciente
Papellido2	varchar(45)	YES		Segundo nombre del paciente
telefono	varchar(45)	YES		Teléfono del paciente
movil	varchar(45)	YES		Móvil del paciente
companySeguro	varchar(45)	YES		Compañía de seguro
NumeroSeguro	varchar(45)	YES		Numero de asegurado
Estado_idEstado	int(11)	NO	MUL	Estado del paciente
GrupoSanguineo	varchar(45)	YES		Tipo de sangre
FechaNac	date	YES		Fecha de nacimiento
NTarjeta	varchar(45)	YES		Número de tarjeta
Npoliza	varchar(45)	YES		Numero de póliza
Observacion	varchar(80)	YES		Observación de salud
Email	varchar(45)	YES		Correo electrónico
Ciudad	varchar(45)	YES		Ciudad del paciente
Sexo	varchar(45)	YES		Sexo del paciente
Direccion	varchar(100)	YES		Dirección del paciente
idUsuario	int(11)	YES	MUL	Identificador del usuario
idClinica	int(11)	YES	MUL	Clínica donde pertenece el paciente

Tabla: Servicio

Columna	Tipo	Null	Llave	Descripción
idServicio	int(11)	NO	PRI	Identificador del servisio
Snombre	varchar(45)	YES		Nombre del servicio
Descripcion	varchar(150)	YES		Descripción del servicio
Precio	float	YES		Precio del servicio
moneda	varchar(45)	YES		Tipo de moneda
idTipoServicio	int(11)	YES	MUL	Tipo de servicio
IdOdontologo	int(11)	YES	MUL	

idEstado	int(11)	YES	MUL	Estado del servicio
idClinica	int(11)	YES	MUL	Identificador de la clinica

Tabla: Usuario

Columna	Tipo	Null	Llave	Descripción
idUsuario	int(11)	NO	PRI	Identificador del usuario
Login	varchar(45)	YES		Nombre de usaurio
Password	varchar(45)	YES		Contraseña de acceso
FechaRegistro	timestamp	YES		Fecha en que fue registrado el usuario
OdontologoU_idOdontologo	int(11)	YES	MUL	Odontólogo
TipoU_idTipoUser	int(11)	YES	MUL	Tipo de usuario
EstadoU_IdEstado	int(11)	NO	MUL	Estado del usuario
Email	varchar(60)	YES		Correo del usuario

Simbología del Odontograma

Los odontogramas forman parte imprescindible de la historia clínica odontológica del paciente.

La simbología a utilizar es la que se expone a continuación:

- Para las ausencias dentales se utiliza una cruz de color azul en el lugar correspondiente de la pieza. Si la pieza debido al pronóstico requiere una extracción se señala con una cruz roja sobre la corona de ésta, y cuando se haya extraído se traza una cruz negra.
- Cuando existe caries, se indica con un punteado rojo en la zona afectada.
- Cuando la pieza tiene una obturación previa, se señala con un sombreado azul, pero si presentara recidiva de caries, es decir, que requiere un nuevo tratamiento de obturación, se limita la zona con un círculo de color rojo.
- Cuando existe una endodoncia previa, se utiliza una cruz azul, pero en este caso en la raíz, o bien se escribe una E azul. Y cuando hay que realizar una endodoncia la cruz sería de color rojo; una vez realizada la endodoncia se pone por encima una cruz negra.
- Cuando existe una corona se señala con un círculo azul alrededor de la corona de la pieza.
- Cuando hay que poner un puente fijo, las coronas de los dientes pilares se rodean de un círculo rojo (indicando que será necesario tallarlos) y el póntico se dibuja de azul (la/s pieza/s que faltan y la línea de unión).
- Si las pruebas radiográficas ponen de manifiesto la existencia de una caries, ésta de colorea de verde.
- Las piezas que comienzan a erupcionar se señalan con una flecha azul hacia arriba, si está situada en la arcada inferior o hacia abajo si está en la arcada superior.
- Cuando una pieza ha sufrido un traumatismo se señala con una T.

A continuación, observamos un ejemplo de aplicación de los códigos de colores del odontograma:

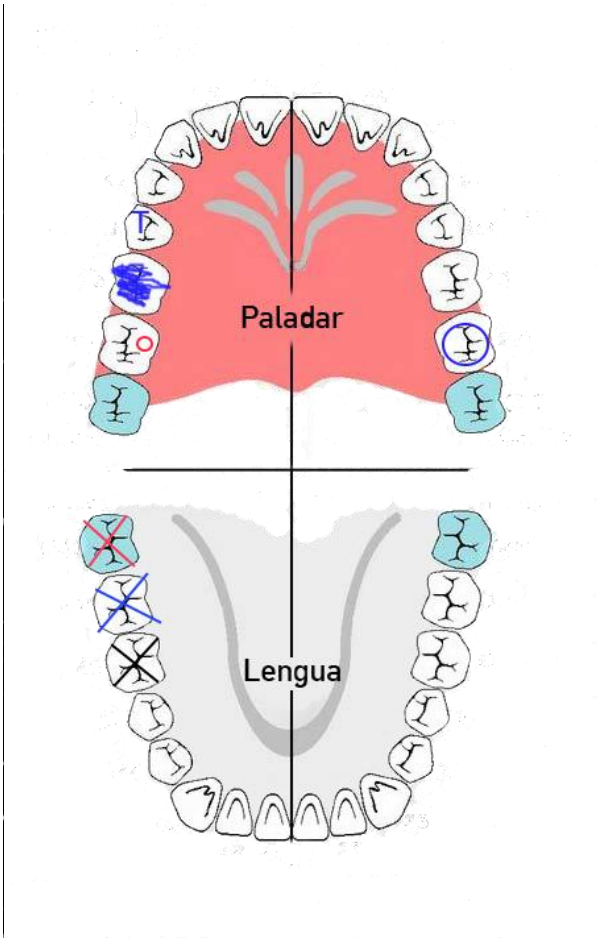


Ilustración 59 odontograma con estados