**Documento de requisitos**

**“OdontOffice” Sistema de administración dental**

**Versión 0.4**

**Integrantes**

* Acosta Oscar
* Barraza José Ramón
* Gastelum Charly
* Guerrero Guadalupe
* Reatiga Rigoberto
* Sánchez Edder.

Culiacán, Sinaloa 13 de Abril 2015

**ÍNDICE**

[PREFACIO](#h.733rp862dvai)

[HISTORIA DEL DOCUMENTO](#h.m1kxcunhwr9c)

[1.INTRODUCCIÓN](#h.ooe5n0bomqdx)

[1.1 PROPÓSITO](#h.y6cq12tuao70)

[1.2 ALCANCE](#h.dyca4gf3iatt)

[1.3 CONTEXTO](#h.uxnnsz1cn1ls)

[1.4 REFERENCIAS](#h.y9uoyi8j1pi7)

[2. REQUISITOS DEL SISTEMA](#h.cufvcjlsrvin)

[2.1. REQUISITOS FUNCIONALES](#h.2qjxlh21qqht)

[2.2 REQUISITOS NO FUNCIONALES](#h.1jbqbmj18p56)

[2.3 REQUISITOS DE INTERFAZ](#h.izsvaohn0iip)

[2.4 ACTORES Y ROLES DEL SISTEMA](#h.djxw3c8wn3vm)

[3. CASOS DE USOS](#h.e1b09f3xuept)

[3.1. MODELO DE CASO DE USO.](#h.qf93e5o6g0m1)

[3.2. ESPECIFICACIÓN DE LOS CASOS DE USO.](#h.3x977f8zz43j)

[4. ARQUITECTURA](#h.dsmytntpxe5l)

[4.1 COMPONENTES DE LA ARQUITECTURA](#h.elztbp3kms80)

[4.1.1 Capa de clientes:](#h.z3jvt081mb1m)

[4.1.2 Capa de presentación:](#h.9t4izvn53gqm)

[4.1.3 Capa de lógica de negocios:](#h.oe1q5lqp3dqq)

[4.1.4 Capa de datos:](#h.aibr03pasj5)

[4.2 RESPONSABLES DE LOS COMPONENTES DE LA ARQUITECTURA](#h.j5rf1hpzonoi)

[5. FURPS](#h.71r5247jgzjn)

[6. VISTAS](#h.6uuoz9p3p0gr)

#### PREFACIO

|  |  |
| --- | --- |
|  | El presente es el documento de requisitos del proyecto “OdontOffice” el cual tiene como función principal la organización de información en una base de datos para la consulta y actualización de esta, la calendarización de citas de los pacientes y la creación de estadísticas de avance. |
| **Alcance del documento** | El documento de requisitos es la base de trabajo para el proyecto, aquí se presentan cuestiones relacionadas con el proyecto , tales como los requisitos funcionales y no funcionales , la arquitectura del sistema, objetivos , riesgos y problemas que se pueden presentar en dicho proyecto. |
| **Documentos relacionados** | No hay documentos relacionados. |
| **Autores** | Acosta Mendoza Oscar, Barraza Duarte José Ramón, Gastelum Quintero Charly Michael, Guerrero Gutiérrez Guadalupe de Jesús, Reatiga Payan Rigoberto, Sánchez López Edder José. |
| **Lectores** | Este documento es de gran interés para todo personal relacionado con el proyecto, en especial personal de desarrollo y mantenimiento del producto. |

#### HISTORIA DEL DOCUMENTO

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Explicación del cambio | Autor |
| 0.1 | 10/03/2015 | Primer Borrador | Acosta Oscar, Barraza José Ramón, Gastelum Charly, Guerrero Guadalupe, Reatiga Rigoberto, Sánchez Edder. |
| 0.2 | 12/03/2015 | Segundo Borrador | Barraza José Ramón, Gastelum Charly, Guerrero Guadalupe, Reatiga Rigoberto, Sánchez Edder. |
| 0.3 | 23/03/2015 | Tercer Borrador | Acosta Oscar, Barraza José Ramón, Gastelum Charly, Guerrero Guadalupe. |
| 0.3.1 | 24/03/2015 | Correcciones Tercer Borrador | Acosta Oscar, Barraza José Ramón, Guerrero Guadalupe, Reatiga Rigoberto, Sánchez Edder. |
| 0.4 | 22/04/2015 | Cuarto Borrador | Barraza José Ramón, Gastelum Charly, Guerrero Guadalupe, Reatiga Rigoberto, Sánchez Edder. |
|  |  |  |  |
|  |  |  |  |

# 1.INTRODUCCIÓN

#### 1.1 PROPÓSITO

El proyecto busca agilizar el proceso de registro y control de pacientes (datos generales, diagnóstico odontomedico y tratamientos odontológicos realizados), así como las citas programadas para los tratamientos dentales correspondientes y estados de cuenta por los servicios hechos en la facultad, así mismo realice estadísticas de avance, donde los alumnos puedan consultar la información y avances de sus pacientes y así mismo puedan realizar estadísticas de avance y trabajos realizados.

#### 1.2 ALCANCE

|  |
| --- |
| **El proyecto incluye** |
| Inserción, eliminación y Actualización de datos de pacientes. |
| Impresión de documentos. |
| Registro de citas diarias. |
| Consulta de saldos generados por los tratamientos. |
| Búsqueda de expedientes de pacientes. |
| Consulta de pacientes por fechas. |

#### 1.3 CONTEXTO

Este software se está desarrollando en JAVA y se trabaja usando el IDE NETBEANS 8.0, como manejador de base de datos usamos PostgreSQL, se están reutilizando componentes desarrollados en proyectos anteriores por nuestro equipo de desarrollo.

#### 1.4 REFERENCIAS

* Documentación oficial

[**http://docs.oracle.com/javase/7/docs/api/**](http://docs.oracle.com/javase/7/docs/api/) **Oracle obtenido el 11/03/2015**

[**http://www.postgresql.org.es/documentacion**](http://www.postgresql.org.es/documentacion) **PostgreSQL obtenido el 11/03/2015**

# 2. REQUISITOS DEL SISTEMA

#### 2.1. REQUISITOS FUNCIONALES

RF1: El sistema debe de iniciar solo cuando el usuario lo ejecute, debe requerir nombre de usuario y contraseña para entrar.

RF2: El sistema permite guardar información en la base de datos a cualquier usuario del sistema, siempre y cuando estén completos ciertos campos del formulario de registro de pacientes.

RF3: El sistema permite que los usuarios consulten la información de los pacientes.

RF4: El sistema permite que cualquier usuario dado de alta edite información de los pacientes.

RF5: El sistema permite la eliminación de información a usuarios con privilegios de administración.

RF6: El sistema cuenta con un apartado que permite al usuario ver estadísticas en base a distintos criterios.

RF7: El Sistema permite la calendarización de citas de los pacientes , esta solo lo hace el usuario Administrador Jr.

#### 2.2 REQUISITOS NO FUNCIONALES

RNF1: El sistema soporta y trabaja a su máxima eficiencia, hasta 10 usuarios concurrentes.

RNF2: Las consultas de búsqueda, inserción, etc. se ejecutan en milisegundos ya que la base de datos y las terminales están en la misma LAN.

RNF3: El sistema cuenta con una interfaz gráfica amigable, incluso para usuarios con pocos conocimientos de computación.

RNF4: La seguridad del sistema está basado en las credenciales de inicio de sesión para esto hay validaciones en el lado de la red, así como de la base de datos para hacer las modificaciones.

#### 2.3 REQUISITOS DE INTERFAZ

RI1: Al dar click al botón iniciar sesión el usuario debe entrar al sistema siempre y cuando existan esos datos de inicio de sesión.

RI2: Dentro del sistemas el usuario puede dar click en ‘Nuevo’ así abre un formulario de registro de un nuevo paciente.

RI3: Al dar click en guardar , se guardan los datos del formulario siempre y cuando se haya llenado el campo de nombre y teléfono de no ser así se muestra una alerta avisando.

RI4: Al dar click en buscar se abre un panel con todos los pacientes registrados en la base de datos.

RI5: Al teclear un nombre o teléfono de paciente en la caja de texto para buscar se filtran los resultados en el panel.

RI6: Al clickear un paciente buscado se abre todo su expediente con sus datos cargados.

RI7: Habrá una navegación del expediente del paciente con las pestañas: antecedentes, examen oral, examen radiológico, examen de laboratorio y hoja de evolución.

RI8: Al dar click en la pestaña de ‘antecedentes’, se muestran los antecedentes personales y familiares del paciente.

RI9: Al dar click en la pestaña ‘examen oral’, ‘examen radiológico’ y ‘examen de laboratorio’, se muestran los exámenes que se han realizado por el paciente.

RI10: Al dar click en la pestaña ‘Hoja de Evolución’ se muestra una tabla con los avances del paciente, así como la fecha de cada avance y por quién fue realizado.

RI11: Al dar click en nuevo tratamiento el usuario puede agregar un nuevo avance a la hoja de evolución del paciente.

RI12: Al dar click en el botón Agenda , se muestra un panel nuevo con un calendario marcado con las citas de los pacientes, donde se muestra fecha de la cita, y nombre del paciente.

RI13: En el panel de agenda al dar click en ‘agendar nueva’ cita se registra una nueva cita en el calendario por medio de un formulario.

RI14: En el botón de Configuraciones , se hacen configuraciones tales como : agregar nuevos usuarios, agregar nuevos Cirujanos Dentales (C.D) , entre otras cosas.

#### 2.4 ACTORES Y ROLES DEL SISTEMA

Actor 1 : SUPER-ADMINISTRADOR: Es el administrador del sistema,el usuario con todos los privilegios , el puede agregar , actualizar y eliminar registros, puede calendarizar citas, buscar pacientes, respaldar información, así como también dar de alta nuevos usuarios y asignarles permisos.

Actor 2: ADMINISTRADOR JR.: puede agregar y actualizar registros, buscar pacientes, y puede agendar citas, lo que no puede hacer es eliminar registros ni dar de alta usuarios.

Actor 3: USUARIO: los usuarios alumnos solo pueden buscar pacientes y actualizar registros, el no puede agregar , eliminar , agendar citas, ni dar de alta usuarios.

# 3. CASOS DE USOS

#### 3.1. MODELO DE CASO DE USO.

#### Basic Use Case Diagram - Use Case(2).png

#### 3.2. ESPECIFICACIÓN DE LOS CASOS DE USO.

|  |  |  |
| --- | --- | --- |
| **No. de caso de uso**  01 | **Nombre de caso de Uso**  Ingreso al sistema | |
| **Actores** | Super-Administrador, Administrador Jr. , Usuario | |
| **Objetivo** | La autenticación de los datos para acceso al sistema | |
| **Precondiciones** | 1. el usuario debe estar dado de alta previamente | |
| **Postcondiciones** | * el sistema muestra un menú de inicio | |
| **Flujo de eventos** | **Actividades del Actor** | 1. llena los campos de nombre de usuario y contraseña. 2. Click en el botón iniciar sesión |
| **Actividades del Sistema** | 3. Sistema da acceso, mostrando un menú de inicio |
| **Manejo de situaciones excepcionales** | * Si los datos no son correctos se despliega un mensaje “DATOS INCORRECTOS” y deja al usuario reintentar el inicio de sesión. * Si el servidor de base de datos no esta disponible despliega un mensaje “ERROR CON LA CONEXIÓN DE BASE DE DATOS, VUELVA A INTENTARLO Y SI EL PROBLEMA PERSISTE CONTACTE A SOPORTE” y se permite el reintento de inicio. * Si alguno de los campos no fue proporcionado se despliega un mensaje “POR FAVOR PROPORCIONE LA INFORMACIÓN SOLICITADA” y permite el reintento de inicio. | |

|  |  |  |
| --- | --- | --- |
| **No. de caso de uso**  02 | **Nombre de caso de Uso**  Alta de pacientes | |
| **Actores** | Super-Administrador, Administrador Jr. , Usuario | |
| **Objetivo** | El almacenamiento de toda una serie de datos del paciente. | |
| **Precondiciones** | 1. Se debe haber ingresado al sistema. 2. Paciente no debe existir en la base de datos 3. Se deben proporcionar NOMBRE, TELÉFONO, FECHA DE NACIMIENTO Y DIRECCIÓN de manera obligatoria. | |
| **Postcondiciones** | * Datos del paciente quedan almacenados en la base de datos. | |
| **Flujo de eventos** | **Actividades del Actor** | 1.usuario ingresa al sistema  2.Elegir opción NUEVA.  4.Usuario llena formulario con datos del paciente.  5.elegir opción GUARDAR. |
| **Actividades del Sistema** | 3. Muestra formulario para el registro de datos.  6. Guarda los datos en la base de datos. |
| **Manejo de situaciones excepcionales** | * Si el servidor de base de datos no esta disponible despliega un mensaje “ERROR CON LA CONEXIÓN DE BASE DE DATOS, VUELVA A INTENTARLO Y SI EL PROBLEMA PERSISTE CONTACTE A SOPORTE” y se permite la edición de la información del formulario. * Si alguno de los campos obligatorios no fue proporcionado se despliega un mensaje “campos nombre, domicilio fecha de nacimiento y teléfono obligatorios.” y permite la edición de la información de los formulario. * Si El nombre y Fecha de Nacimiento ya se encuentran en la base de datos no se insertan los datos y despliega un mensaje “EL PACIENTE YA SE ENCUENTRA REGISTRADO”, y permite la edición de la información del usuario. | |

|  |  |  |
| --- | --- | --- |
| **No. de caso de uso**  03 | **Nombre de caso de Uso**  Baja de pacientes | |
| **Actores** | Super-Administrador | |
| **Objetivo** | Eliminar toda información del paciente seleccionado. | |
| **Precondiciones** | 1. Se debe haber ingresado al sistema. 2. Paciente debe existir en la base de datos | |
| **Postcondiciones** | * Datos del paciente ya no existen en la base de datos. | |
| **Flujo de eventos** | **Actividades del Actor** | 1.usuario ingresa al sistema  2.Elegir opción BUSCAR  4.proporciona el nombre del paciente que se desea eliminar  6. de la lista que aparece selecciona al paciente.  8. Elige opción ARCHIVO  9. elige opción ELIMINAR |
| **Actividades del Sistema** | 3. Muestra formulario para la búsqueda.  5.se van filtrando los registros deacuerdo a la informacion proporcionada.  7. Muestra la información del paciente en los formularios de edición.  10. Sistema elimina el registro de información del paciente. |
| **Manejo de situaciones excepcionales** | * Si el servidor de base de datos no está disponible despliega un mensaje “ERROR CON LA CONEXIÓN DE BASE DE DATOS, VUELVA A INTENTARLO Y SI EL PROBLEMA PERSISTE CONTACTE A SOPORTE” y se permite la edición de la información del formulario. | |

|  |  |  |
| --- | --- | --- |
| **No. de caso de uso**  04 | **Nombre de caso de Uso**  Actualización de pacientes | |
| **Actores** | Super-Administrador, Administrador Jr., Usuario | |
| **Objetivo** | Editar y Actualizar la información del paciente seleccionado. | |
| **Precondiciones** | 1. Se debe haber ingresado al sistema. 2. Paciente debe existir en la base de datos | |
| **Postcondiciones** | * Datos del paciente quedan actualizados en la base de datos. | |
| **Flujo de eventos** | **Actividades del Actor** | 1.usuario ingresa al sistema  2.Elegir opción BUSCAR  4.proporciona el nombre del paciente que se desea eliminar  6. de la lista que aparece selecciona al paciente.  8. Edita los campos que se desean actualizar.  9. elige opción GUARDAR |
| **Actividades del Sistema** | 3. Muestra formulario para la búsqueda.  5.se van filtrando los registros deacuerdo a la informacion proporcionada.  7. Muestra la información del paciente en los formularios de edición.  10. Actualiza la información en la base de datos |
| **Manejo de situaciones excepcionales** | * Si el servidor de base de datos no está disponible despliega un mensaje “ERROR CON LA CONEXIÓN DE BASE DE DATOS, VUELVA A INTENTARLO Y SI EL PROBLEMA PERSISTE CONTACTE A SOPORTE” y se permite la edición de la información del formulario * Si se desea editar la fecha de nacimiento y el nombre del paciente desplegará un mensaje “EL NOMBRE Y LA FECHA SON CAMPOS QUE NO PERMITEN LA EDICIÓN” y se vuelve al formulario con los datos del paciente. | |

|  |  |  |
| --- | --- | --- |
| **No. de caso de uso**  05 | **Nombre de caso de Uso**  Consulta de pacientes | |
| **Actores** | Super-Administrador, Administrador Jr., Usuario | |
| **Objetivo** | Buscar y visualizar la información del paciente seleccionado. | |
| **Precondiciones** | 1. Se debe haber ingresado al sistema. 2. Paciente debe existir en la base de datos | |
| **Postcondiciones** | * Datos del paciente quedan en un formulario para la visualización de estos. | |
| **Flujo de eventos** | **Actividades del Actor** | 1.usuario ingresa al sistema  2.Elegir opción BUSCAR  4.proporciona el nombre del paciente.  6. de la lista que aparece selecciona al paciente. |
| **Actividades del Sistema** | 3. Muestra formulario para la búsqueda.  5.se van filtrando los registros deacuerdo a la informacion proporcionad y van apareciendo e una lista.  7. Muestra la información del paciente en los formularios de edición. |
| **Manejo de situaciones excepcionales** | * Si el servidor de base de datos no está disponible despliega un mensaje “ERROR CON LA CONEXIÓN DE BASE DE DATOS, VUELVA A INTENTARLO Y SI EL PROBLEMA PERSISTE CONTACTE A SOPORTE” y se permite la edición de la información del formulario * Si se busca un paciente que no se encuentra registrado la lista de filtro aparecerá con un mensaje “NO SE HAN ENCONTRADO RESULTADOS” | |

|  |  |  |
| --- | --- | --- |
| **No. de caso de uso**  06 | **Nombre de caso de Uso**  Programación de citas | |
| **Actores** | Super-Administrador, Administrador Jr. | |
| **Objetivo** | Calendariza citas a paciente. | |
| **Precondiciones** | 1. Se debe haber ingresado al sistema. 2. Paciente debe existir en la base de datos 3. Dr. debe existir en la base de datos. | |
| **Postcondiciones** | * la cita del paciente es almacenada en la base de datos. | |
| **Flujo de eventos** | **Actividades del Actor** | 1.usuario ingresa al sistema  2.Elegir opción AGENDAR NUEVA  4.llena los campos que se piden.  5. selecciona un doctor.  6. selecciona día y hora.  7. da click en AGENDAR |
| **Actividades del Sistema** | 3. Muestra formulario para la búsqueda.  8. guarda la cita en la base de datos  9. muestra la cita en un calendario. |
| **Manejo de situaciones excepcionales** | * Si el servidor de base de datos no está disponible despliega un mensaje “ERROR CON LA CONEXIÓN DE BASE DE DATOS, VUELVA A INTENTARLO Y SI EL PROBLEMA PERSISTE CONTACTE A SOPORTE” y se permite la edición de la información del formulario * si existe un paciente ya agendado para ese dia y esa hora con el mismo doctor aparece un mensaje “YA EXISTE UNA CITA” JUNTO CON LOS DATOS DE LA CITA QUE YA ESTÁ PROGRAMADA. y se procede a la edición de la nueva cita. * si para ese dia y esa hora ya se tiene el número de espacios límite aparece mensaje “LÍMITE DE PACIENTES ALCANZADOS POR FAVOR ELIJA OTRA HORA/DÍA”. | |

|  |  |  |
| --- | --- | --- |
| **No. de caso de uso**  07 | **Nombre de caso de Uso**  Cierre de sesión | |
| **Actores** | Super-Administrador, Administrador Jr., Usuario | |
| **Objetivo** | Cierra la conexión de la base de datos y sale del sistema | |
| **Precondiciones** | 1. Se debe haber ingresado al sistema. | |
| **Postcondiciones** | * Aparece la ventana de inicio de sesión | |
| **Flujo de eventos** | **Actividades del Actor** | 1.usuario ingresa al sistema  2.Elegir opción CERRAR SESIÓN |
| **Actividades del Sistema** | 3. cierra la conexion a base de datos.  4. manda al usuario a pantalla de inicio de sesión. |
| **Manejo de situaciones excepcionales** | * Si el servidor de base de datos no está disponible despliega un mensaje “ERROR CON LA CONEXIÓN DE BASE DE DATOS, VUELVA A INTENTARLO Y SI EL PROBLEMA PERSISTE CONTACTE A SOPORTE” . | |

# 4. ARQUITECTURA

Se utilizó la arquitectura de capas para la programación de este proyecto basándonos en los siguientes puntos.

**Definición:** Es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

Partiendo de este punto nos fue fácil la organización y la asignación de las tareas entre los diferentes participantes del proyecto, resulta más eficiente dividir el trabajo en los diferentes componentes de la capa y después unir todos los componentes.

Las capas mencionadas en la arquitectura son:

**Capa de presentación:** la que “ve” el usuario , presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso.

En esta capa trabaja el equipo de interfaz, haciendo que el sistema sea amigable con el usuario, responsiva, así como de capturar los datos correctos haciendo validaciones antes de enviarlas por la red al servidor.

**Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio, porque es aquí donde se establecen todas las reglas que deben cumplirse.

Esta capa es la que lleva más trabajo o la que se complica más, ya que depende en gran medida de la persona que solicitó el software y de cómo él quiere que funcione el software así como que haga tareas específicas para la que el software fue diseñado.

**Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos.

En esta capa solo se desarrollan los componentes para la comunicación con las bases de datos y almacenar correctamente los datos enviados a ella.

**Las ventajas que nos ofrece esta arquitectura son:**

**Funcionalidad claramente definida**. El diseño claramente define la separación entre la funcionalidad de cada capa. Capas superiores como la capa de presentación envía comandos a las capas inferiores como la capa de negocios y la capa de datos y los datos fluyen hacia y desde las capas en cualquier sentido.

• **Alta cohesión**. Cada capa contiene funcionalidad directamente relacionadas con la tarea de dicha capa.

• **Reutilizable**. Las capas inferiores no tienen ninguna dependencia con las capas superiores, permitiéndoles ser reutilizables en otros escenarios, es decir podemos usar los mismos componentes de una capa en otros proyectos o en el mismo en diferentes secciones.

• **Desacople**. La comunicación entre las capas está basada en la abstracción lo que provee un desacople entre las capas, si es necesario reemplazar un componente y es más barato volverlo a realizar que modificar el existente.

• **Aislamiento**. El estilo de arquitectura de capas permite aislar los cambios en tecnologías a ciertas capas para reducir el impacto en el sistema total, un cambio en una capa solo afecta los componentes de esta no a todo el sistema.

• **Mejora en pruebas.** Como las capas y sus componentes son independientes se prueban los componentes y cuando hayan pasado dichas pruebas se pueden interconectar con los demás componentes de manera que al conectar un componente no deje de funcionar lo que ya funcionaba.

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 4.1 COMPONENTES DE LA ARQUITECTURA

##### 4.1.1 Capa de clientes:

Super Administrador, Administrador, Usuario.

##### 4.1.2 Capa de presentación:

Se adjuntan capturas de pantalla de la GUI.



Figura 1. Ventana de acceso al sistema.

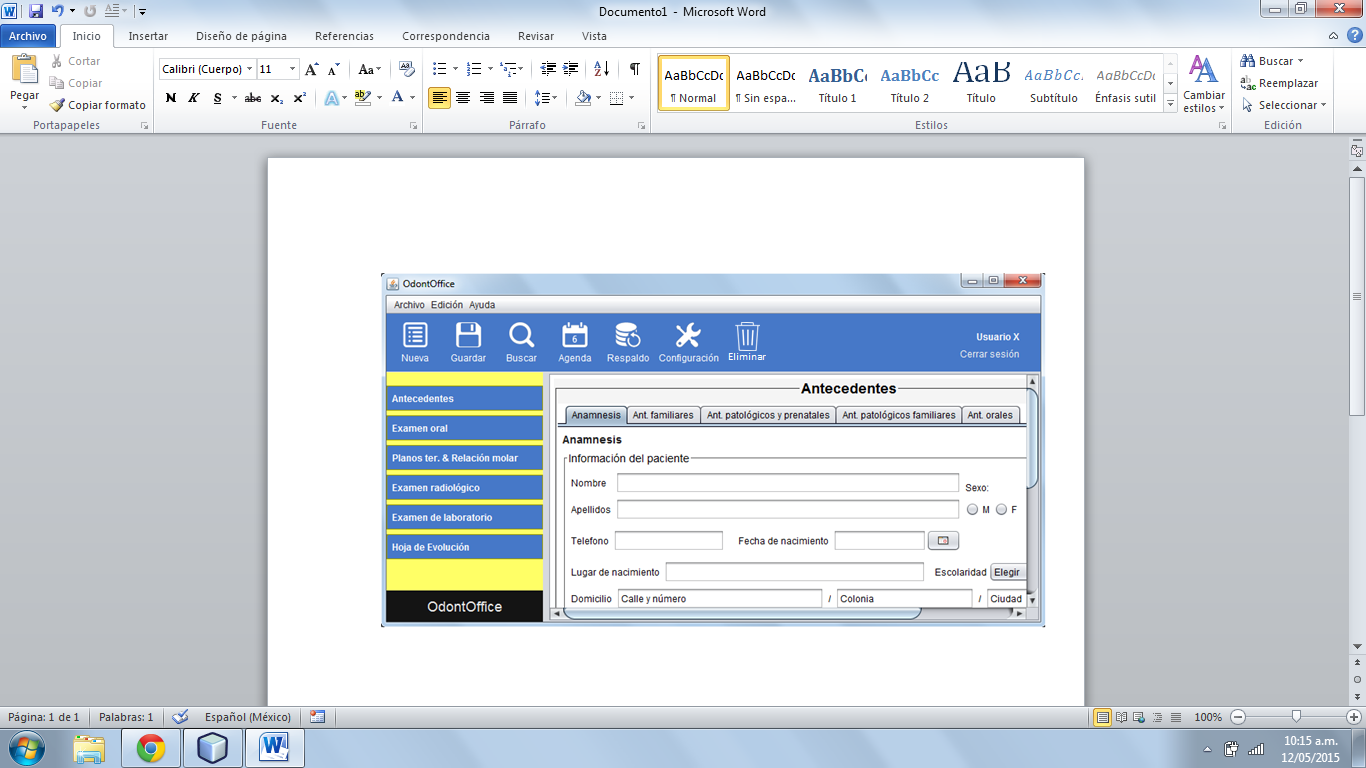
Figura 2.1.Ficha De pacientes Pestaña de datos personales.



Figura 2.2. Pestaña de Odontograma, reporte del estado actual de los dientes del paciente

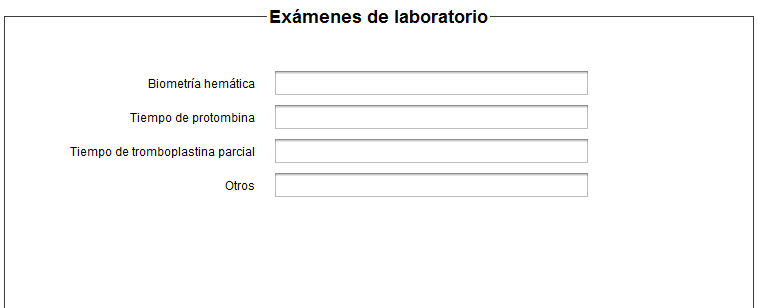


Figura 2.3. Reporte de los exámenes de laboratorio que se le indicó al paciente hacerse.

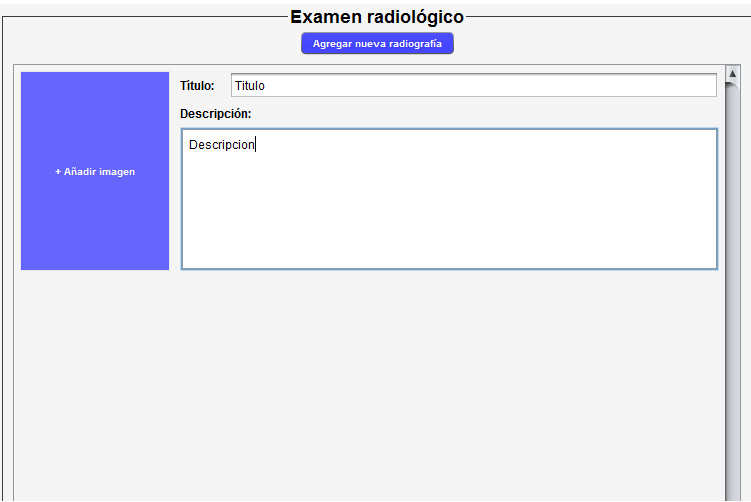


Figura 2.4. Radiografías y descripción del paciente.

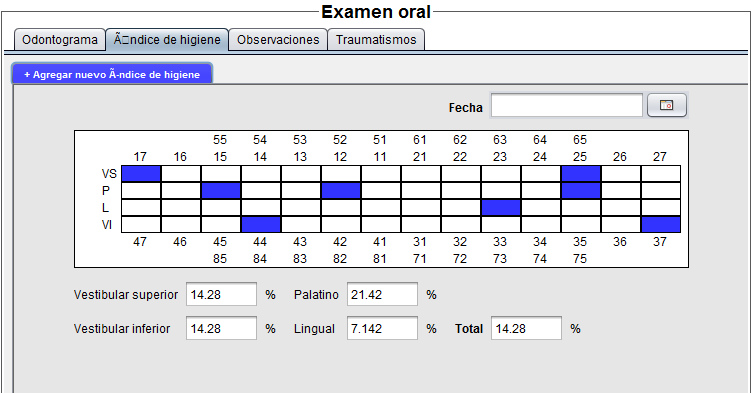


Figura 2.5. Índice de Higiene oral, se automatiza el proceso de cálculo de sarro en diversas áreas de la boca del paciente.

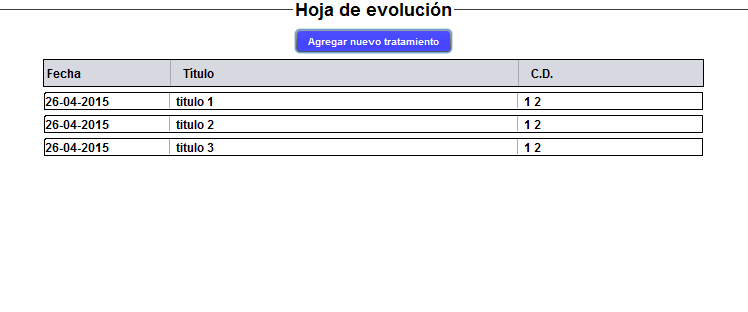
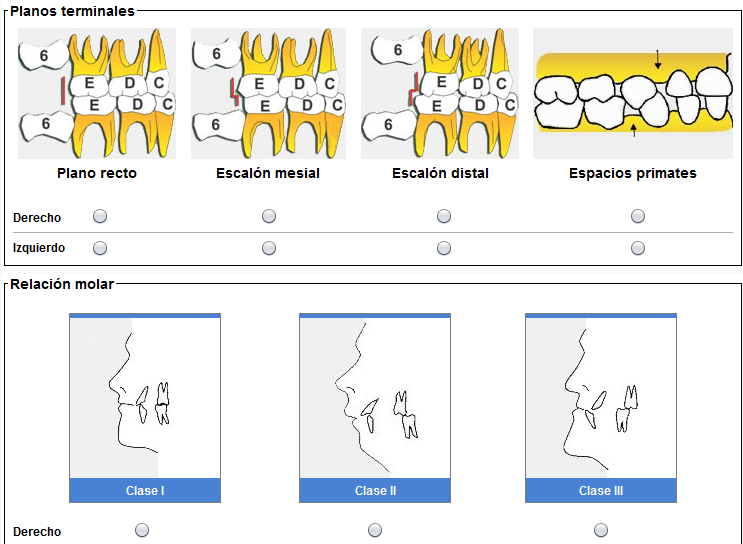


Figura 2.6. Hoja de Evolución, se llevará un reporte de cada actividad realizada al paciente.

  
Figura 2.7. Relaciones Molares del paciente , se selecciona una opción de acuerdo a la información del paciente.

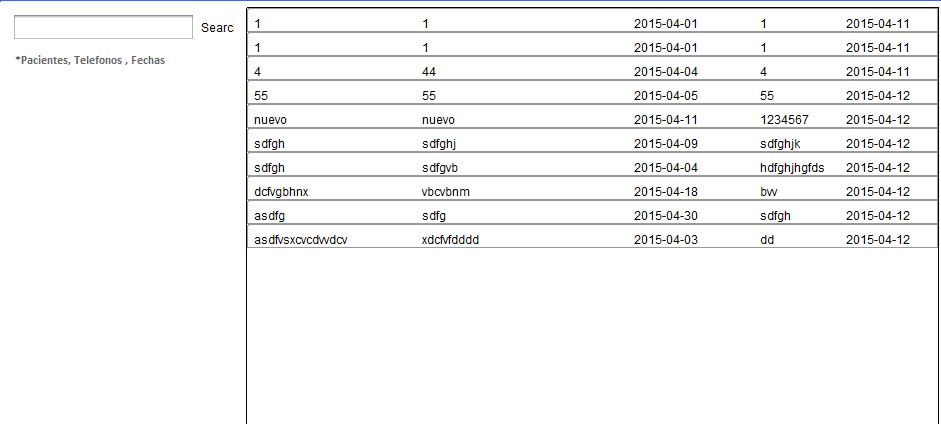
**BUSCAR**

Figura 3. Ventana Buscar,En ella saldrá un listado de pacientes los cuales pueden ser buscados por diversos criterios mismo que irán filtrando la lista.

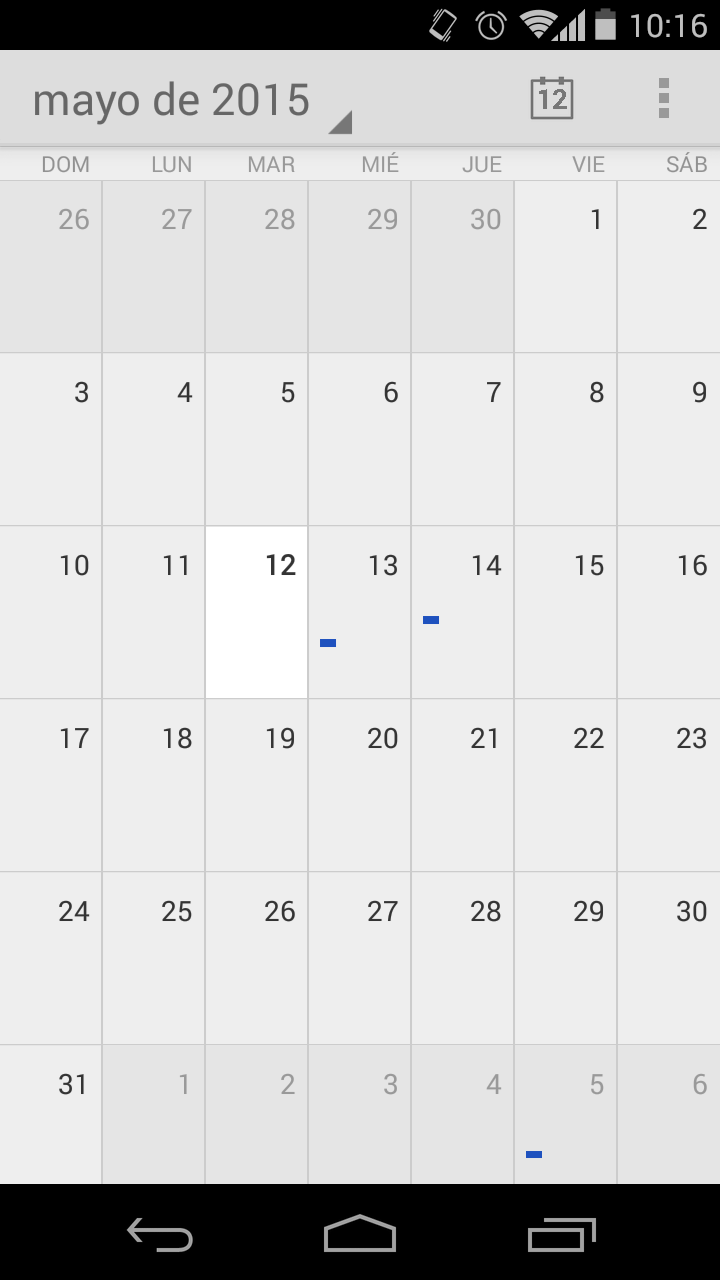
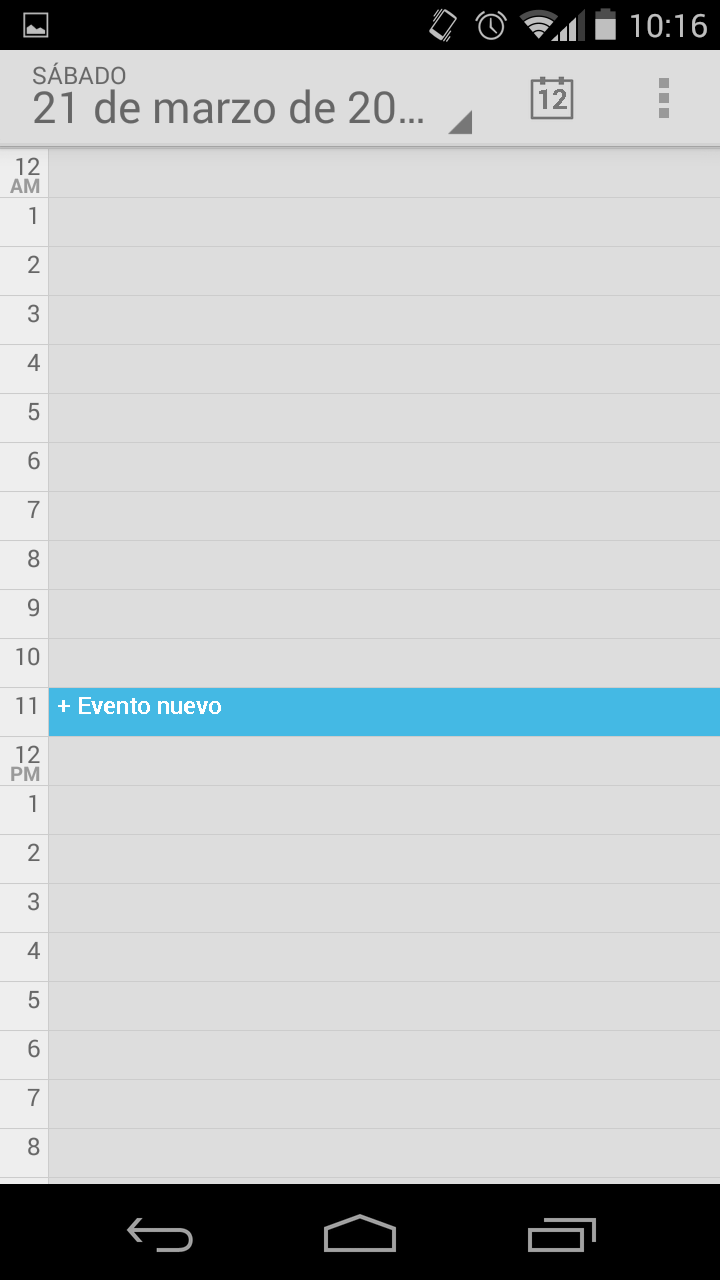


Figura 4.1. Programación de citas, se elige una fecha.

Figura 4.2. Se selecciona una hora para que se abra una pantalla que pida los datos del paciente , el tratamiento a realizar y el Cirujano dentista que lo realizará.

##### 4.1.3 Capa de lógica de negocios:

En la capa de negocio se reciben las peticiones del usuario y se envían las respuestas tras el proceso.

Algunas reglas de nuestra lógica de negocio son :

* Loguearse al sistema , autenticando a un usuario y su contraseña
* Validar si el usuario existe y que sea la contraseña correcta.
* Llenar formularios que se refiere a información del paciente y esta agregarla o actualizarla en la base de datos.
* Validar si el paciente que se quiere registrar , en caso de que no exista lo insertamos como uno nuevo y en caso contrario actualizamos su información.
* Al ir al panel del buscador éste (el panel) debe ser llenado con todos los pacientes registrados ordenados por nombre.
* Al buscar un paciente en la caja de texto , se deben ir filtrando los resultados y posteriormente mostrados en el panel (búsqueda predictiva), el buscador puede identificar : nombres, apellidos y teléfonos.
* En la tabla de índices de higiene al seleccionar casillas se deben ir calculando unos porcentajes y ser mostrados en cada caja de texto según una fórmula preestablecida.
* En la parte de configuración, un usuario podrá agregar nuevos tratamientos a la base de datos y validar si este existe o no.
* En la parte de configuración, se podrán agregar nuevos cirujanos y validar si estos existen o no.
* Al cerrar sesión el sistema muestra nuevamente la ventana de login.

Algunos de los objetos o clases que utilizamos para llevar a cabo el negocio son:

* Clase conexion : la cual nos conecta a la base de datos y ejecuta las operaciones que le enviemos.
* Clase Operaciones: es la clase que recibe los datos de la presentación y ejecuta las consultas necesarias , y envía las respuestas al sistema.
* Objeto Persona
* Objeto Paciente
* Objeto Familiar
* Objeto Examen
* Objeto Actividades
* Objeto Odontograma

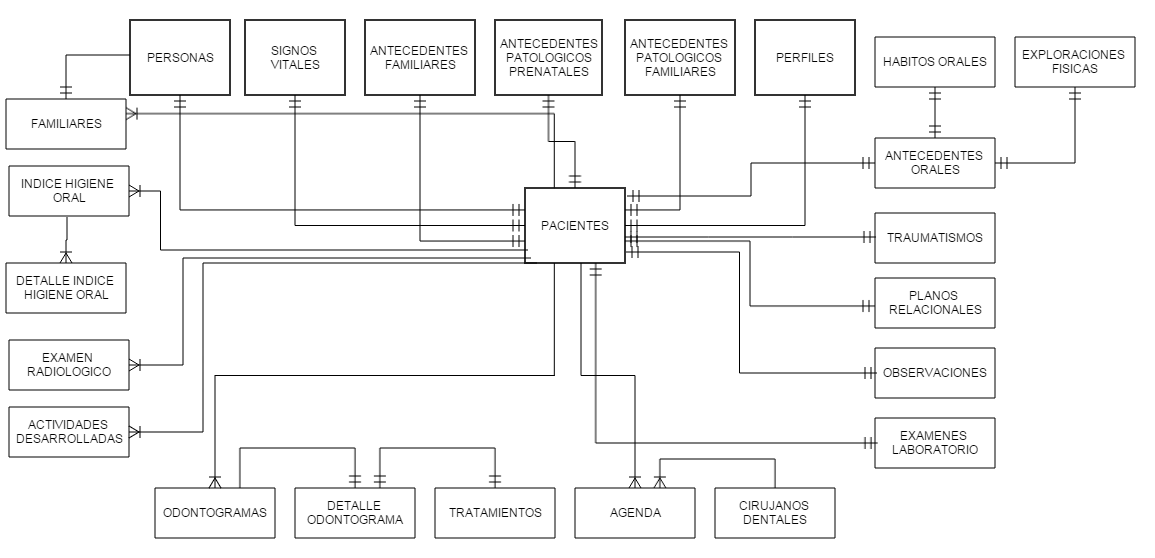
Se tienen diferentes objetos los cuales son: paciente, exámenes radiológicos, familiares, índice de higiene, odontograma, clase conexion, clase operaciones, clase eventos.

El software es capaz de registrar información de los pacientes, cirujanos dentistas y citas para guardarla en la base de datos, de tal forma que en dado de requerirse será posible buscar al paciente y actualizar su información.

##### 

##### 4.1.4 Capa de datos:

A continuación se muestra el diagrama entidad relación de la base de datos.



#### 4.2 RESPONSABLES DE LOS COMPONENTES DE LA ARQUITECTURA

ARQUITECTO DE SOFTWARE:

**Guadalupe de Jesús Guerrero Gutiérrez:** encargado de definir la arquitectura con la que se va a trabajar, así como el análisis de los requisitos y validar la arquitectura.

PRESENTACIÓN:

**Charly Michael Gastelum Quintero**: Diseñador de la GUI, su función será diseñar las ventanas y hacer llegar esos diseños al desarrollador de GUI para su codificación.

**Edder Jose Sanchez Lopez**: Desarrollador GUI, codificará las ventanas de acuerdo a los diseños del diseñador GUI.

LÓGICA DE NEGOCIOS:

**José Ramón Barraza Duarte**: Desarrollador, Encargado de codificar las reglas del negocio de acuerdo a lo definido por el Arquitecto de software.

**Oscar Acosta Mendoza**: Desarrollador, Encargado de codificar las reglas del negocio de acuerdo a lo definido por el Arquitecto de software.

DATOS:

**Rigoberto Reatiga Payan**: Administrador de base de datos, Su función es la construcción de la base de datos y todo lo referente de esta capa.

# 5. FURPS

En este apartado se asignan criterios o métricas de evaluación del software, todo en base a la tabla presentada a continuación

|  |  |  |  |
| --- | --- | --- | --- |
| **SIGLA** | **TIPO REQUERIMIENTO** | | **CRITERIO** |
| **F** | **FUNCTIONAL** | **FUNCIONAL** | * seguridad de los datos * integridad de los datos * características del sistema |
| **U** | **USABILITY** | **FACILIDAD DE USO** | * Pestaña de guía de usuario * Interacción humana sencilla * GUI estética y sencilla |
| **R** | **RELIABILITY** | **FIABILIDAD** | * Frecuencia de fallos * capacidad de recuperación * backups automáticos * Exactitud de los datos de salida |
| **P** | **PERFORMANCE** | **RENDIMIENTO** | * Uso de los recursos * disponibilidad * tiempos de respuesta * tiempo de procesamiento |
| **S** | **SUPPORTABILITY** | **SOPORTE** | * Adaptabilidad * fácil mantenimiento * configuración |
| **+** | **PLUS** | **IMPLEMENTACIÓN** | * acceso a servidor remoto |
| **EMPAQUETAMIENTO** | * forma de distribución |
| **LEGAL** | * licencias |

# 6. VISTAS

En esta sección se incluye una tabla para hacer relación a las imágenes de las pantallas del sistema con los casos de uso.

|  |  |  |  |
| --- | --- | --- | --- |
| # | NOMBRE DEL CASO DE USO | FIGURAS | DESCRIPCIÓN |
| 1 | Ingreso al sistema | 1 | Permite al usuario el ingreso al sistema mediante datos como el user y password. |
| 2 | Alta de pacientes | 2.1  2.2  2.3  2.4  2.5  2.6  2.7 | para dar de alta a un paciente basta con dar clic en el botón de nueva de la barra azul y aparecerá una nueva historia clínica, a un lado del formulario aparecerá un menú de documentos que conforman toda la historia clínica, para guardarlo se da clic en GUARDAR (disket). |
| 3 | Baja de pacientes | 2.1 | El botón con icono de bote de basura elimina la historia clínica que se encuentra desplegada en el momento. |
| 4 | Actualizar pacientes | 3  2.1 | La interfaz de actualizar es la misma que el guardar en este caso se busca un paciente y se modifica algún campo, se da clic guardar (icono disket). |
| 5 | Consulta de pacientes | 3 | Para la consulta de un paciente se da clic en buscar (icono lupa) y aparece el menú de búsqueda. |
| 6 | Programación de citas | 4.1  4.2 | Clic en agenda (icono de calendario) aparece el mes y año actual en caso de requerir otro solo se mueve la fecha, se elige un día y se cambia a una pantalla de horarios para seleccionar un horario para la cita del paciente. |
| 7 | Cerrar sesión | 2.1  1 | Clic en cerrar sesion y vuelve a la pantalla de ingreso al sistema |