



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB DE
GESTIÓN ODONTOLÓGICO PARA LA EMPRESA ELÉCTRICA
RIOBAMBA S.A.**

Trabajo de titulación presentado para optar al grado académico de:
INGENIERO EN SISTEMAS INFORMÁTICOS

AUTOR: JESÚS MESÍAS PONLUIZA HORTA

TUTOR: DR. JULIO ROBERTO SANTILLÁN CASTILLO

Riobamba-Ecuador

2016

©2016, Jesús Mesías Ponluiza Horta

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELÉCTRICA
ESCUELA DE INGENIERÍA EN SISTEMAS

El Tribunal del Trabajo de Titulación certifica que: El trabajo de investigación: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB DE GESTIÓN ODONTOLÓGICO PARA LA EMPRESA ELÉCTRICA RIOBAMBA S.A., de responsabilidad del señor Jesús Mesías Ponluiza Horta, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

Dr. Miguel Tasambay S. Ph. D
DECANO DE LA FACULTAD DE
INFORMÁTICA Y ELECTRÓNICA

Dr. Julio Santillán Castillo
DIRECTOR DE LA ESCUELA DE
INGENIRÍA EN SISTEMAS

Dr. Julio Santillán Castillo
DIRECTOR DEL TRABAJO
DE TITULACIÓN

Ing. Diego Reina
MIEMBRO DEL TRIBUNAL

Yo, Jesús Mesías Ponluiza Horta soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual de la misma pertenece a la Escuela Superior Politécnica de Chimborazo.

Jesús Mesías Ponluiza Horta

DEDICATORIA

A la memoria de mi madre.

Jesús

AGRADECIMIENTO

Agradezco eternamente a Dios por sus bendiciones y por permitirme cumplir una meta más en mi vida. A mi padre Gustavo por todo el apoyo que me ha brindado en mi vida, por su esfuerzo por comprenderme y siempre estar pendiente gracias. A mis herman@s por sus consejos su apoyo gracias a todos ustedes, gracias por todo. A Mayra por ser una persona especial en mi vida, por su paciencia y comprensión gracias de todo corazón. A mi director de proyecto de grado Dr. Julio Santillán que con su conocimiento y experiencia impartida fueron una guía para el desarrollo de este proyecto gracias. Al Ing. Diego Reina por su colaboración en este proyecto de grado. A la Od. Rosa Bonifaz y a la asistente Geoconda Iglesias gracias por todo el apoyo que me han brindado para realizar este proyecto gracias. A mis amigos y compañeros de la poli por su amistad, sus consejos, por su ayuda que de alguna manera contribuyeron con este esfuerzo. Gracias Totales

Jesús

ÍNDICE DE ABREVIATURAS Y ACRÓNIMOS

| | |
|---------------|--|
| ESPOCH | Escuela Superior Politécnica de Chimborazo |
| EERSA | Empresa Eléctrica Riobamba S.A. |
| DAO | Objeto de Acceso a Datos |
| IDE | Integrated Development Environment |
| JAVA | Lenguaje de programación orientado a objetos |
| JDK | Java Development Kit |
| HQL | Lenguaje de Consultas de Hibernate |
| SQL | Structured Query Language |
| JSF | Java Server Faces |
| EJB | Enterprise Java Bean |
| J2EE | Java Enterprise Edition |
| AJAX | Asynchronous Javascript and XML |
| ORM | Mapeo Objeto Relacional |
| PAO | Programación Orientada Aspectos |
| POJO | Plain Object Java Old |
| MVC | Modelo Vista Controlador |
| BD | Base de Datos |
| DI | Inyección de Dependencia |
| IoC | Inversión de Control |
| CRUD | Create, Read, Update, Delete |
| XML | Extensible Markup Language |
| HTML | Hipertext Markup Language |
| XHTML | Extensible Hipertext Markup Language |
| PK | Primary Key |
| FK | Foreign key |
| UI | Interfaz de Usuario |
| RQ | Requerimientos |
| HU | Historia de Usuario |
| HUT | Historia de Usuario Técnica |
| TIC'S | Tecnologías de la Información y Comunicación |

ÍNDICE DE CONTENIDO

| | |
|--------------------------|------------|
| ÍNDICE DE TABLAS | xi |
| ÍNDICE DE FIGURAS | xii |
| ÍNDICE DE ANEXOS | xiv |
| RESUMEN | xv |
| SUMMARY | xvi |
| INTRODUCCIÓN | 17 |

CAPÍTULO I

| | | |
|------------|---|-----------|
| 1 | MARCO TEÓRICO | 24 |
| 1.1 | Aplicaciones Web | 24 |
| 1.1.1 | Características | 24 |
| 1.1.2 | Ventajas | 24 |
| 1.1.3 | Desventajas | 25 |
| 1.1.4 | Arquitectura en capas | 25 |
| 1.2 | Patrones de Diseño | 26 |
| 1.2.1 | Inyección de Dependencia | 26 |
| 1.2.2 | Ventajas | 27 |
| 1.2.3 | Desventajas | 28 |
| 1.3 | Framework | 28 |
| 1.3.1 | Tipos de Framework | 28 |
| 1.3.2 | Framework web para Java | 28 |
| 1.3.3 | Ventajas | 29 |
| 1.3.4 | Desventajas | 30 |
| 1.4 | Por qué Spring, Hibernate y Primefaces | 30 |
| 1.5 | Spring | 30 |
| 1.5.1 | Introducción | 30 |
| 1.5.2 | Definición | 31 |
| 1.5.3 | Características | 31 |
| 1.5.4 | Ventajas | 31 |
| 1.5.5 | Desventajas | 32 |
| 1.5.6 | Inversión de Control | 32 |
| 1.5.7 | Inyección de Dependencia con Spring | 32 |
| 1.5.8 | Spring Bean | 35 |
| 1.5.9 | Módulos de Spring | 36 |
| 1.5.10 | Configuración de la Metadata de Spring | 38 |
| 1.5.11 | Spring-IoC Container | 39 |
| 1.5.12 | BeanFactory | 39 |
| 1.5.13 | ApplicationContext | 40 |
| 1.5.14 | Spring ORM | 40 |
| 1.5.15 | Spring MVC | 41 |

| | | |
|------------|---|-----------|
| 1.6 | Hibernate | 42 |
| 1.6.1 | Características | 42 |
| 1.6.2 | Base de Datos soportadas | 42 |
| 1.6.3 | Ventajas | 42 |
| 1.6.4 | Desventajas | 43 |
| 1.6.5 | Arquitectura de Hibernate | 43 |
| 1.6.6 | HQL | 44 |
| 1.6.7 | Configuración | 44 |
| 1.7 | Java Server Faces | 45 |
| 1.7.1 | Características | 45 |
| 1.7.2 | Primefaces | 46 |
| 1.8 | Herramientas de apoyo para el desarrollo | 46 |
| 1.8.1 | MySQL | 46 |
| 1.8.2 | Netbeans | 47 |
| 1.8.3 | GlassFish | 48 |
| 1.9 | Metodología Scrum | 48 |
| 1.9.1 | Porque Scrum | 49 |
| 1.9.2 | Visión general del proceso | 49 |
| 1.9.3 | Beneficios de la Metodología | 49 |
| 1.9.4 | Fases de Scrum | 50 |
| 1.9.5 | Componentes de Scrum | 50 |
| 1.9.6 | Artefactos | 51 |
| 1.9.7 | Eventos | 51 |

CAPÍTULO II

| | | |
|------------|--|-----------|
| 2 | DESARROLLO DEL SISTEMA WEB DE GESTIÓN ODONTOLÓGICO UTILIZANDO SCRUM | 53 |
| 2.1 | Fase de Pre-Juego | 53 |
| 2.1.1 | Antecedentes | 53 |
| 2.1.2 | Alternativa de solución | 53 |
| 2.1.3 | Beneficios de la aplicación | 54 |
| 2.1.4 | Factibilidad | 54 |
| 2.1.5 | Análisis de riesgos | 55 |
| 2.1.6 | Características de la aplicación | 55 |
| 2.1.7 | Herramientas para el desarrollo | 56 |
| 2.1.8 | Roles | 57 |
| 2.1.9 | Product backlog | 57 |
| 2.1.10 | Planificación de Sprint | 59 |
| 2.2 | Fase de Juego (Desarrollo) | 60 |
| 2.2.1 | Sprint 0 | 61 |
| 2.2.2 | Sprint 1 | 79 |
| 2.3 | Fase de post-juego | 80 |

CAPÍTULO III

| | | |
|----------|---|-----------|
| 3 | ANÁLISIS Y DISCUSIÓN DE RESULTADOS | 82 |
|----------|---|-----------|

| | | |
|------------------------|---|-----------|
| 3.1 | Definición de los parámetros | 82 |
| 3.2 | Criterios de evaluación | 83 |
| 3.3 | Tiempo de respuesta | 84 |
| 3.3.1 | Análisis actual del proceso de gestión de información en el departamento odontológico | 84 |
| 3.3.2 | Tiempo de respuesta con el sistema web de gestión odontológico | 85 |
| 3.3.3 | Comparación de resultados | 86 |
| 3.4 | Capacidad de Análisis | 89 |
| 3.5 | Capacidad de Modificación | 90 |
| 3.5.1 | Interpretación de Resultados | 92 |
| 3.6 | Usabilidad | 92 |
| CONCLUSIONES | | 95 |
| RECOMENDACIONES | | 96 |
| BIBLIOGRAFÍA | | |
| ANEXOS | | |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1-1: Componentes de Spring Framework | 37 |
| Tabla 2-1: Roles de Scrum | 50 |
| Tabla 1-2: Usuarios Directos | 55 |
| Tabla 2-2: Herramientas para el desarrollo | 56 |
| Tabla 3-2: Roles para el desarrollo del proyecto | 57 |
| Tabla 4-2: Product Backlog | 58 |
| Tabla 5-2: Programación del Sprint Backlog | 59 |
| Tabla 6-2: Historia Técnica 1 del sprint 0 | 61 |
| Tabla 7-2: Tarea de Ingeniería 1 de la historia técnica 1 | 61 |
| Tabla 8-2: Historia Técnica 2 del sprint 0 | 63 |
| Tabla 9-2: Historia Técnica 3 del sprint 0 | 65 |
| Tabla 10-2: Diccionario de datos tabla paciente | 67 |
| Tabla 11-2: Historia Técnica 5 del sprint 0 | 67 |
| Tabla 12-2: Historia Técnica 4 del sprint 0 | 69 |
| Tabla 13-2: Historia Técnica 6 del sprint 0 | 77 |
| Tabla 14-2: Historia de Usuario 1 del sprint 1 | 79 |
| Tabla 1-3: Parámetro e Indicador | 83 |
| Tabla 2-3: Criterios de evaluación | 83 |
| Tabla 3-3: Tiempos | 84 |
| Tabla 4-3: Tiempo de respuesta proceso manual | 85 |
| Tabla 5-3: Tiempo de respuesta proceso automatizado | 85 |
| Tabla 6-4: Comparación de resultados | 86 |
| Tabla 7-3: Resultados Generales | 87 |
| Tabla 8-3: Resultados de Tiempos de respuesta | 89 |
| Tabla 9-3: Escala de valoración índice de mantenibilidad | 91 |
| Tabla 10-3: Test de Evaluación | 93 |

ÍNDICE DE FIGURAS

| | | |
|--------------|---|----|
| Figura 1-1: | Arquitectura en capas | 25 |
| Figura 2-1: | Acoplamiento Fuerte | 27 |
| Figura 3-1: | Débilmente Acoplado | 27 |
| Figura 4-1: | Librerías de Spring Framework | 36 |
| Figura 5-1: | Módulos de Spring Framework | 36 |
| Figura 6-1: | Spring Core | 38 |
| Figura 7-1: | Contenedor de Spring IoC | 39 |
| Figura 8-1: | Modelo Vista Controlador | 41 |
| Figura 9-1: | Arquitectura de Hibernate | 43 |
| Figura 10-1: | Netbeans | 48 |
| Figura 11-1: | GlassFish | 48 |
| Figura 12-1: | Scrum | 49 |
| Figura 1-2: | Pantalla Configuración de Frameworks | 57 |
| Figura 2-2: | Duración de Sprints | 60 |
| Figura 3-2: | Arquitectura de la Aplicación | 63 |
| Figura 4-2: | Diagrama de Despliegue | 64 |
| Figura 5-2: | Caso de uso rol Administrador | 64 |
| Figura 6-2: | Caso de uso rol Dentista | 65 |
| Figura 7-2: | Caso de uso rol Paciente | 65 |
| Figura 8-2: | Base de Datos | 66 |
| Figura 9-2: | Estructura de paquetes de la aplicación | 68 |
| Figura 10-2: | Creación de la aplicación Java web | 69 |
| Figura 11-2: | Selección de Frameworks | 70 |
| Figura 12-2: | Verificación de Jar Agregados | 70 |
| Figura 13-2: | Paquete Java Hibernate | 70 |
| Figura 14-2: | Archivo hibernate.cfg.xml | 71 |
| Figura 15-2: | Archivo hibernate.reveng.xml | 71 |
| Figura 16-2: | Archivo generación POJOS | 71 |
| Figura 17-2: | Archivo de configuración POJOS | 72 |
| Figura 18-2: | Archivos generados | 72 |
| Figura 19-2: | Archivo Citamedica.hbm.xml | 73 |
| Figura 20-2: | Clase Citamedica.java | 73 |
| Figura 21-2: | Interfaz Citamedica | 74 |
| Figura 22-2: | Implementación de la interfaz Citamedica | 74 |
| Figura 23-2: | Implementación de la clase CitamedicaimplBo | 75 |
| Figura 24-2: | Lógica del negocio Citamedica | 75 |

| | |
|---|----|
| Figura 25-2: Lógica del negocio Citamedica | 76 |
| Figura 26-2: Lógica del negocio 2 Citamedica | 76 |
| Figura 27-2: Integración de spring y Primefaces | 77 |
| Figura 28-2: Configuración archivo web.xml | 77 |
| Figura 29-2: Autenticación de Usuarios | 78 |
| Figura 30-2: Pantalla principal del sistema web | 79 |
| Figura 31-2: Pantalla ingresar paciente | 80 |
| Figura 32-2: Pantalla crud paciente | 80 |
| Figura 33-2: Gráfico Burndown del proyecto | 81 |
| Figura 1-3: Tiempo de Respuesta | 86 |
| Figura 2-3: Determinación de la Hipótesis | 88 |
| Figura 3-3: Tiempo para las tareas de mantenimiento | 90 |
| Figura 4-3: Tiempo para las tareas de mantenimiento | 91 |
| Figura 5-3: Análisis paquete Bo | 91 |
| Figura 6-3: Detalle del análisis paquete Bo | 92 |
| Figura 7-3: Validación de la Usabilidad | 93 |

ÍNDICE DE ANEXOS

| | |
|-----------------|------------------------------|
| ANEXO A: | ESTUDIO DE FACTIBILIDAD |
| ANEXO B: | ANÁLISIS DE RIESGOS |
| ANEXO C: | DICCIONARIO DE DATOS |
| ANEXO D: | SPRINT Y TAREAS DE INGENIRÍA |
| ANEXO E: | ENTREVISTA |
| ANEXO F: | CAPTURAS CON JHAWK |
| ANEXO G: | DOCUMENTOS |
| ANEXO H: | MANUAL DE USUARIO |

RESUMEN

Se implementó el Sistema Web de Gestión Odontológico en el Departamento de Odontología de la Empresa Eléctrica Riobamba S.A. de la ciudad de Riobamba, el cual se automatizó todos los procesos de gestión de información de los pacientes los mismos que se los realizaba de forma manual en la dependencia, con el único fin de modernizar y facilitar el manejo de esta información. Para el desarrollo de esta investigación se utilizó el método inductivo con el fin de analizar e interpretar los resultados obtenidos para los indicadores tiempo de respuesta, mantenibilidad, usabilidad y posteriormente cumplir con los objetivos planteados en esta investigación, además se emplearon varias técnicas como la observación, entrevistas, revisión de documentos y encuesta para obtener los datos relacionados con los procesos de gestión de información antes de la implementación del sistema web en el Departamento Odontológico, Se desarrolló el sistema web empleando la metodología ágil Scrum dando cumplimiento con todos los requisitos expuestos por la Odontologa encargada del departamento. El desarrollo de la aplicación se realizó en la plataforma Java, lenguaje de programación Java utilizando los Frameworks, Spring 4.0.1 para gestionar los Beans, Hibernate 4.3 para persistir la base de datos, Primefaces 5.3 para la creación de interfaces enriquecidas, base de datos MySQL 5.5.8, IDE de desarrollo Netbeans 8.1 y el servidor web GlassFish 4.1. En base al análisis de los indicadores tales como tiempo de respuesta, mantenibilidad y usabilidad se obtuvieron los siguientes resultados: el tiempo usado con la gestión manual es de 56 minutos, el Sistema Web lo redujo a 10 minutos, disminuyendo en un 82.14% el tiempo empleado en los procesos de gestión de información de los pacientes, posteriormente se analizó la aplicación con la herramienta JHAWK obteniendo que el índice de mantenibilidad es excelente lo cual el sistema web es fácil de actualizar y mantener. Se concluye que la aplicación web es aceptable en un 94.78% en la población analizada. La implementación del sistema web de gestión Odontológico tiene como fin facilitar la gestión de información de los pacientes en el departamento, se recomienda a los empleados de la Empresa Eléctrica Robamba utilizar la aplicación web y de igual manera la actualización por parte del personal encargado del mantenimiento del software.

Palabras clave:

<SISTEMA WEB DE GESTIÓN ODONTOLÓGICO> <(SPRING) FRAMEWORK>
<FRAMEWORK (HIBERNATE)> <FRAMEWORK (PRIMEFACES)> <LENGUAJE JAVA>
<METODOLOGÍA SCRUM> <BASE DE DATOS (MYSQL)> <SERVIDOR (GLASSFISH)>
<EMPRESA ELÉCTRICA RIOBAMBA S.A.> <RIOBAMBA (CANTÓN)> <SISTEMAS INFORMÁTICOS>

SUMMARY

The Web System of Dental Management at the Dental Department of Riobamba Power Enterprise S.A in Riobamba city was implemented, which automatized all the processes of management and information from the patients which were carried on manually at the department, the only aim was to modernize and ease the use of this information. The inductive method was used in this research addressed to analyze and understand the results obtained for the indicators response time, maintenance, usage and eventually achieving the proposed goals for this research work, it was also used several techniques such as observation, interviews, documents review and surveys to obtain the data related to the process of information management before the implementation of the web system at the Dental Department, the system was developed by using the Scrum methodology fulfilling all the requirements of the dentist in charge of the department. The development of the app was done with Java platform, Java programming language by using Frameworks, Spring 4.0.1 to manage Beans, Hibernate 4.3 to persist the data base, Primefaces 5.3 for the creation of enriched interfaces, data base MySQL 5.5.8, IDE of Netbeans 8.1 development, and GlassFish 4.1 web server. Basad on the analysis of the indicators such as response time, maintenance and usage, the following results were obtained: The time used with the manual method in 56 minutes, the Web System reduced it to 10 minutes, reducing in 82.14% the time used for the management processes of information of the patients, later on, the app with the JHAWK tool was analyzed having that the rate of maintenance is excellent since the web system is easy to update and maintain. It is concluded that the web app is acceptable in 94.78% in the analyzed population. The implementation of the web system of Dental Management aims to ease the use of the information of patients at the department, it is recommended to the employees of Riobamba Power Enterprise to use the web application as well as the training to the staff in charge of the software maintenance.

Key words:

<WEB SYSTEM OF DENTAL MANAGEMENT><(SPRING) FRAMEWORK>
<FRAMEWORK (HIBERNATE)> <FRAMEWORK (PRIMEFACES)> <JAVA LANGUAGE>
<SCRUM METHODOLOGY><(MYSQL) DATA BASE> <(GLASSFISH) SERVER>
<RIOBAMBA POWER ENTERPRISE S.A><RIOBAMBA (CITY)><COMPUTER PROGRAMS>

INTRODUCCIÓN

Las Aplicaciones Web se han convertido en herramientas útiles tanto para las entidades públicas como privadas, ya que son de gran ayuda en la realización de ciertas tareas para las cuales están programadas, todas con requerimientos diferentes pero con propósitos en comunes como ahorrar tiempo y simplificar el trabajo en fin mejorar la gestión de información en distintas áreas de dichas instituciones.

Los desarrolladores de software tienen a su elección una variedad de tecnologías al momento de diseñar y desarrollar aplicaciones web como lenguajes de programación, patrones de diseño, Frameworks, IDEs, en distintas plataformas. Sin embargo estas aplicaciones web desarrolladas deben ser susceptibles ante los cambios que podrían presentarse en el transcurso del tiempo, he ahí la importancia al momento de elegir dichas tecnologías para crear aplicaciones flexibles.

El presente trabajo de titulación tiene como objetivo el desarrollo de una Aplicación Web de gestión odontológica para el departamento de odontología de la Empresa Eléctrica Riobamba S.A. basado en herramientas de Software Libre.

La aplicación web está desarrollada bajo el lenguaje de programación JAVA, los frameworks Spring, Java Server Faces con Primefaces, Hibernate como Frameworks de Desarrollo y MySQL como motor de Base de Datos, utilizando el Entorno de Desarrollo Integrado (IDE) Netbeans y el servidor de aplicaciones GlassFish con el fin de obtener una aplicación robusta haciendo uso de buenas prácticas de programación. Para el desarrollo de la aplicación se utilizará la metodología ágil Scrum, la misma que se basa en obtener resultados pronto ante requerimientos cambiantes. La estructura del trabajo de titulación está comprendido de cuatro capítulos que son.

CAPÍTULO I, Marco Referencial, en este capítulo se describe los antecedentes, la justificación del trabajo de titulación, y los objetivos a alcanzar, es decir responde a ciertas inquietudes: Qué, Con qué y Por qué se va hacer el presente trabajo de titulación.

CAPÍTULO II, Marco Teórico, en este capítulo se describen conceptos fundamentales de las distintas herramientas como: Spring, Hibernante, Primefaces, Scrum etcétera que son utilizados en el desarrollo de la Aplicación Web.

CAPITULO III, Desarrollo e implementación del sistema web de gestión odontológico, aquí se detalla el desarrollo del sistema web de gestión odontológico, bajo los distintos elementos descritos en el capítulo II, y utilizando la metodología SCRUM.

CAPITULO IV, Marco de resultados y discusión de los resultados, se detalla la eficiencia del software desarrollado en base a parámetros establecidos.

En este apartado se detallará los antecedentes, la justificación teórica como aplicada, los objetivos planteados para el trabajo de grado así como también los métodos y técnicas a utilizar.

Antecedentes

En la actualidad el progreso de las aplicaciones web, han tenido un enfoque significativo en el mundo, debido a que la mayoría de las personas necesitan acceder a cierta información de su interés desde diferentes lugares. La web ha evolucionado conjuntamente con la tecnología, permitiendo a los desarrolladores la construcción de aplicaciones web mediante el uso de diferentes tecnologías como Frameworks.

Los Frameworks permiten el desarrollo rápido de aplicaciones, ya que disponen de ciertas funcionalidades desarrolladas y testeadas así como librerías, componentes y clases, en diferentes tipos de lenguajes de programación que facilitan el desarrollo ágil de las aplicaciones, también ayudan a disminuir el tiempo en tareas repetidas haciendo que los desarrolladores de software tengan una estructura para empezar a diseñar y desarrollar las aplicaciones dedicando más tiempo al desarrollo de requerimientos específicos del sistema.

Para la elaboración de las aplicaciones web se puede utilizar distintos frameworks con funcionalidades específicas en cada capa por ejemplo para la capa de presentación (vista) se tiene Struts, JSF (Java Server Faces), Spring MVC, Vaadin, Grails, etcétera para la capa lógica del negocio se tiene Spring, para la capa acceso a datos o persistencia de datos se tiene Hibernate, JPA, MyBatis, etcétera, ofreciendo una serie de ventajas para los desarrolladores.

La Empresa Eléctrica Riobamba S.A. suministra el servicio de energía eléctrica, con transparencia y efectividad, gracias a la capacidad, responsabilidad y compromiso de su talento humano con la ciudadanía. La EERSA cuya estructura organizacional se compone de varios departamentos uno de estos es el departamento médico odontológico encargado de brindar servicio de salud dental a todos sus empleados, dicho departamento no cuenta con un sistema informático especializado,

por lo cual todo el proceso se realiza de forma manual y semiautomática como el llenado de los datos de los pacientes en formularios pre impresos, la emisión de reportes en documentos de Word, Excel etcétera, lo que dificulta al momento de gestionar los datos de los pacientes, emitir reportes a las personas encargadas del control del departamento y en si la gestion de información.

En base a los problemas suscritos se ha planteado la elaboración de un sistema web de Gestión Odontológico para el Departamento Odontológico de la Empresa Eléctrica Riobamba S.A. que resuelva los inconvenientes presentados en el mismo, para la construcción del sistema web se aplicará una metodología ágil y para el desarrollo se empleará Frameworks Open Source.

Formulación Del Problema

¿Falta de un Sistema Automatizado con soporte web para el departamento odontológico de la EERSA?

Sistematización Del Problema

¿Qué beneficios se obtienen al realizar el sistema web de gestión Odontológico en la Empresa Eléctrica Riobamba S.A.?

¿Qué tecnologías se puede usar para el desarrollo de la aplicación web?

¿Qué funcionalidades ofrece el sistema web?

Justificación Del Trabajo De Grado

A continuación se describe la justificación Teórica y Aplicativa del trabajo de titulación señalando las razones, importancia, ventajas que conllevará al desarrollo de esta investigación.

Justificación Teórica

Al apoyarse de algún Framework para la construcción de aplicaciones web hace que los desarrolladores de software no dediquen el tiempo en realizar procesos repetitivos, y se enfoquen a las partes más importantes de las aplicaciones que estén desarrollando, permitiendo un desarrollo ágil de software ya que los frameworks implementan librerías ya testeadas, desarrollados con patrones de diseño como Modelo, Vista, Controlador (MVC), que facilitan el entorno de trabajo haciendo que las aplicaciones sean flexibles ante los cambios que se puedan presentar en el transcurso del tiempo. Existen una variedad de frameworks en el mercado,

gratuitas y de pago, algunos con características distintas como distintos lenguajes de programación, frameworks con funciones específicas. La elección de algún marco de trabajo dependerá de los requerimientos del sistema a desarrollarse así como también de la experiencia del desarrollador de software. El lenguaje de programación Java es uno de los más populares según la página TIOBE lo que impulsó a seleccionar esta plataforma.

Spring es un Framework para el desarrollo de aplicaciones de software el cual ayuda que el código escrito sea limpio, elegante y reutilizable ya que la filosofía de este marco de trabajo es promover el uso de buenas prácticas de programación, para que la aplicación desarrollada sea altamente modular y sus componentes estén débilmente acoplados, obteniendo una aplicación flexible. Spring tiene dos características interesantes como la Inyección de Dependencias y la Programación Orientada a Aspectos, además Spring se integra fácilmente con otras tecnologías y librerías por lo que no es necesario utilizar todos los módulos que nos proporciona ya que spring se adapta a la aplicación que se está desarrollando más no la aplicación tiene que adaptarse al framework. Por tal motivo se seleccionó este framework para el desarrollo de la aplicación.

Primefaces es un framework que tiene una gran cantidad de componentes enriquecidos que facilitan la construcción de interfaces de usuario (vistas), dichos componentes tienen soporte Ajax el cual mejoran la interactividad con el usuario, usa jQuery para los efectos visuales, además dispone de varios temas para cambiar la apariencia obteniendo un estándar de diseño y haciendo que la aplicación web sea amigable para el usuario del sistema.

Hibernate es un framework de persistencia de datos que nos facilita el mapeo de los atributos de la base de datos relacional configurados en archivos XML, además dispone de su propio lenguaje de consultas Hibernate Query Language (HQL).

MySQL es un sistema de gestión de base de datos (SGBD) relacionales flexible y proporciona una interfaz amigable.

La metodología ágil Scrum es ideal para el desarrollo de proyectos de software, ya que por sus componentes, artefactos, roles y valores nos garantiza generar software de calidad.

Por tal razón se ha optado en realizar el sistema web de gestión odontológico bajo la plataforma Java y con el apoyo de los Frameworks Open Source como: Spring, Primefaces, Hibernate y el gestor de base de datos MySQL, con el fin de automatizar la información que se llevan a cabo en el departamento Odontológico. Esta aplicación se desarrollará utilizando de la metodología ágil Scrum.

Justificación Aplicativa

El Departamento Médico Odontológico de la Empresa Eléctrica Riobamba S.A. no cuenta con un sistema automatizado, por lo tanto los procesos como citas médicas, diagnósticos, tratamientos, gestión de pacientes, se los realiza de forma manual, provocando inconsistencia de información, uso excesivo de papel, así como también información aislada y redundante, no existe un estándar para el registro de información de los pacientes como: datos personales, diagnóstico, tratamiento, control de citas.

Por ejemplo el paciente ingresa por primera vez al consultorio se registran los datos de forma manual en un cuaderno, el mismo paciente ingresa por segunda vez al consultorio y se repite el mismo proceso ocasionando pérdida de tiempo. Retardo en la generación de reportes: al no existir un proceso automático que realice el registro de los pacientes atendidos diariamente en el departamento, estos informes son solicitados en tiempos imprevistos por los jefes inmediatos. Pérdida de información: los datos históricos de los pacientes atendidos no son archivados de una manera adecuada dando poca importancia a estos registros.

En base a los problemas identificados en el Departamento Médico Odontológico es necesario implementar un Sistema Web de gestión Odontológico con el fin de agilizar la gestión de información en el departamento.

El sistema web permitirá realizar citas médicas, diagnóstico y tratamiento dental, tener un historial del pacientes, así como también una lista de los empleados cuyo contrato a caducado, entre otras opciones más, con el fin de tener un registro de datos actualizados para la generación de reportes correspondientes para la toma de decisiones.

La aplicación web ofrecerá las siguientes ventajas.

- Reducir el tiempo en la emisión de reportes
- Reducir el uso excesivo de papel
- Disponibilidad de la información
- Desde el punto de vista técnico; flexibilidad y mantenibilidad.

El sistema web que se pretende desarrollar contendrá los siguientes módulos.

- Módulo Gestión de Usuarios: Este módulo contendrá la asignación de rol de usuarios como: Administrador, Dentista, Asistente, Paciente. Así como también la asignación de acceso al sistema, de la misma manera denegar el acceso a los usuarios.
- Módulo Gestión de Pacientes: Este módulo permitirá la gestión de los datos de los pacientes, aquí podrá ver a los pacientes cuyo contrato de trabajo ha caducado los cuales no serán atendidos por el odontólogo.
- Módulo Gestión de Citas Médicas: Este módulo permitirá el ingreso, modificación, eliminación de citas médicas, también se listarán las citas médicas cuyos pacientes no se han presentado.
- Módulo Gestión de Diagnóstico: En este módulo se podrá verificar si un paciente ya cuenta con su diagnóstico.
- Módulo Gestión de Tratamiento: En este módulo se podrá ver un historial de todos los tratamientos realizados.
- Módulo Gestión de Odontología: En este módulo contará con sub módulos como departamento, horario, diagnóstico, tratamiento.
- Módulo reportes: En este módulo se generarán todo tipo de reportes, y todos los comprobantes que necesite la administración.

Objetivos

En esta sección se describirá los objetivos propuestos para el siguiente trabajo de titulación.

Objetivo General

- Desarrollar un sistema web de gestión odontológico para la Empresa Eléctrica Riobamba S.A. utilizando herramientas de software libre.

Objetivos Específicos

- Analizar el funcionamiento del departamento odontológico de la EERSA.
- Estudiar la arquitectura, estructura y los componentes del Framework Spring y las herramientas que integran la aplicación.
- Construir un sistema web que se ajuste a las necesidades del departamento odontológico y que sea factible de modificaciones ante cambios de políticas y reglamentos.
- Mejorar el manejo de la gestión de información en el departamento odontológico mediante la implementación del sistema web.

Métodos y Técnicas

A continuación se describen los métodos y técnicas a emplearse en el siguiente trabajo de titulación.

Métodos

Para el desarrollo del presente trabajo de titulación se aplicó el Método Inductivo que consta de los siguientes pasos.

- Observación
- Experimentación
- Comparación
- Abstracción
- Generalización
- La metodología que se empleará en el desarrollo de software es SCRUM.

Técnicas

Las técnicas para la recolección de información que se van a utilizar para la elaboración del presente trabajo de titulación son las siguientes.

- Observación
- Entrevistas
- Revisión de Documentos
- Encuesta

CAPÍTULO I

1 MARCO TEÓRICO

En este apartado se hace una conceptualización de todas las tecnologías, metodología, herramientas básicas y necesarias para el desarrollo de la aplicación web.

1.1 Aplicaciones Web

Una aplicación web es un programa informático que se ejecuta parcialmente en un servidor remoto, al que se accede a través de Internet o intranet por medio de un navegador web (Moreira Gibaja, 2009, p 1). Dicho de otra manera es una aplicación de software diseñada y desarrollada en cualquier lenguaje de programación bajo requisitos específicos y se puede acceder a esta aplicación a través de una red utilizando un web browser.

1.1.1 Características

- Accesibilidad desde cualquier lugar, si está en red.
- El usuario solo necesita de un browser para utilizar la aplicación web.
- Múltiples usuarios utilizan la misma aplicación web.
- Gran cantidad de tecnologías para el desarrollo.
- Se puede cambiar de tecnologías con las que fue desarrollado.
- Fácil de actualizar y mantener.

1.1.2 Ventajas

- Disponibilidad de la aplicación hacia los clientes.
- Los datos son centralizados en el servidor por lo que es fácil realizar respaldos de la información.
- Actualizaciones transparentes para los clientes.
- Estas aplicaciones se pueden ejecutar en distintas plataformas como Windows, Linux, Mac.
- Mayor demanda en el mercado.

1.1.3 Desventajas

- Mayor tiempo en la etapa de desarrollo.
- Riesgos de seguridad de la información.

Las aplicaciones web, se han convertido en herramientas útiles para la gestión de información, ayudando a reducir recursos como tiempo y dinero.

1.1.4 Arquitectura en capas

El objetivo de construir aplicaciones en capas es minimizar las dependencias entre estas (presentación, lógica de negocio y acceso a datos), para tener un código flexible y de fácil mantenimiento, así como también una aplicación bien estructurada ver la **Figura 1-1**.

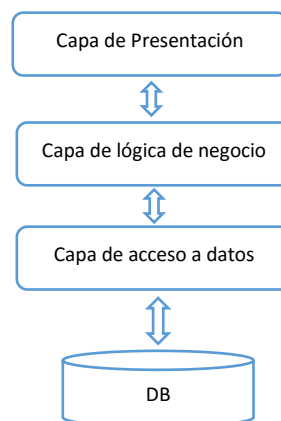


Figura 1-1: Arquitectura en capas
Realizado por: Jesús Mesías Ponluiza Horta. 2016

1.1.4.1 Capa de Presentación

Esta capa realiza la interacción entre el usuario y la aplicación web también es conocida como interfaz gráfica, es la encargada de la visualización de la información a través de pantallas diseñadas por el desarrollador de software.

1.1.4.2 *Capa de Lógica de Negocio*

Esta capa realiza las funciones principales de la aplicación web, como la ejecución de procesamiento de datos, validaciones, reglas del negocio, esta capa interactúa con la capa de presentación y con la capa de acceso a datos para posteriormente enviar los datos al usuario

1.1.4.3 *Capa de Acceso a Datos*

Esta capa se encarga de la gestión de información en la base de datos, esta capa recibe solicitudes de almacenamiento o recuperación desde la capa lógica del negocio.

1.2 **Patrones de Diseño**

Un patrón de diseño es una solución general de un problema común en el desarrollo de software, es una plantilla de cómo resolver un problema. El uso principal de patrones de diseño es.

- Tener una plataforma común para los desarrolladores.
- Mejores prácticas de programación.

1.2.1 **Inyección de Dependencia**

La inyección de dependencia es un patrón de desarrollo de software orientado a objetos cuyo objetivo es tener un código desacoplado y que sea fácil de mantener. Este patrón es un subconjunto del Inversión de control. Hay tres formas de hacer la inyección de dependencia.

- Inyección de constructor
- Inyección por setter
- Inyección por Interfaz

Al aplicar este patrón lo que hacemos es que las clases sean independientes unas de otras, proporcionando a la aplicación extensibilidad, reusabilidad y mantenibilidad y fácil de realizar pruebas unitarias.

Observemos un simple ejemplo de inyección de dependencia, donde un objeto necesita de otro para su funcionamiento, el objeto Cliente tiene una dependencia con Servicio. Ver **Figura 2-1**.

```

public class Service
{
    public void Serve()
    {
        // do some serving stuff
    }
}

public class Client
{
    private Service _service;

    public Client()
    {
        this._service = new Service();
    }

    public void Start()
    {
        this._service.Serve();

        // do some client stuff
    }
}

```

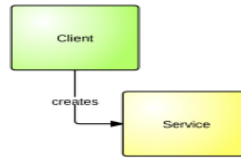


Figura 2-1: Acoplamiento Fuerte
Fuente: Flohr, 2011.

Aplicando la inyección de dependencia se obtiene un código débilmente acoplado haciendo de nuestra aplicación más flexible y utilizable. Ver la **Figura 3-1**.

```

public interface IService
{
    public void Serve();
}

public class Service : IService
{
    public void Serve()
    {
        // do some serving stuff
    }
}

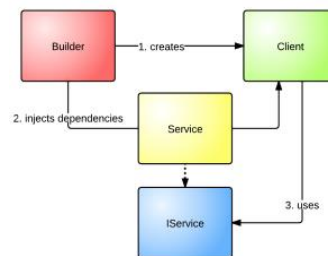
public class Client
{
    private IService _service;

    public Client(IService service)
    {
        this._service = service;
    }

    public void Start()
    {
        this._service.Serve();

        // do some client stuff
    }
}

```



```

var client = new Client(new Service());
client.Start();

```

Figura 3-1: Débilmente Acoplado
Fuente: Flohr, 2011.

Se está inyectando la dependencia en el constructor, pasando el Service que implementa la interfaz IService. Las dependencias se ensamblan por un Builder.

1.2.2 Ventajas

- Orientado a cualquier lenguaje de programación.
- Resuelve problemas similares pero en entornos diferentes.
- Proporcionan un estándar de codificación.

1.2.3 Desventajas

- No es un diseño final que se pueda transformar en código.

1.3 Framework

Un Framework Web es un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web (Gutiérrez 2014, p 1).

1.3.1 Tipos de Framework

Hoy en día existen una gran cantidad de Frameworks que nos proporcionan diversas ventajas al momento de desarrollar aplicaciones tanto de escritorio como orientado a la web por ejemplo.

- Para la interfaz de usuario JSF con Primefaces, ZK.
- Para la lógica de negocio Spring framework.
- Para la persistencia de datos: Hibernate, MyBatis, Entity Framework, nHibernate.
- Para pruebas JUnit, TestNG, JTiger.

La mayoría de frameworks Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC o con el modelo 2 de Servlets y JSP, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación (Gutiérrez 2014, p 3).

1.3.2 Framework web para Java

Spring: es un framework para el desarrollo de aplicaciones Java y contenedor de inversión de control, de código abierto creado por Rod Johnson.

Vaadin: es un framework Java que permite crear aplicaciones web modernas sin necesidad de plugins, además hace posible a los desarrolladores enfocarse principalmente en la lógica y dejar el HTML, JavaScript y XML al framework. Es muy importante mencionar que Vaadin está basado en Google Web Toolki (Tovar Salazar 2010, p 1).

Google Web Toolkit: framework creado por google para acelerar el desarrollo rápido de aplicaciones RIA y mejorar la experiencia del usuario, permitiendo a los desarrolladores utilizar las herramientas java existentes para construir aplicaciones Ajax independientes del navegador (Bravo Garcia 2013, p 2).

Tapestry: es un marco de trabajo para java que está orientado completamente a objetos para implementar aplicaciones acorde a la arquitectura mvc.

Hibernate: es un framework de persistencia de datos, que facilita el mapeo de atributos de la base de datos relacional de código abierto.

JSF: es un marco de trabajo para el desarrollo de aplicaciones web bajo la arquitectura mvc, existen varias implementaciones de este framework para el manejo de componentes.

- IceFaces
- RichFaces
- PrimeFaces
- OpenFaces
- ADF
- Zk

Struts: es un framework de código abierto que se extiende la API Java Servlet y emplea el patrón Modelo, Vista, Controlador (MVC). Permite crear aplicaciones web mantenibles, extensibles y flexibles basados en tecnologías estándar, como las páginas JSP, JavaBeans, paquetes de recursos, y XML (Netbeans 2016).

1.3.3 Ventajas

- Reduce el tiempo en el desarrollo del software.
- Utilizan patrones de diseño.
- Amplia documentación y equipos de apoyos, foros de comunidad donde podemos encontrar soluciones rápidas.
- Los Framework nos proporcionan una estructura predefinida para nuestra aplicación.
- Código testeado.
- Los desarrolladores pueden comprender la aplicación en un tiempo más corto.

1.3.4 Desventajas

- Dedicar tiempo en el aprendizaje por cada Framework que se va a utilizar.
- Limitaciones a la manera de trabajar el Framework.
- El Código es público está disponible para todo el mundo.

El uso de frameworks ayuda al desarrollador de software a crear aplicaciones estandarizadas, testeadas permitiendo el desarrollo rápido de las aplicaciones este factor se puede evaluar en base a parámetros como si el equipo de desarrollo conoce el Framework o si existen capacitaciones constantes en ciertos frameworks que se utilice.

1.4 Por qué Spring, Hibernate y Primefaces

Estas herramientas fueron seleccionadas para el desarrollo de la aplicación web de gestión odontológico por ser de libre uso en otras palabras, se las puede descargar gratuitamente de las páginas oficiales de cada una de estas, y esto a su vez da por cumplimiento con el objetivo general de este trabajo de titulación que es utilizar herramientas de software libre. Para construir la aplicación web se optó por Spring ya que es un Framework web que se acopla a la aplicación que se está desarrollando dicho de otra manera, la aplicación no depende de las clases propias de Spring, de la misma manera este marco de trabajo impulsa a construir aplicaciones web débilmente acopladas, otro de los motivos importantes es la facilidad de integración con otras herramientas especialmente con la tecnología Hibernate. De igual manera se prefirió Hibernate como Framework de persistencia ya que agiliza la relación entre la aplicación desarrollada y base de datos. Para la interfaz de usuario se seleccionó Primefaces ya que este Framework proporciona varios componentes con comportamiento Ajax para la creación de interfaces enriquecidas

1.5 Spring

1.5.1 Introducción

Framework Spring fue escrito por Rod Johnson y fue lanzado por primera vez bajo la licencia Apache 2.0 en junio de 2003. Spring Framework es una solución liviana en cuanto a su peso, pero robusta para el desarrollo de aplicaciones de software, Spring es modular, lo que permite al desarrollador hacer uso solo de los módulos que necesite en su proyecto, ignorando el resto.

El objetivo de Spring es no ser intrusivo, aquellas aplicaciones configuradas para usar beans mediante Spring no necesitan depender de interfaces o clases de Spring, pero obtienen su

configuración a través de las propiedades de sus beans. Este concepto puede ser aplicado a cualquier entorno, desde una aplicación JEE a un applet.

El paquete “Core” es la parte más fundamental del Framework ya que proporciona las características de Inversión de Control (IoC) e Inyección de dependencias (ID). El concepto básico dentro del “Core” es el “BeanFactory”, el cual proporciona una sofisticada implementación del patrón de diseño "Factory", que elimina la necesidad generalizada de “Singletons” y nos permite desacoplar la configuración y especificación de las dependencias de nuestra lógica de programación (Juntadeandalucia 2014, p 1).

1.5.2 Definición

Spring es un Framework liviano para el desarrollo de aplicaciones empresariales basado en la técnica de inversión de control, es de código abierto, aplica el patrón de diseño inyección de dependencia, proporcionando objetos reutilizables y débilmente acoplados.

1.5.3 Características

Las características de este Spring son.

- La Inyección de Dependencias.
- Framework Open Source
- Contenedor ligero IoC.
- Integración con otros frameworks y librerías.
- Promueve buenas prácticas de programación (uso de patrones de diseño) y diseño creando aplicaciones eficientes de calidad y mantenibilidad.
- La Programación Orientada a Aspectos.
- El código escrito no debe depender de las APIs de Spring.

1.5.4 Ventajas

- Spring está organizado de forma modular. A pesar de que el número de paquetes y clases son sustanciales, usted tiene que preocuparse solamente sobre los que necesita e ignorar el resto.
- Spring no trata de reinventar la rueda, lo que realmente hace es uso de algunas de las tecnologías existentes como Frameworks ORM (Hibernate), JEE, entre otras tecnologías.
- Spring es un Framework web MVC bien diseñado.
- Estándares de programación.
- Acoplamiento débil.

- Programación basada en POJOS.

1.5.5 Desventajas

- La curva de aprendizaje es pronunciada.

1.5.6 Inversión de Control

IoC se expresa en el Principio de Hollywood: “No me llames, yo te llamo”, también se conoce como inyección de dependencias (DI). Es un proceso mediante el cual los objetos definen sus dependencias, es decir, los otros objetos con los que trabajan, sólo a través de argumentos del constructor, argumentos a un método Factory, o método setter que son invocados después de que el objeto se ha construido. En este caso en vez de ser el mismo objeto quien se encargue de instanciar, o localizar, las dependencias con las que trabaja (usando directamente su constructor o un localizador de servicios), es el contenedor el que inyecta estas dependencias cuando crea al bean. Como podemos notar el proceso es inverso al que normalmente se hace, y de ahí el nombre de Inversión de Control.

Los paquetes `org.springframework.beans` y `org.springframework.context` son la base para el contenedor IoC del Framework Spring. La interfaz `BeanFactory` proporciona un mecanismo de configuración avanzada capaz de gestionar cualquier tipo de objeto. `ApplicationContext` es una sub-interfaz de `BeanFactory`. Añade una integración más fácil con la característica AOP, manejo de recursos de mensajes (para la internacionalización), publicación de eventos, y contextos específicos para ciertas capas de aplicaciones.

En resumen el `BeanFactory` proporciona la configuración y funcionalidad básica del Framework y el `ApplicationContext` añade más funcionalidad específica de la empresa. El `ApplicationContext` es un súper conjunto completo del `BeanFactory` (Johnson, et al. 2013, p 22).

IoC es una de las características fundamentales que se espera que sea proporcionada por cualquier contenedor. Tiene básicamente dos formas: Búsqueda de la Dependencia y la Inyección de Dependencia (Caliskan et al. 2015, p 9).

1.5.7 Inyección de Dependencia con Spring

Inyección de Dependencia es una forma de implementación de IoC, que elimina la dependencia explícita sobre las APIs de los contenedores, métodos. La frase Inyección de Dependencia puede sonar intimidante, evocando nociones de una técnica de programación compleja. Pero resulta que

la DI no es tan compleja como parece, mediante la aplicación de la DI en los proyectos de software, el código será mucho más simple, fácil de entender, y más fácil de hacer pruebas (Walls 2015, p 5).

Cuando usamos el mecanismo de DI de Spring, el código es más limpio, y está desacoplado de una forma más efectiva. El objeto no busca sus dependencias de hecho ni siquiera conoce la ubicación o clase de su dependencia. Por esta misma razón, las clases se hacen más fáciles de probar, en particular cuando las dependencias las definimos como interfaces o clases abstractas (Johnson et all. 2013, p 32).

Existen tres métodos de Inyección de Dependencia principales:

- DI basada en Constructor
- DI basada en Setter
- DI basada en Interfaz

1.5.7.1 DI basada en constructor

Los beans expresan sus dependencias a través de argumentos del constructor. En este método, las dependencias se inyectan durante la creación del componente. Se puede inyectar propiedades simples, tales como valores int o boolean como argumentos del constructor. La ventaja de la inyección de constructor es que cada componente administrado en el contenedor está garantizado a estar en un estado coherente y listo para usar después de su creación. Otro punto positivo es que la cantidad de código escrito con inyección de constructor será ligeramente menor en comparación con el código escrito cuando se utiliza la inyección de setter.

1.5.7.2 DI basada en Setter

Los métodos setter se invocan inmediatamente después de que el objeto es instanciado por el contenedor. La inyección se produce durante la fase de creación de componentes o de inicialización, que se realiza mucho antes en el proceso de gestión de las llamadas de métodos comerciales. La ventaja más importante de la inyección de setter es que permite la reconfiguración del componente después de su creación. Las dependencias de los componentes se pueden cambiar en tiempo de ejecución. Muchas clases ya existentes se pueden utilizar con la programación de

estilo JavaBean estándar. En otras palabras, ofrecen métodos get y set para acceder a sus propiedades (Caliskan et al. 2015, p 11).

A continuación vemos un ejemplo de aplicación de la inyección de dependencia con Spring configurado con anotaciones.

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
@Service("datosPersonas")
public class DatosPersonas
{
    private IPersonaDao personasDao;
    @Autowired
    public void setPersonaDao(IPersonaDao personasDao)
    {
        this.personasDao = personasDao;
    }
    public void escribirDatospersona(String id)
    {
        personasDao.leerPersonaDao(id);
    }
}
```

Como podemos ver la clase DatosPersonas no está acoplada a ninguna clase concreta pero si a la interfaz. El contenedor de spring es el que inyectara una implementación de la interfaz.

@Service: indica que es un bean gestionado por spring.

@Autowired: indica a spring que debe pasar en el método setPersonaDao una instancia de un objeto que implementa la interfaz.

La interfaz seria de la siguiente manera.

```
-
public interface IPersonaDao
{
    public void leerPersonaDao(String id);
}
```

La clase PersonaDao que implementa la interfaz IPersonaDao seria de la siguiente manera.

```
import org.springframework.stereotype.Service;
@Service
public class PersonaDao implements IPersonaDao
{
    @Override
    public void leerPersonaDao(String id)
    {
        System.out.println("Datos de la persona: " + id);
    }
}
```

El applicationContext.xml archivo de configuración de Spring se crea en la carpeta WEB-INF, donde en el context:component-scan se especifica el paquete donde se encuentra los servicios que sería de la siguiente manera.

```
<beans xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.springframework.org/schema/beans"
xsi:schemalocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">
<context:component-scan base-package="spring.java">
</context:component-scan></beans>
```

La clase para probar la aplicación sería de la siguiente manera.

```
import org.springframework.beans.factory.BeanFactory;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import spring.java.DatosPersona;
    public class TestSpring {

        public static void main(String args[]) {
            ApplicationContext context = new ClassPathXmlApplicationContext("WEB-
            INF/applicationContext.xml");
            BeanFactory factory = context;
            DatosPersonas p = (DatosPersonas) factory.getBean("datosPersonas");
            p.escribirDatospersona("Jesus");
        }
    }
```

1.5.8 Spring Bean

En Spring los objetos forman la columna vertebral de la aplicación, y son administrados por el contenedor de IoC de Spring. Un bean es un objeto que es instanciado, ensamblado (cuando sus dependencias son inyectadas), y en general, administrado por el contenedor de IoC. Dicho de otra forma un bean es simplemente uno de muchos objetos de nuestra aplicación. Los Beans, y las dependencias entre ellos, las declaramos en los metadatos del contenedor, por ejemplo en forma de xml (Johnson et al. 2013, p 27).

- Como se crean
- Detalles del ciclo de vida
- Dependencias

1.5.9 Módulos de Spring

Spring Framework se puede descargar libremente de su página oficial, solo hay que tener en cuenta con que versión vamos a trabajar, en este caso se trabajara con la versión 4.0.1 para posteriormente añadir estas librerías (jar) al proyecto, dando soporte a la aplicación para trabajar con Spring. Podemos ver en la **Figura 4-1**.

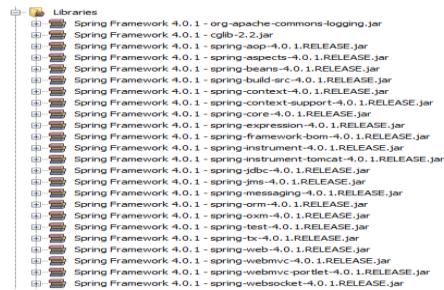


Figura 4-1: Librerías de Spring Framework
Realizado por: Jesús Mesías Ponluiza Horta. 2016

El Spring Framework consta de funciones organizadas en cerca de 20 módulos. Estos módulos se agrupan en; Core Container, acceso a datos / Integración, Web, AOP (Aspect Oriented Programming), Instrumentación, y prueba, como se muestra en la **Figura 5-1**.

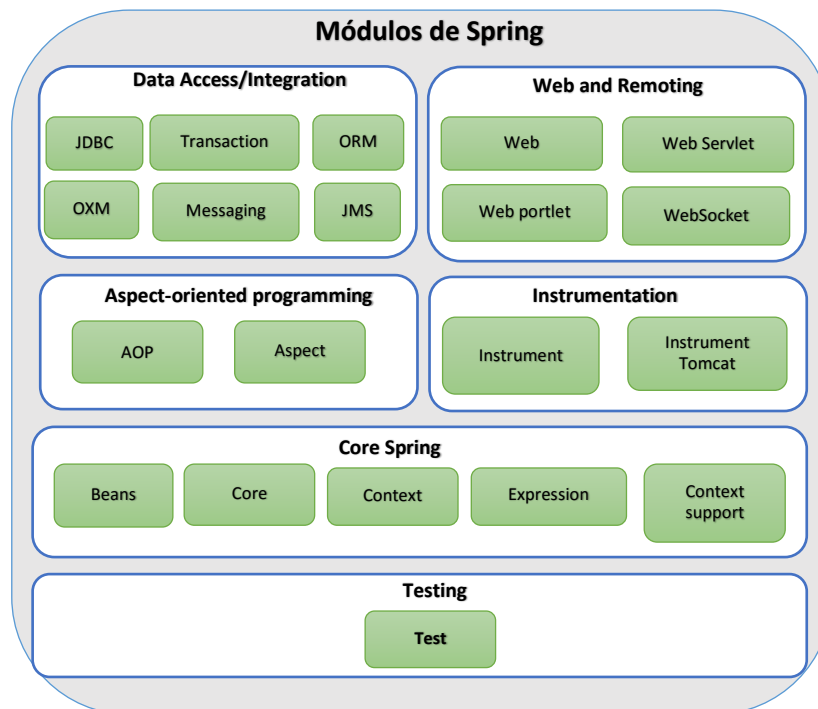


Figura 5-1: Módulos de Spring Framework
Fuente: Walls, 2015.

La mejor opción de Spring es que podemos elegir los módulos que se adapten a nuestra aplicación. Spring incluso se integra con otros Frameworks. A continuación en la siguiente **Tabla 1-1** se describen los componentes de cada uno de los módulos del Framework Spring.

Tabla 1-1: Componentes de Spring Framework

| Aspect-oriented programming | |
|------------------------------------|--|
| Aop | Este módulo permite programación orientada aspectos. |
| Aspects | Este módulo proporciona integración con AspectJ. |
| Core Spring Container | |
| Beans | El módulo proporciona un BeanFactory que es una aplicación sofisticada del patrón Factory, instancia los beans registrados en el framework. |
| Context | Contexto de la aplicación, podemos ver los objetos definidos y su configuración. |
| context-support | En este módulo contiene librerías de terceros como Velocity, FreeMarker y JasperReports. |
| Core | Este módulo es principal para la aplicación de Spring, incluye características de IoC y de Inyección de Dependencia. |
| Expression | Este módulo proporciona un lenguaje para consultar y manipular objetos en tiempo de ejecución. |
| Instrumentation | |
| Instrument | Agente de instrumentación para la JVM bootstrapping. |
| instrument-tomcat | Agente de instrumentación para Tomcat. |
| Data Access/Integration | |
| Jdbc | Este módulo proporciona una capa de abstracción JDBC. |
| Jms | El módulo de servicio de mensajería de Java (JMS) contiene características para producir y consumir mensajes. |
| Messaging | Soporte para mensajería arquitecturas y protocolos. |
| Orm | Este módulo proporciona integración con frameworks más populares para el mapeo de los datos como: Hibernate, JDO, JPA, MyBatis, etc. |
| Oxm | El módulo de OXM proporciona una capa de abstracción que soporta implementaciones de mapeo objeto / XML para JAXB, Castor, XMLBeans, JiBX y xstream. |
| Testing | |
| Test | Los componentes de Spring soportan pruebas con otros frameworks como JUnit o TestNG. |
| Web and Remoting | |
| Web | Este módulo proporciona funciones de integración web básica como: inicialización del contenedor IoC, un application context orientado a la web. |
| Webmvc | Este módulo contiene todas las clases de MVC propias del Framework Spring. |
| web-portlet | Este módulo proporciona la implementación MVC para ser utilizado en un entorno de portlet y refleja la funcionalidad del módulo de Web-Servlet. |
| Websocket | Este módulo proporciona soporte para la API Java para WebSocket. |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

El paquete Core el contenedor fundamental de Spring el cual proporciona funciones específicas como podemos ver en la **Figura 6-1**.

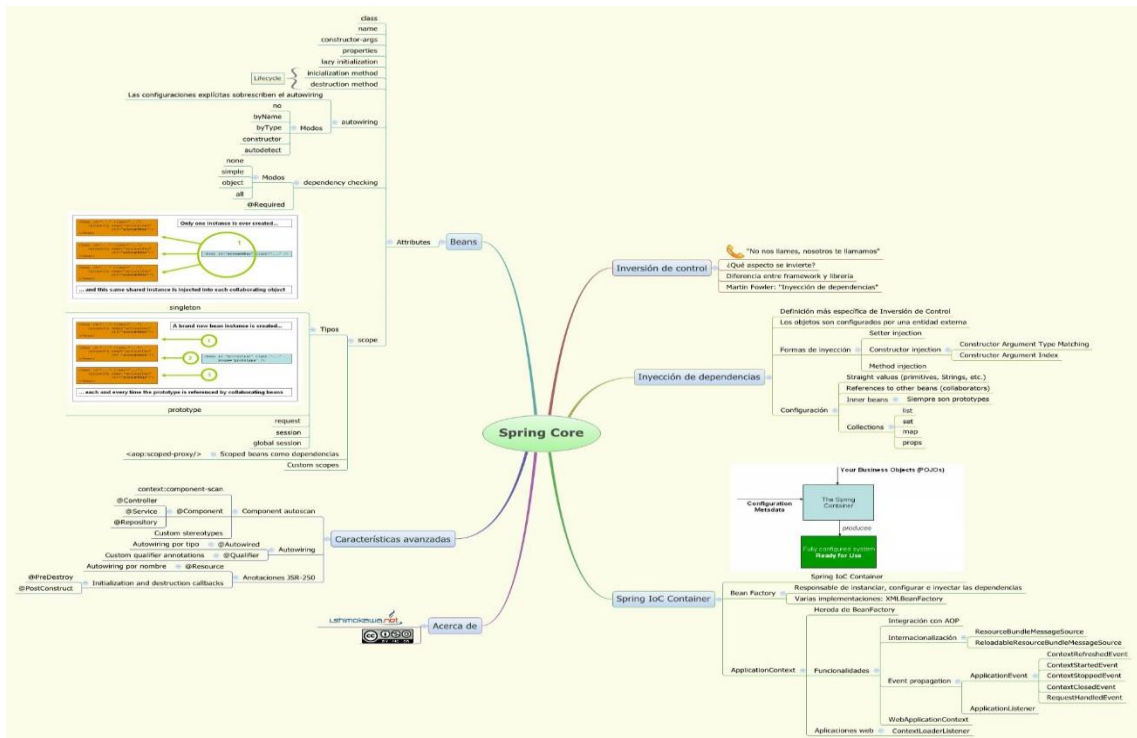


Figura 6-1: Spring Core
Fuente: Shimokawa, 2010.

1.5.10 Configuración de la Metadata de Spring

Los metadatos es un conjunto de información de características de los beans, a través de esta configuración el contenedor de Spring los gestiona.

Los metadatos de configuración para el contenedor de Spring pueden ser representados de tres distintas maneras.

- Configuración basada en archivos XML
- Configuración basada en anotaciones
- Configuración basada en Java

A continuación se presenta un archivo de configuración basado en XML.

```
<?xml version="1.0" encoding="UTF-8"?>
  <beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd">
    <bean id="..." class="...">
      <!-- collaborators and configuration for this bean go here -->
    </bean>
    <bean id="..." class="...">
      <!-- collaborators and configuration for this bean go here -->
    </bean>
  </beans>
```

```
</bean>
<!-- more bean definitions go here -->
</beans>
```

1.5.11 Spring-IoC Container

El contenedor de Spring es el núcleo del Framework, ya que es aquí donde se gestionan todos los objetos (Beans) desde su creación hasta su destrucción. Este contenedor hace uso de la Inyección de Dependencia para gestionar todos los componentes que conforman la aplicación. El contenedor recibe las instrucciones sobre qué objetos crear una instancia, configurar mediante la lectura de los metadatos de configuración. La siguiente **Figura 7-1** es una vista de alto nivel de cómo funciona Spring.

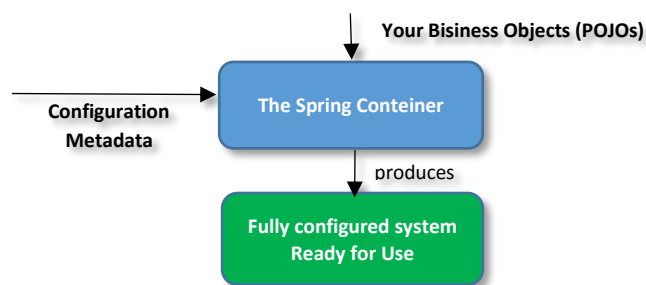


Figura 7-1: Contenedor de Spring IoC
Fuente: Johnson et al. 2013, p 23.

El contenedor de Spring IoC hace uso de las clases de Java POJO y metadatos de configuración para producir un sistema o aplicación totalmente configurable y ejecutable. Spring proporciona dos distintos tipos de contenedores el BeanFactory y ApplicationContext.

1.5.12 BeanFactory

BeanFactory es una implementación del patrón de diseño Factory, es una clase cuya responsabilidad es crear Beans. Este es el contenedor más simple que proporciona soporte básico para DI y definida por la interfaz `org.springframework.beans.factory.BeanFactory`. La aplicación BeanFactory más comúnmente utilizada es la clase `XmlBeanFactory`. Este contenedor lee los metadatos de configuración desde un archivo XML y lo usa para crear un sistema o aplicación completamente configurada. El BeanFactory normalmente se recomienda usar para aplicaciones de peso ligero, cuando los recursos son limitados, como dispositivos móviles o aplicaciones basadas en applet (Johnson et al. 2013).

1.5.13 **ApplicationContext**

Un "**BeanFactory**" está bien para aplicaciones simples, pero no toma ventaja de todo el poder que nos proporciona el framework de **Spring**, y como desarrolladores queremos cargar los beans de nuestra aplicación usando el contenedor más avanzado que nos proporciona **Spring**, el **ApplicationContext**.

Usar "**ApplicationContext**" es similar a usar "**BeanFactory**". Ambos cargan definiciones de beans, cablean los beans y los envían a quienes se los piden. Pero "**ApplicationContext**" ofrece más cosas.

- Proporciona un medio para resolver mensajes de texto (no, no son SMSs), incluyendo soporte para internacionalización (I18N) de estos mensajes.
- Proporciona una manera genérica de cargar archivos de recursos, como imágenes.
- Puede publicar eventos a beans que están registrados como listeners.

Debido a que "**ApplicationContext**" proporciona más funcionalidades, es preferible el uso de este sobre "**BeanFactory**" en casi todo tipo de aplicaciones, la única excepción es cuando estamos ejecutando una aplicación donde los recursos son escasos, como en un dispositivo móvil.

Las implementaciones más comúnmente usadas de "**ApplicationContext**" son:

- **ClassPathXmlApplicationContext:** Carga la configuración desde un archivo XML ubicado en el classpath.
- **FileSystemXmlApplicationContext:** Carga la configuración desde un archivo XML ubicado en cualquier parte del sistema de archivos de la computadora.
- **XmlWebApplicationContext:** Carga la configuración desde un archivo XML ubicado en el contexto de una aplicación web.

1.5.14 **Spring ORM**

Spring Framework soporta la integración con Hibernate, Java Persistence API (JPA) y Java Data Objects (JDO) para la gestión de recursos, objeto de acceso a datos (DAO) implementaciones, y las estrategias de transacción. Para Hibernate hay un gran apoyo por parte de Spring con varias características convenientes a IoC (Johnson et al. 2013, p 396).

1.5.14.1 Ventajas

- Facilidad para realizar pruebas.
- Excepciones de acceso a datos comunes.
- Manejo de recursos genérico.
- Mapeo de transacciones integrado.

1.5.15 Spring MVC

Spring proporciona un Framework de desarrollo web basado en el paradigma Modelo-Vista-Controlador (MVC), ofrece componentes listos que se pueden utilizar para desarrollar aplicaciones web flexibles y débilmente acoplados. El patrón MVC se traduce en la separación de los diferentes aspectos de la aplicación (lógica de entrada, la lógica de negocio, y la lógica de interfaz de usuario), mientras que proporciona un acoplamiento débil entre estos elementos. En la **Figura 8-1** se muestra el diagrama de funcionamiento del patrón MVC.

Modelo encapsula los datos de la aplicación y, en general, consistirá en POJO.

Vista es el encargado de mostrar los datos del modelo y, en general, se genera una salida HTML que el navegador del cliente puede interpretar.

Controlador es responsable de procesar las peticiones del usuario y la construcción de modelo apropiado y lo pasa a la vista para la representación.

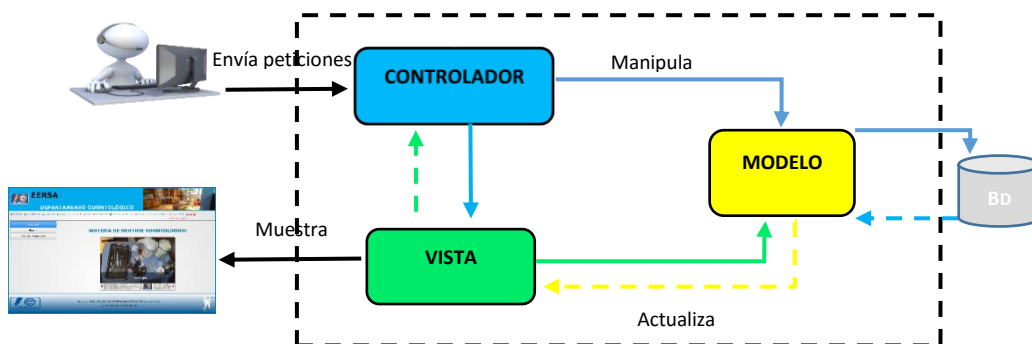


Figura 8-1: Modelo Vista Controlador
Realizado por: Jesús Mesías Ponluiza Horta. 2016

1.6 Hibernate

Hibernate es una herramienta de mapeo objeto/relacional para entornos Java de código libre creado por Gavin King. El término de mapeo objeto/relacional (ORM) se refiere a la técnica de mapear una representación de datos desde un modelo de objeto a un modelo de datos relacionales con un esquema basado en SQL. Hibernate no solamente se ocupa del mapeo desde las clases Java a las tablas de las bases de datos (y desde los tipos de datos de Java a los tipos de datos de SQL), sino que también facilita la consulta y recuperación de datos. Esto puede reducir de manera importante el tiempo de desarrollo que se tomaría con el manejo de datos de forma manual en SQL y JDBC (King et al. 2010).

1.6.1 Características

- Licencia LGPL
- Tiene sus propio lenguaje de consulta HQL
- Se puede implementar con ficheros XML o Anotaciones

1.6.2 Base de Datos soportadas

- Oracle
- MySQL
- PostgreSQL
- DB2/NT
- FrontBase
- Microsoft SQL Server
- Sybase SQL Server
- Informix Dynamic Server
- HSQL Database Engine

1.6.3 Ventajas

- Facilidad de programación.
- Las líneas de código son mínimas lo que hace fácil de entender y mantener.
- Portabilidad, se puede cambiar fácilmente a otras bases de datos.
- Fácil de configurar.

1.6.4 Desventajas

- La funcionalidad se reduce con respecto a las consultas nativas.
- Tiempo en aprender el Framework.

1.6.5 Arquitectura de Hibernate

La arquitectura Hibernate es en capas, está formado por varias APIs de Java como JDNI, JDBC, JTA. **Figura 9-1.**

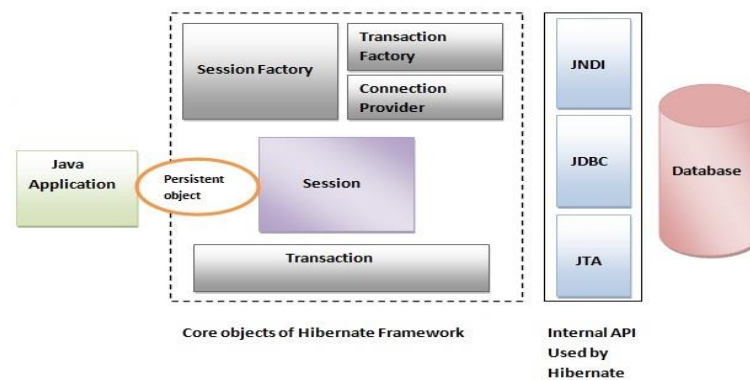


Figura 9-1: Arquitectura de Hibernate

Fuente: javatpoint, 2014

A continuación las definiciones de los elementos Hibernate (King et al. 2010).

1.6.5.1 *SessionFactory*

Es una fábrica de Session y un cliente de ConnectionProvider, SessionFactory puede mantener un caché opcional de datos reusables entre transacciones a nivel de proceso o de clúster.

1.6.5.2 *Session*

Es un objeto mono-hebra, de corta vida que representa una conversación entre la aplicación y el almacenamiento persistente. Envuelve una conexión JDBC y es una fábrica de Transaction. Session mantiene un caché requerido de primer nivel de objetos persistentes, que se utiliza cuando se navega el gráfico de objetos o mientras se buscan objetos por identificador.

1.6.5.3 *Transaction*

Es un objeto de corta vida, mono-hebra que la aplicación utiliza para especificar unidades atómicas de trabajo. Abstrae la aplicación de las transacciones subyacentes JDBC, JTA o CORBA, es opcional.

1.6.5.4 *ConnectionProvider*

Una fábrica de conexiones JDBC. Abstrae a la aplicación del DataSource o DriverManager subyacente, es opcional.

1.6.5.5 *TransactionFactory*

Una fábrica de instancias de Transaction. No se expone a la aplicación pero puede ser extendido/implementado por el desarrollador.

1.6.6 HQL

Hibernate Query Language es un lenguaje de consultas orientado a objetos de Hibernate es similar a Sql a diferencia de Sql, Hql trabaja con objetos persistentes y sus propiedades es decir con las entidades que se persisten en Hibernate. Las palabras reservadas del lenguaje como Select, from, where, etc no son sensibles a mayúsculas y minúsculas, pero el nombre de las clases y sus propiedades si lo son. A continuación un ejemplo de una consulta hql.

```
select usr.id, usr.name from User as usr left join usr.messages as msg group by usr.id,  
usr.name order by count(msg)
```

1.6.7 Configuración

Para la configuración de Hibernate se puede hacer de dos maneras mediante un archivo hibernate.properties simple, o un archivo hibernate.cfg.xml un poco más sofisticado. A continuación se muestra las características del archivo hibernate.cfg.xml.

1.6.7.1 *hibernate.cfg.xml*

Este es un fichero de configuración de Hibernate, este xml contiene información como las propiedades de configuración las clases que se van a mapear desde la base de datos, el fichero tiene la siguiente estructura.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD
3.0//EN" "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="connection.url">jdbc:mysql://localhost/hibernate1</property>
    <property name="connection.username">hibernate1</property>
    <property name="connection.password">hibernate1</property>
    <property name="dialect">org.hibernate.dialect.MySQL5Dialect</property>
    <property name="hibernate.show_sql">>true</property>
    <mapping resource="ejemploHibernate/Alumno.hbm.xml"/>
    <mapping class="ejemploHibernate.Alumno"/>
  </session-factory>
</hibernate-configuration>
```

1.6.7.2 *hbm.xml*

Es un archivo de asignación XML Hibernate contiene la relación de correspondencia entre la clase Java y la tabla de base de datos. Esto siempre es nombrado como "xx.hbm.xml" y declarado en el "hibernate.cfg.xml" archivo de configuración de Hibernate (King et al. 2010).

1.7 Java Server Faces

Java Server Faces es un framework de interfaz de usuario para la construcción de aplicaciones web Java. JSF es un framework MVC (Modelo-Vista-Controlador) basado en el API de Servlets que proporciona un conjunto de componentes en forma de etiquetas definidas en páginas XHTML mediante el framework Facelets. Facelets se define en la especificación 2 de JSF como un elemento fundamental de JSF que proporciona características de plantillas y de creación de componentes compuestos. Antes de la especificación actual se utilizaba JSP para componer las páginas JSF (Such 2014, p 1).

1.7.1 Características

- Definición de las interfaces de usuario mediante vistas que agrupan componentes gráficos.

- Conexión de los componentes gráficos con los datos de la aplicación mediante los dominados beans gestionados.
- Conversión de datos y validación automática de las entradas del usuario.
- Navegación entre vistas.
- Internacionalización.
- A partir de la especificación 2.0 un modelo estándar de comunicación Ajax entre la vista y el servidor (Such 2014, p 1).

1.7.2 Primefaces

Primefaces es un conjunto de componentes para Java Server Faces de código abierto con varias extensiones que permiten el desarrollo de aplicaciones web enriquecidas.

1.7.2.1 Características

Primefaces es una potente biblioteca de componentes fácil de utilizar y nos proporciona un conjunto de características para la construcción de interfaces de usuario (Reddy 2013, p 8).

- Más de 100 componentes de IU.
- El soporte integrado AJAX.
- Cero configuraciones, solo necesita de un jar.
- No requiere dependencias de bibliotecas de terceros para la mayor parte de los componentes.
- Integrado con ThemeRoller
- Más de 30 temas fácilmente disponibles y pre-configurados.
- Es compatible con navegadores IE8 +, Chrome, Firefox, Safari y Opera

1.8 Herramientas de apoyo para el desarrollo

1.8.1 MySQL

Es un sistema de gestión de base de datos relacional (RDBMS), de código abierto, lo desarrolla, y distribuye Oracle.

1.8.1.1 Características

- Portabilidad: multiplataforma.
- Es muy rápido, fiable y fácil de usar.
- Es de código abierto.
- Es ideal para aplicaciones pequeñas y medianas.
- Es fácil de instalar y configurar.
- Puede ser integrado con diferentes lenguajes de programación.
- No requiere especificaciones altas en hardware.

1.8.2 Netbeans

Netbeans es un Entorno de Desarrollo Integrado (IDE) escrito en el lenguaje de programación Java de código abierto, para el desarrollo de aplicaciones de software. Es una herramienta donde los desarrolladores de software pueden escribir, compilar, depurar y ejecutar aplicaciones, está escrito en Java y soporta varias tecnologías como PHP, AJAX, Ruby, SOA, Java SE, Java EE, XML, UML, Spring, Hibernate, entre otras. En la **Figura 10-1** se puede ver la versión de Netbeans que se va a utilizar en el desarrollo de este proyecto.

1.8.2.1 Características

- Gestión de la Interfaz de Usuario.
- Herramientas para desarrolladores de Java EE (Frameworks).
- Permite un desarrollo modular organizado, ya que proporciona una plantilla para empezar a desarrollar nuestro proyecto.
- Soporta una gran variedad de tecnologías.
- Fácil y extensible a través de plugins.
- Completación de código Javadoc.
- Amplia documentación.

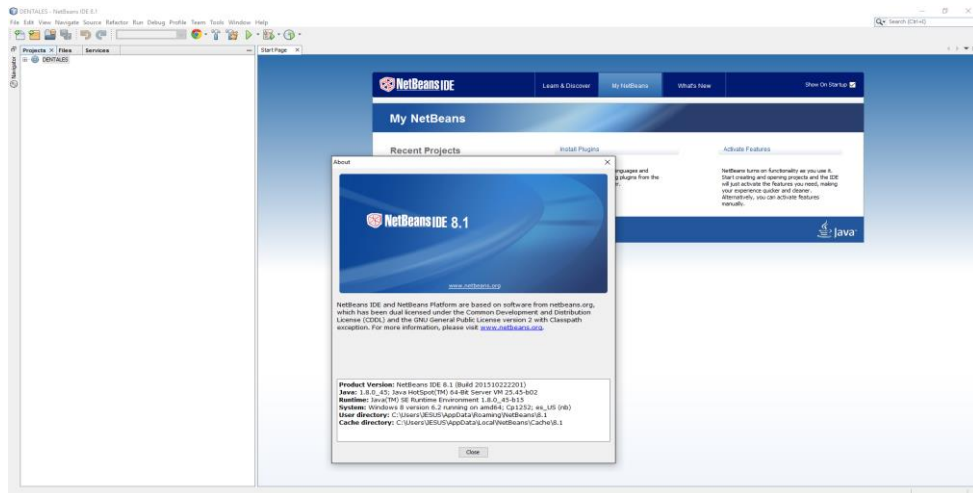


Figura 10-1: Netbeans
Realizado por: Jesús Mesías Ponluiza Horta. 2016

1.8.3 GlassFish

GlassFish es un servidor de aplicaciones para Java EE y permite el desarrollo y despliegue de las aplicaciones, es de código libre, se distribuye bajo la licencia GNU GPL, en la siguiente **Figura 11-1** podemos ver la consola de administración de GlassFish.

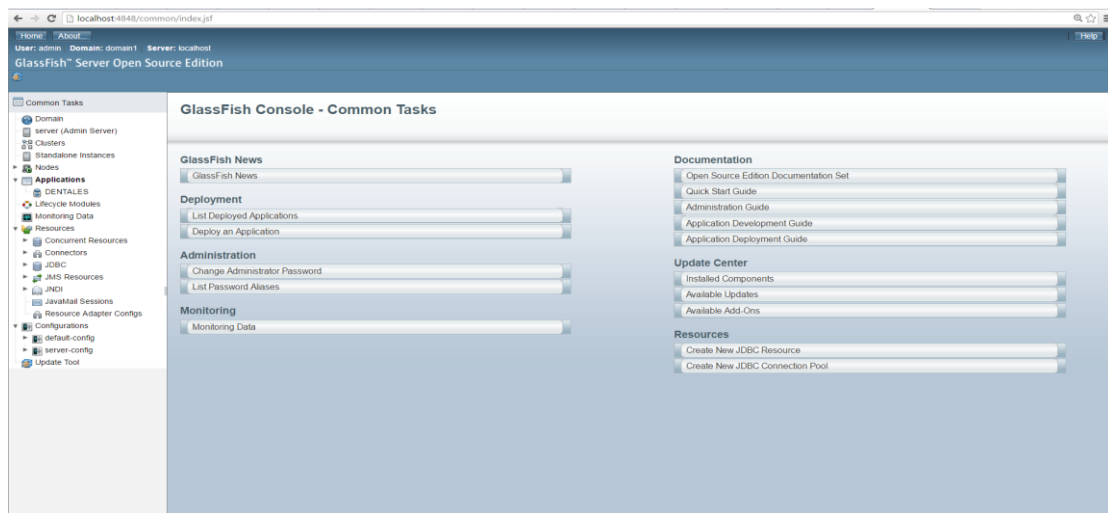


Figura 11-1: GlassFish
Realizado por: Jesús Mesías Ponluiza Horta. 2016

1.9 Metodología Scrum

Scrum es una metodología de desarrollo muy simple que requiere trabajo duro, porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto (Palacio 2006, p 2). Como método ágil.

- Es un modo de desarrollo adaptable, antes que predictivo.
- Orientado a las personas, más que a los procesos.
- Emplea el modelo de construcción incremental basado en iteraciones y revisiones.

1.9.1 Porque Scrum

Se seleccionó esta metodología ágil por que no exige documentar para iniciar el proyecto y por otro lado Scrum ayuda a planificar los requerimientos mediante sprint los cuales deben ser resueltos en tiempos cortos donde el cliente (dentista) pueda ver los avances del producto cubriendo todas sus expectativas con el fin de quedar satisfecho y en este caso evitando pérdidas de tiempo por requerimientos mal interpretados.

1.9.2 Visión general del proceso

Scrum denomina “sprint” a cada iteración de desarrollo y según las características del proyecto y las circunstancias del sprint puede determinarse una duración desde una hasta dos meses, aunque no suele ser recomendable hacerlos de más de un mes. El sprint es el núcleo central que proporciona la base de desarrollo iterativo e incremental (Palacio 2014, p 24-25). Para comprender la visión general del proceso ver la **Figura 12-1**.

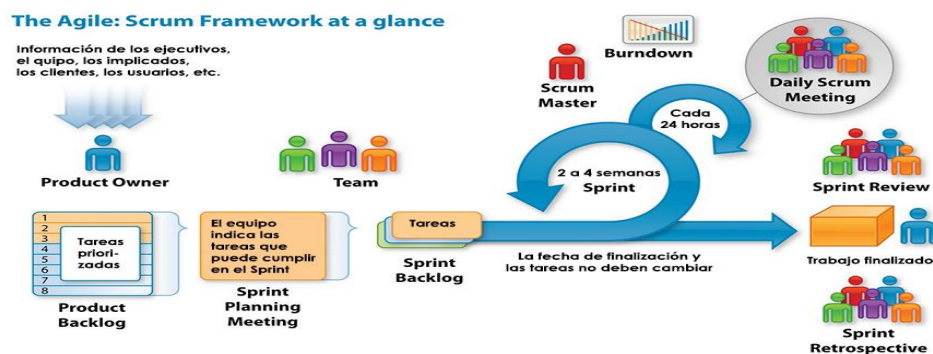


Figura 12-1: Scrum
Fuente: Torras, 2015

1.9.3 Beneficios de la Metodología

- Flexibilidad de requerimientos cambiables.
- Productos de mejor calidad.
- Reducción del tiempo de comercialización.
- Mejora de la satisfacción de los interesados.
- Autodisciplina y autogestión del equipo.

- Reducción de riesgos.

Scrum se puede dividir de forma general en 3 fases, que podemos entender como reuniones. Las reuniones forman parte de los artefactos de esta metodología junto con los roles y los elementos que lo forman (Tigras 2013, p 35).

1.9.4 Fases de Scrum

Scrum se compone de las siguientes fases.

Pre-juego: En esta fase se especifica la planificación y arquitectura del sistema.

Juego: En esta fase se realiza el desarrollo de los sprint planificados.

Post-Juego: En esta fase se realiza las pruebas y la documentación previa a su implementación.

1.9.5 Componentes de Scrum

1.9.5.1 Roles

Todas las personas que intervienen directa o indirectamente en la construcción del proyecto scrum los clasifica en dos grupos comprometidos e implicados (Tigras 2013, p 36). Ver la **Tabla 2-1**.

Tabla 2-1: Roles de Scrum

| Comprometidos(cerdos) | |
|------------------------------|---|
| Product Owner | Es la persona que toma las decisiones, y es la que realmente conoce el negocio del cliente y su visión del producto. Se encarga de escribir las ideas del cliente, las ordena por prioridad y las coloca en el Product Backlog. |
| ScrumMaster | Es el encargado de comprobar que el modelo y la metodología funcionen. Eliminará todos los inconvenientes que hagan que el proceso no fluya e interactuará con el cliente y con los gestores. |
| Equipo | Suele ser un equipo pequeño de unas 5-9 personas y tienen autoridad para organizar y tomar decisiones para conseguir su objetivo. Realizan el trabajo. |
| Implicados(gallinas) | |
| Usuarios | Es el destinatario final del producto. |
| Stakeholders | Las personas a las que el proyecto les producirá un beneficio. Participan durante las revisiones del Sprint. Clientes, Proveedores |
| Managers | Toma las decisiones finales participando en la selección de los objetivos y de los requisitos. |

Fuente: Tigras 2013.

1.9.6 Artefactos

1.9.6.1 Product Backlog

Product Backlog o pila del producto es una lista priorizada de requerimientos con tiempos estimados para convertirlos en la funcionalidad del producto terminado. Esta lista va evolucionando durante el desarrollo.

1.9.6.2 Sprint Backlog

Sprint Backlog o pila del sprint es una lista de tareas que define el equipo de trabajo para cada sprint las cuales serán entregadas tras su finalización, cada tarea detalla la persona responsable, tiempo, costo.

1.9.6.3 Sprint

Es una iteración o un ciclo de repetición de un trabajo similar que produce un incremento del producto o sistema, la duración es de 2 a 4 semanas.

1.9.6.4 Incremento

Es la funcionalidad del producto que es desarrollado por el equipo durante cada Sprint que es potencialmente productivo o de utilidad para los interesados del propietario del producto.

1.9.7 Eventos

1.9.7.1 Reunión de Planificación del sprint

Reunión de trabajo previa al inicio de cada sprint en la que se determina cuál va a ser el objetivo del sprint y las tareas necesarias para conseguirlo.

1.9.7.2 Scrum diario

Breve reunión diaria del equipo en la que cada miembro responde a tres cuestiones.

1. ¿Qué trabajo se realizó desde la reunión anterior?
2. ¿Qué trabajo se hará hasta una nueva reunión?
3. Inconvenientes que han surgido y qué hay que solucionar para poder continuar.

1.9.7.3 Revisión del sprint

Análisis e inspección del incremento generado, y adaptación de la pila del producto si resulta necesario.

1.9.7.4 Retrospectiva del sprint

Revisión de lo sucedido durante el Sprint. Reunión en la que el equipo analiza aspectos operativos de la forma de trabajo y crea un plan de mejoras para aplicar en el próximo sprint (Palacio 2014, p 29).

CAPÍTULO II

2 DESARROLLO DEL SISTEMA WEB DE GESTIÓN ODONTOLÓGICO UTILIZANDO SCRUM

En este apartado se detalla el desarrollo e implementación de la aplicación web de Gestión Odontológico, en base a las herramientas conceptualizadas en el capítulo II y haciendo uso de la metodología ágil Scrum, la misma que se compone de las siguientes fases Pre-juego, Juego, Post-juego con el fin de generar entregables en tiempos cortos, y atender a los requerimientos inestables sin interrumpir en el desarrollo del proyecto.

2.1 Fase de Pre-Juego

2.1.1 Antecedentes

La Empresa Eléctrica Riobamba S.A. es la encargada de suministrar el servicio público de energía eléctrica, su estructura interna está formada por varios departamentos, los mismos que brindan servicios tanto a sus empleados como al público en general. Uno de estos es el departamento médico Odontológico que se encarga de brindar salud dental a todos sus empleados.

Este departamento gestiona su información de manera semiautomática, mediante hojas de Excel, hojas realizadas en Word que después son llenadas a mano, provocando una serie de inconvenientes como integridad de datos, la pérdida de tiempo, consumo de recursos, y dificultando la generación de reportes.

El departamento odontológico requiere una aplicación de software que gestione la información y que resuelva estos inconvenientes.

2.1.2 Alternativa de solución

Se pretende desarrollar un sistema web que ayude en la gestión de información en base a requerimientos específicos del departamento odontológico, haciendo uso de herramientas de software libre.

2.1.3 Beneficios de la aplicación

A continuación se listan los beneficios que ofrecerá el sistema web.

- Integridad de datos.
- Información actualizada.
- Ahorro de recursos.
- Disminución de tiempo en los reportes.
- Disponibilidad de la información.
- Facilidad y rapidez para gestionar grandes volúmenes de información.

2.1.4 Factibilidad

Se realizó el estudio de factibilidad técnica, económica, operativa con el fin de determinar si el sistema es viable o no.

2.1.4.1 Factibilidad Técnica

La Empresa Eléctrica Riobamba S.A. cuenta con el hardware requerido para el funcionamiento de la aplicación, y en cuanto al software se utilizará herramientas de software libres, por lo tanto se puede decir que el proyecto es técnicamente factible. Para más información podemos ver el ANEXO A factibilidad técnica.

2.1.4.2 Factibilidad Económica

El costo de la aplicación es de \$ 3.052 el mismo que no será financiado por la empresa, ya que este proyecto de tesis es un requisito para la obtención del título como Ingeniero en Sistemas Informáticos, la empresa brindará todas las facilidades que se necesite en el transcurso del desarrollo del proyecto para que este pueda ser implementado satisfactoriamente. Por tal motivo el desarrollo del sistema es factible económicamente, para más detalles de los costos calculados ver el ANEXO A factibilidad económica.

2.1.4.3 Factibilidad Operativa

La aplicación web está desarrollada y diseñada en base a requerimientos específicos del departamento, siendo una herramienta de fácil uso para los técnicos como para los usuarios finales. La falta de mejorar la gestión de información en el departamento y agilizar la generación de reportes lleva a su aceptación, por lo que es factible operativamente. A continuación en la **Tabla 1-2** se detallan la los usuarios directos para el sistema a implementarse, se realizará una capacitación para su respectivo uso.

Tabla 1-2: Usuarios Directos

| USUARIO | CAPACITACIÓN |
|----------------------|--|
| Administrador | Funcionamiento general del sistema Base de Datos, código, arquitectura, diseño de interfaces. |
| Dentista | Administración de las funciones como gestión de citas, gestión de odontograma, gestión de pacientes etc. |
| Asistente | Administración de funciones en base a su rol de usuario. |
| Paciente | Administración de citas, ver su diagnóstico, ver su tratamiento. |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

2.1.5 Análisis de riesgos

Al concluir con el análisis de riesgos se puede determinar que el riesgo “*la mala especificación de requisitos por parte del cliente*” tiene una probabilidad alta, por tal motivo se propone la solución de usar la metodología ágil Scrum la cual permite que el cliente interactúe más con el equipo de desarrollo. Se puede concluir que existe una probabilidad baja para que los demás riesgos analizados se conviertan en inconvenientes en el desarrollo de este proyecto. Para más detalles del análisis de riesgos podemos ver el ANEXCO B.

2.1.6 Características de la aplicación







Se lista las características importantes de la aplicación a desarrollarse.

- Es una aplicación web compatible con varios navegadores.
- Multiplataforma.
- Desarrollada bajo herramientas de software libre Spring, Hibernate, Primefaces.
- Resuelve los requerimientos específicos del departamento.
- Diseño en capas persistencia, lógica del negocio, presentación.
- Fácil de usar su interfaz es amigable.
- Fácil de modificar ante cambios.

2.1.7 Herramientas para el desarrollo

Para el desarrollo de la aplicación web se hizo uso de las siguientes herramientas ver la **Tabla 2-2**.

Tabla 2-2: Herramientas para el desarrollo

| Nombre | Versión | Logotipo | Operaciones |
|------------|---------|---|--|
| Netbeans | 8.1 |  | IDE para el desarrollo Instalación |
| Spring | 4.0.1 |  | Framework para la lógica del negocio Instalación Configuración |
| Hibernate | 4.3 |  | Framework de persistencia Instalación Configuración |
| Primefaces | 5 |  | Framework de presentación Instalación Configuración |
| MySQL | 5.5.8 |  | Motor de base de datos Gestión de base de datos |
| GlassFish | 4 |  | Servidor de aplicaciones |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Estas herramientas las podemos conseguir de manera gratuita de sus páginas oficiales, primero tenemos que instalar el JDK para poder crear lenguajes en plataforma JAVA, una vez instalado nuestro kit de desarrollo tenemos que instalar el IDE de desarrollo en este caso el Netbeans, como este IDE tiene soporte para algunos frameworks incluyendo Spring, Primefaces, Hibernate podemos ver la **Figura 1-2**, la creación de un proyecto en Netbeans incluyendo estas herramientas para generar nuestro proyecto, pero si queremos trabajar con versiones específicas podemos ingresar a las paginas oficiales de cada tecnología y posteriormente agregarle al proyecto.

En la siguiente pantalla (**Figura 1-2**) seleccionamos los frameworks con los que vamos a trabajar primero dar soporte para spring framework, segundo soporte para Java Server Faces de la misma manera hacemos un checklist y el la pestaña de componentes y seleccionamos el componente de Primefaces, y por ultimo checklist en hibernate aquí configuramos nuestra database connection y listo.

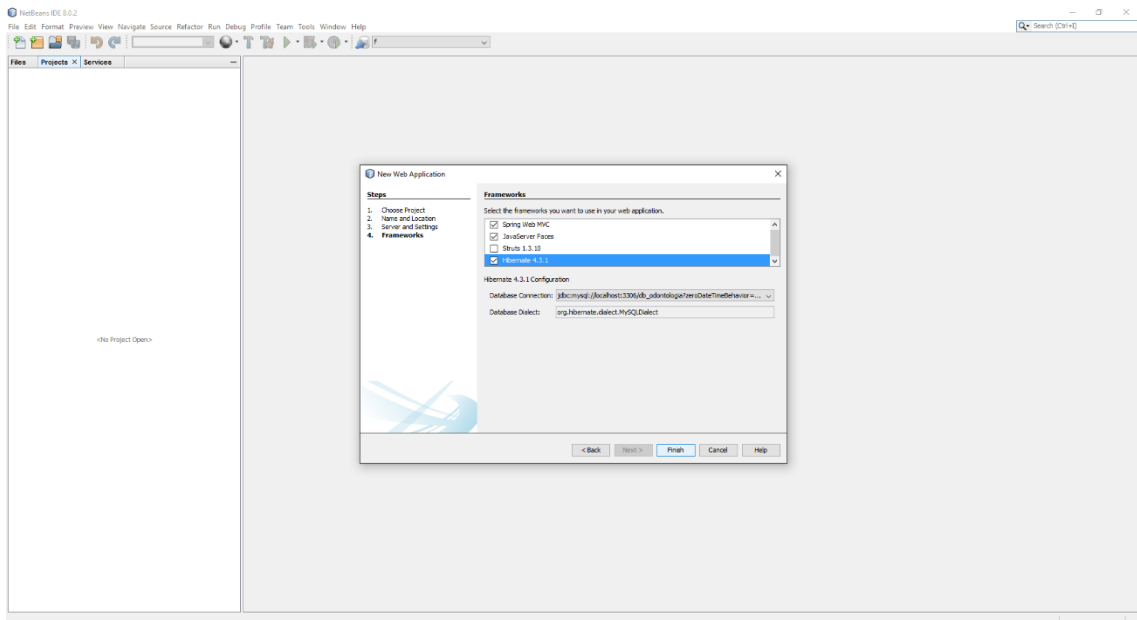


Figura 1-2: Pantalla Configuración de Frameworks

Realizado por: Jesús Mesías Ponluiza Horta. 2016

2.1.8 Roles

Para el desarrollo del sistema web se formó los siguientes roles: ver **Tabla 3-2**.

Tabla 3-2: Roles para el desarrollo del proyecto

| Persona | Rol | Descripción |
|---------------------|---------------|------------------------|
| Dr. Julio Santillán | Scrum Master | Director de tesis |
| Od. Rosa Bonifaz | Product Owner | Odontóloga de la EERSA |
| Jesús Ponluiza | Team | Tesista |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

2.1.9 Product backlog

En la **Tabla 4-2** se detalla la lista de requerimientos planteados desde la primera reunión que se realizó en el departamento odontológico, con la presencia del Ing. Darwin Morejón, ingeniero de software del departamento de tecnologías de la información de la EERSA. Los requerimientos o historias de usuario están priorizadas con una escala de (1 a 10) de acuerdo a la importancia del Product Owner, además se utilizan las abreviaturas HUT para las historias de usuario técnicas y HU para las historias de usuario. Es importante mencionar que para la recolección de las funciones requeridas se utilizó la observación así como también entrevistas personales a la odontóloga encargada del departamento y de igual manera se revisó los reportes que realizaban. De la misma manera se listan los requerimientos implícitos que no son expuestos por el cliente pero que son indispensables en el desarrollo del software.

Tabla 4-2: Product Backlog

| PRODUCT BACKLOG | | | |
|------------------------|--|-------------------------|-------------------------|
| ID | Historia de Usuario | Prioridad (0-10) | Estimación Horas |
| HUT1 | Como administrador de software necesito conocer todos los requerimientos con la finalidad de ver la funcionalidad del sistema. | 10 | 24 |
| HUT2 | Como administrador de software necesito conocer la arquitectura del sistema con la finalidad de conocer su funcionamiento. | 10 | 10 |
| HUT3 | Como administrador de software necesito conocer el diseño de la base de datos con la finalidad de análisis. | 10 | 30 |
| HUT4 | Como administrador de software necesito conocer el estándar de codificación con la finalidad de comprender. | 10 | 15 |
| HUT5 | Como administrador de software necesito conocer como está estructurado las carpetas y sus respectivas clase con la finalidad de comprender el sistema. | 10 | 8 |
| HUT6 | Como administrador de software necesito conocer el diseño de las interfaces de usuario. | 10 | 30 |
| HU7 | Como dentista necesito ingresar, editar, eliminar los datos de los pacientes. | 9 | 15 |
| HU8 | Como dentista necesito ingresar, editar, eliminar las citas médicas. | 8 | 10 |
| HU9 | Como dentista necesito realizar el diagnostico dental a cada paciente. | 9 | 80 |
| HU10 | Como dentista necesito consultar el diagnostico dental de cada paciente. | 7 | 10 |
| HU11 | Como dentista necesito editar el diagnostico dental de cada paciente. | 7 | 20 |
| HU12 | Como dentista necesito ingresar el tratamiento dental que el paciente se realizó. | 9 | 80 |
| HU13 | Como dentista necesito editar el tratamiento dental del paciente. | 9 | 10 |
| HU14 | Como dentista necesito consultar los tratamientos que se ha realizado el paciente. | 7 | 20 |
| HU15 | Como dentista necesito ingresar las patologías | 6 | 10 |
| HU16 | Como dentista necesito ingresar los tratamientos que se brindan en el departamento. | 6 | 10 |
| HU17 | Como dentista necesito tener una lista de los pacientes atendidos. | 6 | 15 |
| HU18 | Como dentista necesito tener una lista de las citas médicas pendientes. | 6 | 15 |
| HU19 | Como dentista necesito conocer los pacientes que ya han finalizado el contrato. | 7 | 10 |
| HU20 | Como dentista necesito conocer los pacientes que se atienden por primera vez en el departamento. | 7 | 10 |
| HU21 | Como administrador necesito gestionar los roles de usuario en el sistema. | 8 | 60 |
| HU22 | Como administrador necesito gestionar el acceso al sistema. | 8 | 50 |
| HU23 | Como paciente necesito ingresar la cita médica. | 6 | 10 |
| HU24 | Como paciente necesito modificar la cita médica. | 6 | 5 |
| HU25 | Como paciente necesito cancelar la cita médica. | 6 | 5 |
| HU26 | Como paciente necesito ver los tratamientos que me aplicaron. | 6 | 10 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

El proyecto cuenta con 26 historias de usuario, y una estimación de 572 horas para concluir en su totalidad con el desarrollo del proyecto.

2.1.10 Planificación de Sprint

El proyecto está dividido en 4 sprint o iteraciones los mismos que serán desarrollados en días laborables de lunes a viernes de 9:00 a 18:00 para más detalles ver la **Tabla 5-2**.

Tabla 5-2: Programación del Sprint Backlog

| SPRINT BACKLOG | | | | | |
|----------------|--|----------------------------|----------------|--------------|------------|
| ID | Historia de Usuario | S p r i n t | Responsable | Fecha Inicio | Fecha Fin |
| HUT1 | Como administrador de software necesito conocer todos los requerimientos con la finalidad de ver la funcionalidad del sistema. | 0 | | 15-10-2015 | 05-11-2015 |
| HUT2 | Como administrador de software necesito conocer la arquitectura del sistema con la finalidad de conocer su funcionamiento. | | | | |
| HUT3 | Como administrador de software necesito conocer el diseño de la base de datos con la finalidad de análisis. | | | | |
| HUT4 | Como administrador de software necesito conocer el estándar de codificación con la finalidad de comprender. | | | | |
| HUT5 | Como administrador de software necesito conocer como está estructurado las carpetas y sus respectivas clase con la finalidad de comprender el sistema. | | | | |
| HUT6 | Como administrador de software necesito conocer el diseño de las interfaces de usuario. | | | | |
| HU7 | Como dentista necesito ingresar, editar, eliminar los datos de los pacientes. | 1 | Jesús Ponluiza | 11-01-2016 | 11-02-2016 |
| HU8 | Como dentista necesito ingresar, editar, eliminar las citas médicas. | | | | |
| HU9 | Como dentista necesito realizar el diagnostico dental a cada paciente. | | | | |
| HU10 | Como dentista necesito consultar el diagnostico dental de cada paciente. | | | | |
| HU11 | Como dentista necesito editar el diagnostico dental de cada paciente. | | | | |
| HU12 | Como dentista necesito ingresar el tratamiento dental que el paciente se realizó. | | | | |
| HU13 | Como dentista necesito editar el tratamiento dental del paciente. | 2 | | | |
| HU14 | Como dentista necesito consultar los tratamientos que se ha realizado el paciente. | | | | |
| HU15 | Como dentista necesito ingresar las patologías | | | | |
| HU16 | Como dentista necesito ingresar los tratamientos que se brindan en el departamento. | | | | |
| HU17 | Como dentista necesito tener una lista de los pacientes atendidos. | | | | |
| HU18 | Como dentista necesito tener una lista de las citas médicas pendientes. | | | | |

| | | | | | |
|-------------|--|----------------------------|---|------------|------------|
| HU19 | Como dentista necesito conocer los pacientes que ya han finalizado el contrato. | | | 06-11-2015 | 26-11-2015 |
| HU20 | Como dentista necesito conocer los pacientes que se atienden por primera vez en el departamento. | | | | |
| HU21 | Como administrador necesito gestionar los roles de usuario en el sistema. | s p r i n t | 3 | 27-11-2015 | 22-12-2015 |
| HU22 | Como administrador necesito gestionar el acceso al sistema. | | | | |
| HU23 | Como paciente necesito ingresar la cita médica. | | | | |
| HU24 | Como paciente necesito modificar la cita médica. | | | | |
| HU25 | Como paciente necesito cancelar la cita médica. | | | | |
| HU26 | Como paciente necesito ver los tratamientos que me aplicaron. | | | | |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

A continuación se visualiza de forma gráfica las iteraciones planificadas para el desarrollo de la aplicación **Figura 2-2**.



Figura 2-2: Duración de Sprints

Realizado por: Jesús Mesías Ponluiza Horta. 2016

2.2 Fase de Juego (Desarrollo)

En esta fase se realiza el desarrollo de cada sprint que fueron planificadas y priorizadas en la fase anterior. En esta sección se detallará el sprint 0 el cual contiene historias técnicas como la especificación de requerimientos, arquitectura del sistema, diseño de la base de datos, el estándar de codificación, la arquitectura de las carpetas del sistema. De la misma manera se especificará una iteración del sprint 1, con el fin de exponer algunas de las características de la aplicación web y ejemplificar como se desarrollarán los demás sprint. El resto de sprint y tareas de ingeniería se encuentran detallados en el ANEXO D.

2.2.1 Sprint 0

El sprint 0, engloba a las tareas iniciales, aquí se detalla las historias técnicas para tener una mejor visión de la aplicación.

2.2.1.1 Requerimientos del sistema

La primera historia técnica desarrollada es la definición de requerimientos podemos ver la **Tabla 6-2**.

Tabla 6-2: Historia Técnica 1 del sprint 0

| HISTORIA TECNICA | |
|---|---|
| Número: HUT1 | Usuario: Administrador de Software |
| Nombre de la Historia: análisis de requerimientos | |
| Prioridad en el Negocio: Alta (Alta / Media / Baja) | Spring asignado: 0 |
| Riesgos en el Desarrollo: Alto (Alto / Media / Bajo) | Puntos Estimados: 10 |
| Descripción: Como administrador de software necesito conocer todos los requerimientos con la finalidad de ver la funcionalidad del sistema. | |
| Observaciones: se debe realizar reuniones con los clientes que harán uso del sistema para poder definir de manera más objetiva los requisitos. | |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

De la misma manera se realiza las tareas de ingeniería para cada historia de usuario, las mismas que deben cumplirse en las fechas planificadas. La tarea consta del número de tarea, número de historia a la que pertenece, nombre de la tarea, tipo de tarea, los puntos estimados, fecha de inicio y fin, programador responsable, y una descripción. Ver la **Tabla 7-2**.

Tabla 7-2: Tarea de Ingeniería 1 de la historia técnica 1

| TAREA DE INGENIERÍA | |
|---|---------------------------------|
| Número de Tarea: 1 | Número Historia: HUT1 |
| Nombre de la Tarea: reunión con el personal involucrado (clientes). | |
| Tipo de Tarea: Actividad inicial (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 4 |
| Fecha de Inicio: 15-10-2015 | Fecha de Fin: 05-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: recolección de requerimientos | |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Se listan los requerimientos funcionales específicos y necesarios para el funcionamiento del sistema de gestión odontológico. Se antepone la palabra Req (requerimiento) posteriormente la numeración.

- Req-01:** Análisis de requerimientos.
- Req-02:** Diseño de la arquitectura del sistema web.
- Req-03:** Análisis y diseño de la base de datos.
- Req-04:** Definir el estándar de codificación.
- Req-05:** Definir la estructura de las carpetas del proyecto.
- Req-06:** Diseño de Interfaz de Usuario.
- Req-07:** El sistema permitirá la gestión (CRUD) de los datos del paciente.
- Req-08:** El sistema permitirá la gestión (CRUD) de las citas médicas.
- Req-09:** El sistema permitirá la gestión (CRUD) del diagnóstico dental por paciente.
- Req-10:** El sistema permitirá consultar el diagnóstico de cada paciente.
- Req-11:** El sistema permitirá la gestión (CRUD) de los tratamientos que se realiza a cada paciente.
- Req-12:** El sistema permitirá consultar los tratamientos que se realiza cada paciente.
- Req-13:** El sistema permitirá la gestión (CRUD) de patologías.
- Req-14:** El sistema permitirá la gestión (CRUD) de tratamientos que se brindan en el departamento.
- Req-15:** El sistema permitirá tener una lista de los pacientes atendido de acuerdo a algún criterio.
- Req-16:** El sistema permitirá tener una lista de las citas médicas pendientes.
- Req-17:** El sistema permitirá conocer los pacientes cuyo contrato ha caducado.
- Req-18:** El sistema permitirá tener una lista de los pacientes que se atienden por primera vez.
- Req-19:** El sistema permitirá gestionar los roles para el respectivo acceso al sistema.

De la misma manera se lista los requerimientos no funcionales de la aplicación web.

- **Rendimiento**, cumple con la expectativa de los usuarios.
- **Interfaces**, amigables con el usuario.
- **Disponibilidad**, el sistema funcionara las 24 horas del día.
- **Mantenibilidad**, fácil de entender el código ya que está desarrollado de forma modular.
- **Escalabilidad**, fácil de modificar, quitar o añadir funcionalidades ya que su arquitectura está en capas.

2.2.1.2 *Arquitectura de la Aplicación*

La segunda historia técnica desarrollada es la arquitectura de la aplicación ver la **Tabla 8-2**.

Tabla 8-2: Historia Técnica 2 del sprint 0

| HISTORIA TECNICA | |
|---|------------------------------------|
| Número: HUT2 | Usuario: Administrador de Software |
| Nombre de la Historia: arquitectura de la aplicación | |
| Prioridad en el Negocio: Alta (Alta / Media / Baja) | Spring asignado: 0 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 10 |
| Descripción: Como administrador de software necesito conocer la arquitectura del sistema con la finalidad de conocer su funcionamiento. | |
| Observaciones: tener en cuenta las características que debe poseer la aplicación. | |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

La arquitectura de la aplicación de gestión odontológica está desarrollada en 3 capas, capa de presentación, capa de lógica del negocio, capa de acceso a datos como podemos ver en la **Figura 3-2**. También se puede apreciar los frameworks utilizados en cada capa de la aplicación.

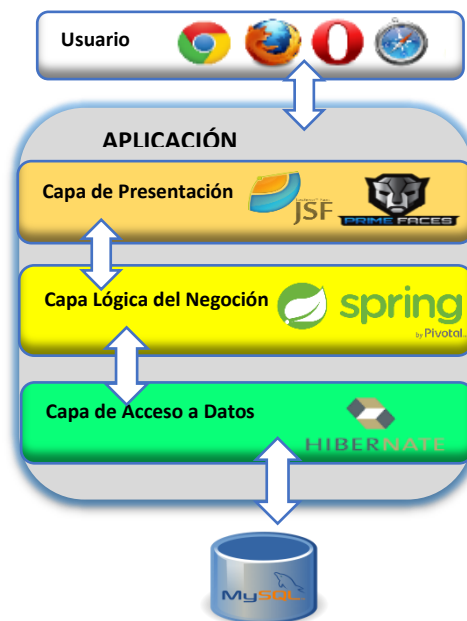


Figura 3-2: Arquitectura de la Aplicación

Realizado por: Jesús Mesías Ponluiza Horta. 2016

2.2.1.3 Diagrama de despliegue

En la **Figura 4-2** podemos ver la configuración de la estructura de los componentes de hardware (nodos) y el funcionamiento del software sobre dichos nodos.

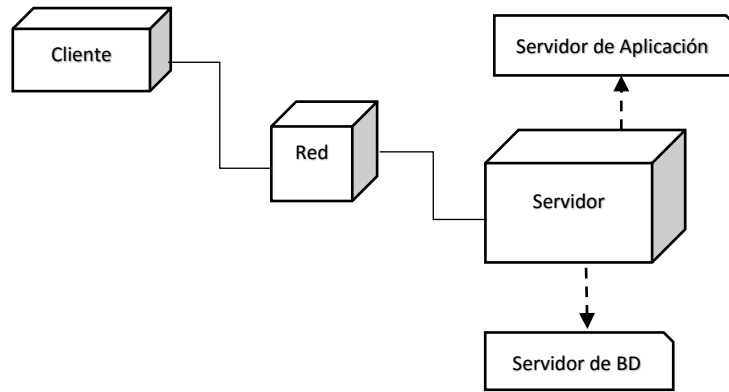


Figura 4-2: Diagrama de Despliegue
Realizado por: Jesús Mesías Ponluiza Horta. 2016

2.2.1.4 Diagramas de caso de uso

Los diagramas de caso de uso se especifican en base a los roles que tendrá la aplicación web, para comprender los procesos que realizará cada usuario posteriormente a su autenticación en la aplicación web. La aplicación tendrá 4 roles como son: Administrador, Dentista, Asistente, Paciente. El rol de asistente será una derivación de las actividades del Dentista. El rol administrador tendrá acceso a todas las funcionalidades del sistema ver la **Figura 5-2**.

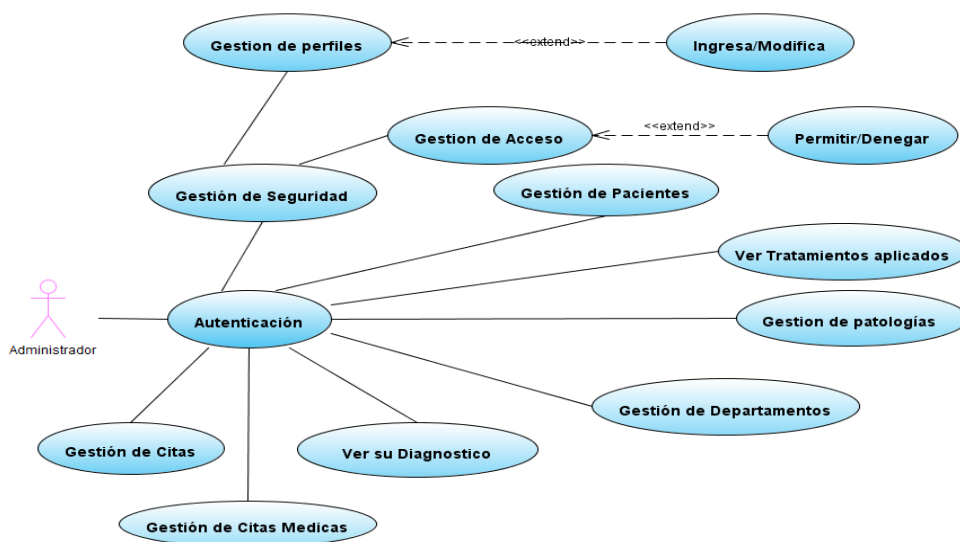


Figura 5-2: Caso de uso rol Administrador
Realizado por: Jesús Mesías Ponluiza Horta. 2016

El rol Dentista contiene funciones importantes como de realizar el diagnóstico, realizar los tratamientos, gestión de pacientes, gestión de citas médicas etcétera ver la **Figura 6-2**.

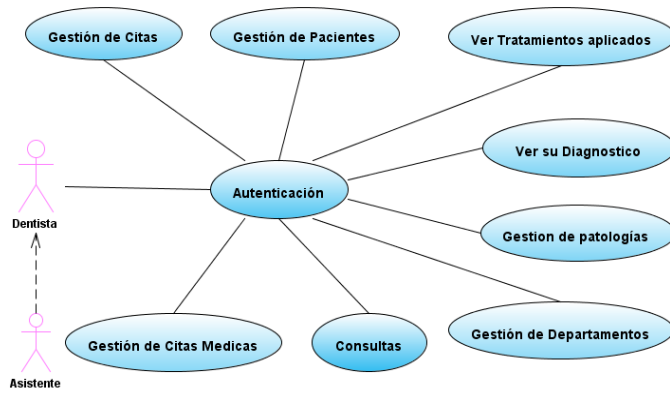


Figura 6-2: Caso de uso rol Dentista
Realizado por: Jesús Mesías Ponluiza Horta. 2016

El rol Paciente contiene funcionalidades limitadas como de realizar citas, ver su diagnóstico, ver su tratamiento **Figura 7-2**.

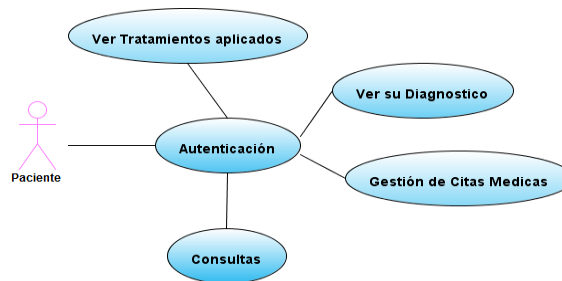


Figura 7-2: Caso de uso rol Paciente
Realizado por: Jesús Mesías Ponluiza Horta. 2016

2.2.1.5 Diseño de la Base de Datos

La tercera historia técnica desarrollada es el análisis y diseño de la base de datos ver la **Tabla 9-2**.

Tabla 9-2: Historia Técnica 3 del sprint 0

| HISTORIA TECNICA | |
|---|---|
| Número: HUT3 | Usuario: Administrador de Software |
| Nombre de la Historia: diseño de la base de datos | |
| Prioridad en el Negocio: Alta (Alta / Media / Baja) | Spring asignado: 0 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 10 |
| Descripción: Como técnico necesito conocer el diseño de la base de datos con la finalidad de análisis. | |
| Observaciones: analizar bien los requerimientos antes de diseñar la BD. | |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Después de haber analizado los requerimientos, se diseña la base de datos la misma que contiene las tablas y relaciones necesarias para las distintas funcionalidades de la aplicación. Las tablas que se utilizaron en el sistema web de gestión odontológico podemos ver en la **Figura 8-2**.

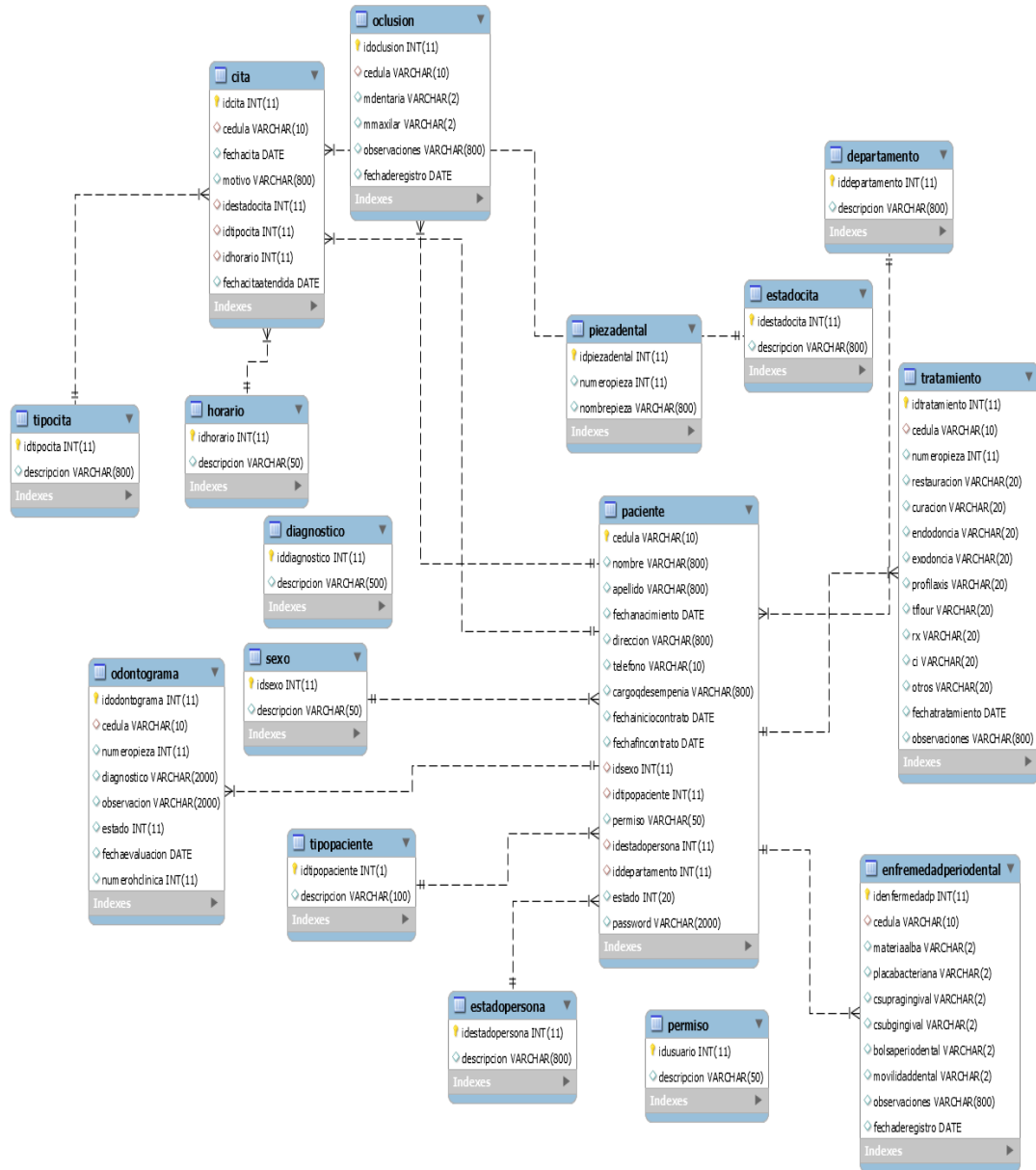


Figura 8-2: Base de Datos
Realizado por: Jesús Mesías Ponluiza Horta. 2016

2.2.1.6 Diccionario de Datos

Después de realizar el diagrama lógico de la base de datos, se genera el diccionario de datos el cual contiene las características lógicas de los datos como Tipo, Tamaño, si el campo acepta datos nulos o no, como también su clave primaria y su clave foránea. A continuación en la **Tabla 10-2**

se especifica el diccionario de datos de la tabla Paciente, el diccionario completo de la base de datos se especifica en el ANEXO C.

Tabla 10-2: Diccionario de datos tabla paciente

| Campo | Tipo/tamaño | Nulo | Enlaces a | Pk | FK |
|---------------------|--------------------|-------------|----------------------------------|-----------|-----------|
| Cedula | varchar(10) | No | | Si | |
| nombre | varchar(800) | Sí | | | |
| apellido | varchar(800) | Sí | | | |
| fechanacimiento | Date | Sí | | | |
| direccion | varchar(800) | Sí | | | |
| telefono | varchar(10) | Sí | | | |
| cargodesempenia | varchar(800) | Sí | | | |
| Fechainiciocontrato | date | Sí | | | |
| fechafincontrato | date | Sí | | | |
| Idsexo | int(11) | Sí | sexo -> idsexo | | Si |
| idtipopaciente | int(11) | Sí | tipopaciente -> idtipopaciente | | Si |
| permiso | varchar(50) | Sí | | | |
| idestadopersona | int(11) | Sí | estadopersona -> idestadopersona | | Si |
| iddepartamento | int(11) | Sí | departamento -> iddepartamento | | Si |
| Estado | int(20) | Sí | | | |
| password | varchar(2000) | Sí | | | |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

2.2.1.7 Estructura de paquetes

La quinta historia técnica desarrollada es la estructura de paquetes de la aplicación web ver la **Tabla 11-2**.

Tabla 11-2: Historia Técnica 5 del sprint 0

| HISTORIA TECNICA | |
|--|---|
| Número: HUT5 | Usuario: Administrador de Software |
| Nombre de la Historia: estructura de paquetes | |
| Prioridad en el Negocio: Media (Alta / Media / Baja) | Spring asignado: 0 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 10 |
| Descripción: Como administrador de software necesito conocer como está estructurado las carpetas y sus respectivas clase con la finalidad de comprender el sistema. | |
| Observaciones: crear las carpetas necesarias para organizar mejor. | |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

La estructura de la aplicación desarrollada está organizada en paquetes, como podemos apreciar en la **Figura 9-2**.

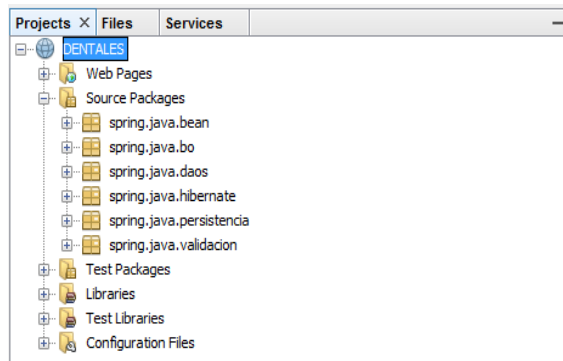


Figura 9-2: Estructura de paquetes de la aplicación
Realizado por: Jesús Mesías Ponluiza Horta. 2016

A continuación se detalla el contenido de cada uno de los paquetes que contiene la aplicación web.

Web Pages: En este paquete se encuentran todas las interfaces de usuario, y los recursos utilizados como las imágenes, hojas de estilo.

Spring.java.bean: en este paquete se encuentra las clases que serán los controladores.

Spring.java.daos: en este paquete se encuentran todas las interfaces con sus funciones definidas y todas las implementaciones de dichas interfaces, una interfaz por cada tabla de la base de datos mapeada.

Spring.java.bo: este paquete business objects contiene las interfaces como sus implementaciones de dichas interfaces de todas las clases que se definieron e el paquete DAO.

Spring.java.hibernate: este paquete contiene los archivos de configuración de Hibernate.

Spring.java.persistencia: en este paquete encontramos todas las clases .java como también los archivos hbm.xml de cada clase, que Hibernate genera cuando se realiza el mapeo de la base de datos.

Spring.java.validacion: en este paquete se encuentran las clases para realizar validaciones al sistema.

Libraries: en este paquete se encuentran las librerías necesarias para la construcción de la aplicación web.

Configuration Files: en este paquete se encuentran los archivos de configuración de Spring Framework, Primefaces y el archivo de despliegue de la aplicación el fichero web.

2.2.1.8 Codificación del sistema

La cuarta historia técnica desarrollada es el estándar de codificación ver la **Tabla 12-2**.

Tabla 12-2: Historia Técnica 4 del sprint 0

| HISTORIA TECNICA | |
|---|---|
| Número: HUT4 | Usuario: Administrador de Software |
| Nombre de la Historia: codificación del sistema | |
| Prioridad en el Negocio: Media (Alta / Media / Baja) | Spring asignado: 0 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 10 |
| Descripción: Como administrador de software necesito conocer el estándar de codificación con la finalidad de comprender. | |
| Observaciones: documentar el código (lo importante). | |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

La aplicación está estructurada en capas y codificada mediante el lenguaje de programación Java, con el apoyo de los frameworks Spring, Hibernate y Primefaces los mismos que nos proveen un estándar de codificación en base a una estructura definida como el uso de las funciones predefinidas que nos proporcionan cada Framework. A continuación se detalla cómo está compuesta cada capa.

Capa de persistencia

Después de haber diseñado la base de datos se procede a crear la aplicación web en Netbeans para mapear nuestros objetos (todas las entidades de la BD creada) con la herramienta de Hibernate ver la **Figura 10-2** la creación de un New Project en Netbeans.

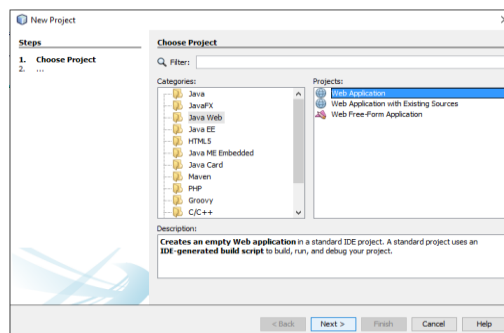


Figura 10-2: Creación de la aplicación Java web

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Después de haber puesto el nombre, seleccionado el servidor, tenemos que seleccionar los frameworks que se van a utilizar, y uno de estos es que la aplicación tenga soporte para Hibernate. A continuación podemos ver la configuración **Figura 11-2**.

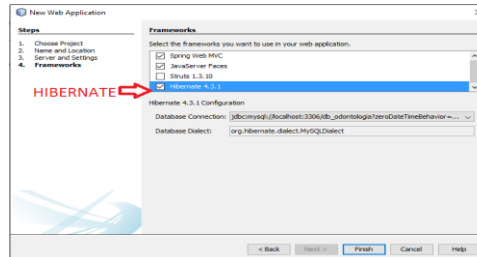


Figura 11-2: Selección de Frameworks
Realizado por: Jesús Mesías Ponluiza Horta. 2016

En la **Figura 12-2** se puede verificar todos los jar (librerías) de cada Framework que son añadidos al proyecto dando soporte para Spring, Hibernate, JSF y Primefaces así como también el conector para MySQL entre otra dependencias más, para empezar con el desarrollo de la aplicación web.

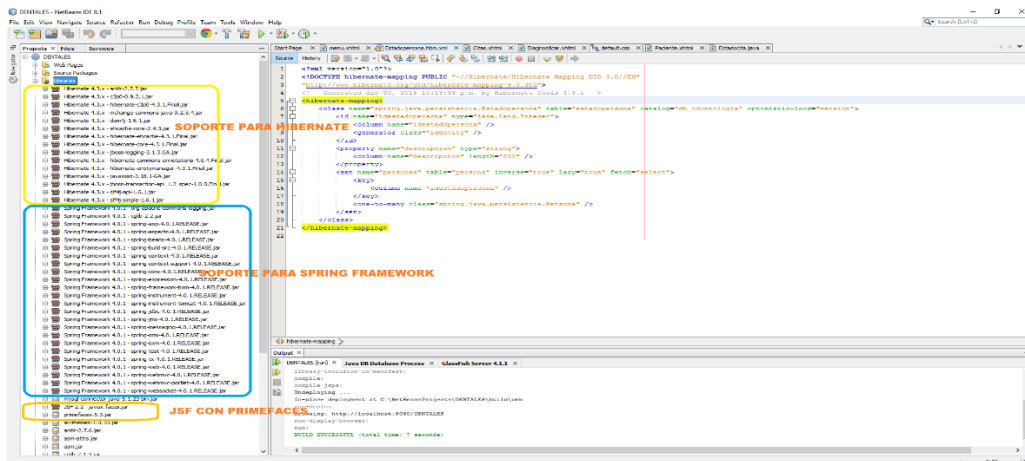


Figura 12-2: Verificación de Jar Agregados
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Posteriormente a la creación del proyecto se define un paquete en este caso *spring.java.hibernate* para guardar los archivos de configuración de Hibernate. Ver **Figura 13-2**.

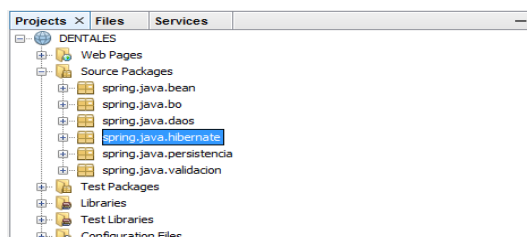


Figura 13-2: Paquete Java Hibernate
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Después creamos el archivo de configuración hibernate.cfg.xml de Hibernate para poder mapear los objetos de la base de datos. Ver **Figura 14-2**.

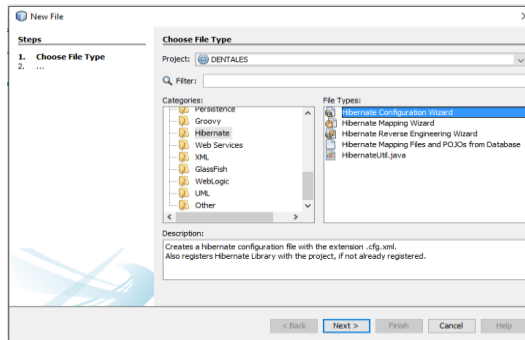


Figura 14-2: Archivo hibernate.cfg.xml
Realizado por: Jesús Mesías Ponluiza Horta. 2016

De igual manera se crea el archivo Hibernate Reverse Engineering Wizard como se puede ver en la siguiente **Figura 15-2**. En esta ventana seleccionamos todas las entidades que van a ser mapeadas.

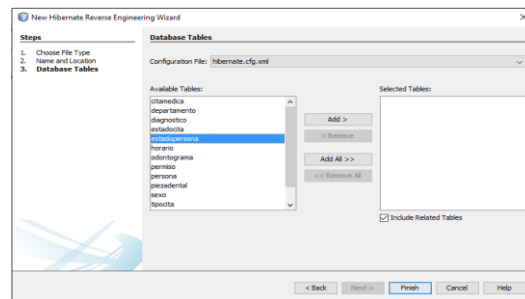


Figura 15-2: Archivo hibernate.reveng.xml
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Después de haber realizado los archivos de configuración se procede a generar los POJOS ver **Figura 16-2**, pero antes se crea un paquete llamado *spring.java.persistencia*, por cuestiones de organización de código ver la siguiente ventana.

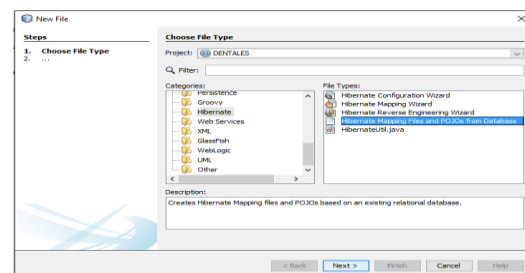


Figura 16-2: Archivo generación POJOS
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Para realizar el mapeo de los archivos y POJOs en la siguiente ventana realizamos las siguientes configuraciones con los archivos configurados anteriormente, seleccionar el paquete donde queremos generar, dar un checklist en Domain Code (.java) y en Hibernate XML Mappings (.hbm.xml). Ver **Figura 17-2**.

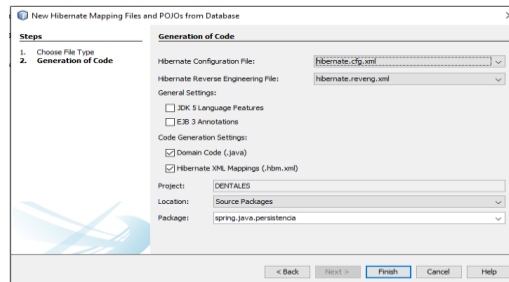


Figura 17-2: Archivo de configuración POJOs
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Una vez generado los POJOs, el paquete *spring.java.persistencia* contendrá los siguientes archivos ver la siguiente **Figura 18-2**.

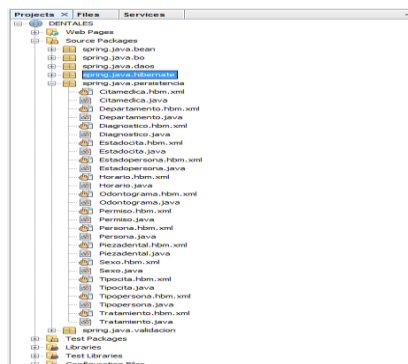
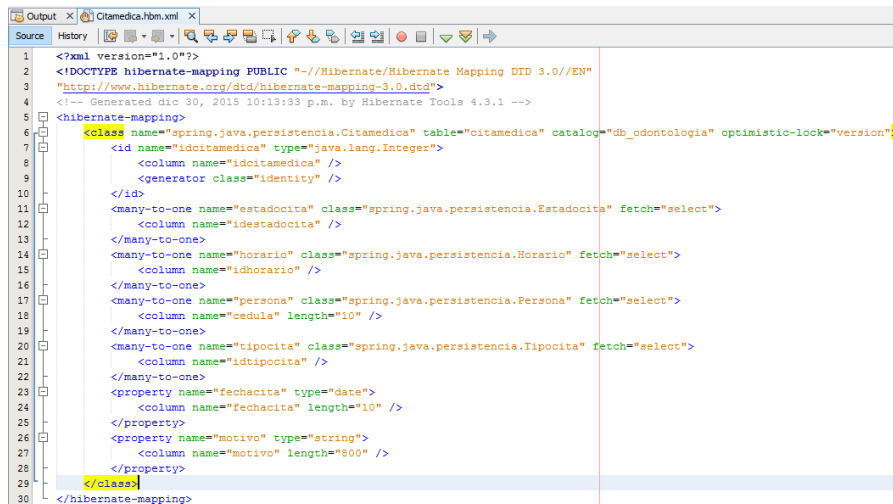


Figura 18-2: Archivos generados
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Por cada tabla de la base de datos se generan dos archivos, se muestra un ejemplo de la tabla “citamedica” donde se generan los siguientes archivos Citamedica.hbm.xml y Citamedica.java necesarios para gestionar los datos a la BD.

En el archivo Citamedica.hbm.xml podemos encontrar características de la tabla de la BD como la longitud, el tamaño de los atributos de la tabla etcétera. Ver **Figura 19-2**.



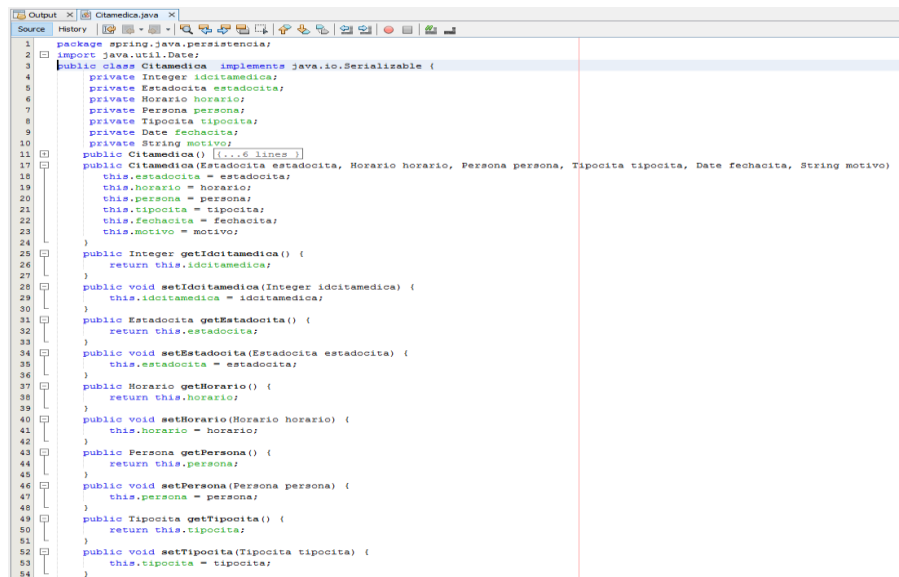
```

1 <?xml version="1.0"?>
2 <DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0/EN"
3 "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
4 <!-- Generated dic 30, 2015 10:13:33 p.m. by Hibernate Tools 4.3.1 -->
5 <hibernate-mapping>
6 <class name="spring.java.persistencia.Citamedica" table="citamedica" catalog="db_odontologia" optimistic-lock="version">
7 <id name="idcitamedica" type="java.lang.Integer">
8 <column name="idcitamedica" />
9 <generator class="identity" />
10 </id>
11 <many-to-one name="estadocita" class="spring.java.persistencia.Estadocita" fetch="select">
12 <column name="idestadocita" />
13 </many-to-one>
14 <many-to-one name="horario" class="spring.java.persistencia.Horario" fetch="select">
15 <column name="idhorario" />
16 </many-to-one>
17 <many-to-one name="persona" class="spring.java.persistencia.Persona" fetch="select">
18 <column name="cedula" length="10" />
19 </many-to-one>
20 <many-to-one name="tipocita" class="spring.java.persistencia.Tipocita" fetch="select">
21 <column name="idtipocita" />
22 </many-to-one>
23 <property name="fechacita" type="date">
24 <column name="fechacita" length="10" />
25 </property>
26 <property name="motivo" type="string">
27 <column name="motivo" length="800" />
28 </property>
29 </class>
30 </hibernate-mapping>

```

Figura 19-2: Archivo Citamedica.hbm.xml
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Y la clase Citamedica.java encontramos todos los atributos con sus respectivos set y get, constructor. Como podemos ver la siguiente **Figura 20-2**.



```

1 package spring.java.persistencia;
2 import java.util.Date;
3 public class Citamedica implements java.io.Serializable {
4     private Integer idcitamedica;
5     private Estadocita estadocita;
6     private Horario horario;
7     private Persona persona;
8     private Tipocita tipocita;
9     private Date fechacita;
10    private String motivo;
11
12    public Citamedica() {}
13    public Citamedica(Estadocita estadocita, Horario horario, Persona persona, Tipocita tipocita, Date fechacita, String motivo) {
14        this.estadocita = estadocita;
15        this.horario = horario;
16        this.persona = persona;
17        this.tipocita = tipocita;
18        this.fechacita = fechacita;
19        this.motivo = motivo;
20    }
21
22    public Integer getIdcitamedica() {
23        return this.idcitamedica;
24    }
25
26    public void setIdcitamedica(Integer idcitamedica) {
27        this.idcitamedica = idcitamedica;
28    }
29
30    public Estadocita getEstadocita() {
31        return this.estadocita;
32    }
33
34    public void setEstadocita(Estadocita estadocita) {
35        this.estadocita = estadocita;
36    }
37
38    public Horario getHorario() {
39        return this.horario;
40    }
41
42    public void setHorario(Horario horario) {
43        this.horario = horario;
44    }
45
46    public Persona getPersona() {
47        return this.persona;
48    }
49
50    public void setPersona(Persona persona) {
51        this.persona = persona;
52    }
53
54    public Tipocita getTipocita() {
55        return this.tipocita;
56    }
57
58    public void setTipocita(Tipocita tipocita) {
59        this.tipocita = tipocita;
60    }
61
62    public Date getFechacita() {
63        return this.fechacita;
64    }
65
66    public void setFechacita(Date fechacita) {
67        this.fechacita = fechacita;
68    }
69
70    public String getMotivo() {
71        return this.motivo;
72    }
73
74    public void setMotivo(String motivo) {
75        this.motivo = motivo;
76    }
77
78    }

```

Figura 20-2: Clase Citamedica.java
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Una vez generado los archivos POJOs con Hibernate, se crea las interfaces para cada una de las tablas de la base de datos. El paquete *spring.java.daos* contiene todas las interfaces con sus respectivas declaraciones de los métodos, estas interfaces sirven para que la lógica del negocio no sepa nada de Hibernate y siempre que quiera acceder a los datos lo realice a través de estas interfaces. A continuación se presenta un ejemplo de la declaración de una interface Dao, para posteriormente realizar las implementaciones necesarias. Ver la **Figura 21-2**.

```

1 package spring.java.dao;
2
3 import java.util.List;
4 import spring.java.persistencia.Citamedica;
5
6 public interface CitamedicaDao {
7
8     public void insertar(Citamedica objCitaM);
9
10    public void eliminar(Citamedica objCitaM);
11
12    public void actualizar(Citamedica objCitaM);
13
14    public List<Citamedica> listar();
15
16    public List<Citamedica> listarCitaspendientes(String cedula);
17
18    public List<Citamedica> listarCitasaatendidas(String cedula);
19
20    public List<Citamedica> listarCitasaatendidas();
21
22    public Citamedica obtenerdatos(Citamedica objCitaM);
23
24    public Object totalCitas();
25
26 }
27

```

Figura 21-2: Interfaz Citamedica
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Posterior a la definición de las interfaces se realiza la implementación (clases) de cada uno de los métodos de cada interfaz definidas en el paquete Dao, estas clases se crean en el mismo paquete. A continuación se presenta un ejemplo de una clase que implementa a una interfaz. Ver la **Figura 22-2**.

```

3 import java.util.List;
4 import org.springframework.orm.hibernate4.support.HibernateDaoSupport;
5 import spring.java.persistencia.Citamedica;
6
7 public class CitamedicaImplDao extends HibernateDaoSupport implements CitamedicaDao {
8
9     @Override
10    public void insertar(Citamedica objCitaM) {
11        getHibernateTemplate().save(objCitaM);
12    }
13
14    @Override
15    public void eliminar(Citamedica objCitaM) {
16        getHibernateTemplate().delete(objCitaM);
17    }
18
19    @Override
20    public void actualizar(Citamedica objCitaM) {
21        getHibernateTemplate().update(objCitaM);
22    }
23
24    @Override
25    public List listar() {
26        return getHibernateTemplate().find("FROM Citamedica c left join fetch c.horario left join fetch c.persona left join fetch c.tipocita left join fetch c.estadocita");
27    }
28
29    @Override
30    public Citamedica obtenerdatos(Citamedica objCitaM) {
31        return getHibernateTemplate().load(Citamedica.class, objCitaM);
32    }
33
34    @Override
35    public Object totalCitas() {
36        return getHibernateTemplate().find("select count(c) from Citamedica c where idestadocita=1");
37    }
38
39    @Override
40    public List<Citamedica> listarCitaspendientes(String cedula) {
41        List listarcitaspendientes = null;
42        String hql = "FROM Citamedica c left join fetch c.horario left join fetch c.persona as p left join fetch c.tipocita left join fetch c.estadocita as e where e.ides";
43        listarcitaspendientes = getHibernateTemplate().find(hql);
44        return listarcitaspendientes;
45    }
46
47    @Override
48    public List<Citamedica> listarCitasaatendidas(String cedula) {
49        List listarcitasatendidas = null;
50    }
51

```

Figura 22-2: Implementación de la interfaz Citamedica
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Después de implementar los Dao se procede a implementar la clase de los Bo a continuación se presenta la implementación de la clase CitamedicaImplBo ver **Figura 23-2**.

```

1 package spring.java.bo;
2 import java.util.List;
3 import org.springframework.transaction.annotation.Transactional;
4 import org.springframework.dao.CitamedicaImplDao;
5 import org.springframework.dao.CitamedicaImplDao;
6 import org.springframework.dao.CitamedicaImplDao;
7 public class CitamedicaimplBo implements CitamedicaBo {
8     private CitamedicaImplDao citamedicaDao;
9
10    @Override
11    @Transactional
12    public void insertar(Citamedica objcitam) {
13        citamedicaDao.insertar(objcitam);
14    }
15
16    @Override
17    @Transactional
18    public void eliminar(Citamedica objcitam) {
19        citamedicaDao.eliminar(objcitam);
20    }
21
22    @Override
23    @Transactional
24    public void actualizar(Citamedica objcitam) {
25        citamedicaDao.actualizar(objcitam);
26    }
27
28    @Override
29    @Transactional
30    public List listar() {
31        return citamedicaDao.listar();
32    }
33
34    @Override
35    @Transactional
36    public Citamedica obtenerdatos(Citamedica objcitam) {
37        return citamedicaDao.obtenerdatos(objcitam);
38    }
39
40    @Override
41    public Object totalcitas() {
42        return citamedicaDao.totalcitas();
43    }
44
45    @Override
46    @Transactional
47    public List<Citamedica> listarcitaspendientes(String cedula) {
48        return citamedicaDao.listarcitaspendientes(cedula);
49    }
50
51    @Override
52    @Transactional
53    public List<Citamedica> listarcitassatendidas(String cedula) {
54        return citamedicaDao.listarcitassatendidas(cedula);
55    }
56
57 }

```

Figura 23-2: Implementación de la clase CitamedicaimplBo
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Capa lógica de negocio

De igual manera de debe implementar todas las clases definidas anteriormente a continuación se muestra un ejemplo de una clase implementada ver la **Figura 24-2**.

```

1 package spring.java.bean;
2
3 import ...31 lines
4
5 @ViewScoped
6 public class CitamedicaBean implements Serializable {
7
8     //ToDo
9     private CitamedicaDao citamedicaDao;
10    private List<Citamedica> listacitamedica;
11    private List<Citamedica> listacitamedicasatendidas;
12    private List<Citamedica> listacitamedicaspendientes;
13    private List<Citamedica> listacitassatendidaspaciente;
14    private Estadocita estadocita;
15    private Citamedica citamedica;
16
17    private Integer horario;
18    private String cedula;
19    private Integer tipocita;
20    private Date fechacita;
21    private String motivo;
22
23    java.util.Date fecha = new Date();
24
25    public CitamedicaBean() {
26        citamedica = new Citamedica();
27    }
28
29    public void actualizar() {
30        try {
31            getCitamedicaDao().actualizar(citamedica);
32            listar();
33            FacesContext.getCurrentInstance().addMessage(null, new FacesMessage("Datos Modificados Correctamente"));
34        } catch (Exception e) {
35            FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(FacesMessage.SEVERITY_WARN, "Error: Al Intentar Modificar los Datos", ""));
36        }
37    }
38
39    public String insertar() { ...25 lines }
40
41 }

```

Figura 24-2: Lógica del negocio Citamedica
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Configuración de contenedor de beans SpringXMLConfig.xml

Esta es la parte más importante del desarrollo de la aplicación, ya que aquí es donde se define los objetos (beans) para realizar la inyección de dependencia. En este archivo de configuración de spring encontramos los beans como el *dataSource*, *sessionFactory*, y posteriormente todos los beans implementados de la lógica del negocio ver la **Figura 25-2**.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:aop="http://www.springframework.org/schema/aop"
5       xmlns:c="http://www.springframework.org/schema/c"
6       xmlns:context="http://www.springframework.org/schema/context"
7       xmlns:flow="http://www.springframework.org/schema/webflow-config"
8       xmlns:jee="http://www.springframework.org/schema/jee"
9       xmlns:jms="http://www.springframework.org/schema/jms"
10      xmlns:lang="http://www.springframework.org/schema/lang"
11      xmlns:oasgi="http://www.springframework.org/schema/oasgi"
12      xmlns:p="http://www.springframework.org/schema/p"
13      xmlns:tx="http://www.springframework.org/schema/tx"
14      xmlns:util="http://www.springframework.org/schema/util"
15      xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
16                        http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop.xsd
17                        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd
18                        http://www.springframework.org/schema/webflow-config http://www.springframework.org/schema/webflow-config/spring-webflow-config-2.0.xsd
19                        http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee.xsd
20                        http://www.springframework.org/schema/jms http://www.springframework.org/schema/jms/spring-jms.xsd
21                        http://www.springframework.org/schema/lang http://www.springframework.org/schema/lang/spring-lang.xsd
22                        http://www.springframework.org/schema/oasgi http://www.springframework.org/schema/oasgi/spring-oasgi-1.2.xsd
23                        http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd
24                        http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd
25
26 >
27 <context:annotation-config/>
28 <context:component-scan base-package="spring.java"/>
29
30 <bean id="transactionManager" class="org.springframework.orm.hibernate4.HibernateTransactionManager">
31   <property name="sessionFactory" ref="sessionFactory"/>
32 </bean>
33
34 <tx:annotation-driven transaction-manager="transactionManager"/>
35
36 <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
37   <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
38   <property name="url" value="jdbc:mysql://localhost:3306/db_odontologia"/>
39   <property name="username" value="root"/>
40   <property name="password" value="" />
41 </bean>
42
43 <bean id="sessionFactory" class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
44   <property name="dataSource" ref="dataSource"/>
45   <ref bean="dataSource"/>
46 </property>

```

Figura 25-2: Lógica del negocio Citamedica
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Para inyectar las dependencias, cada objeto se debe definir los Beans Dao y Bo ver la **Figura 26-2**.

```

46
47 <property name="hibernateProperties">
48   <prop>
49     <prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>
50     <prop key="hibernate.show_sql">>true</prop>
51   </prop>
52 </property>
53
54 <property name="mappingResources">
55   <list>
56     <value>/spring/java/persistencia/Citamedica.hbm.xml</value>
57     <value>/spring/java/persistencia/Departamento.hbm.xml</value>
58     <value>/spring/java/persistencia/Estadocita.hbm.xml</value>
59     <value>/spring/java/persistencia/Estadopersona.hbm.xml</value>
60     <value>/spring/java/persistencia/Odontograma.hbm.xml</value>
61     <value>/spring/java/persistencia/Persona.hbm.xml</value>
62     <value>/spring/java/persistencia/Piezadental.hbm.xml</value>
63     <value>/spring/java/persistencia/Tipocita.hbm.xml</value>
64     <value>/spring/java/persistencia/Tipopersona.hbm.xml</value>
65     <value>/spring/java/persistencia/Tratamiento.hbm.xml</value>
66     <value>/spring/java/persistencia/Sexo.hbm.xml</value>
67     <value>/spring/java/persistencia/Horario.hbm.xml</value>
68     <value>/spring/java/persistencia/Diagnostico.hbm.xml</value>
69     <value>/spring/java/persistencia/Permiso.hbm.xml</value>
70   </list>
71 </property>
72 </bean>
73
74 <!-- Tratamiento -->
75 <bean id="tratamientoDao" class="spring.java.daos.TratamientoImplDao">
76   <property name="sessionFactory" ref="sessionFactory"/>
77 </bean>
78 <bean id="tratamientoBo" class="spring.java.bo.TratamientoImplBo">
79   <property name="tratamientoDao" ref="tratamientoDao"/>
80 </bean>
81
82 <!-- Paciente -->
83 <bean id="pacienteDao" class="spring.java.daos.PacienteImplDao">
84   <property name="sessionFactory" ref="sessionFactory"/>
85 </bean>
86 <bean id="pacienteBo" class="spring.java.bo.PacienteImplBo">
87   <property name="pacienteDao" ref="pacienteDao"/>
88 </bean>
89
90 <!-- Departamento -->
91 <bean id="departamentoDao" class="spring.java.daos.DepartamentoImplDao">

```

Figura 26-2: Lógica del negocio 2 Citamedica
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Integración de spring con Primefaces

En el archivo de configuración de Primefaces el faces-config.xml se realiza la declaración de la siguiente línea de código ver la **Figura 27-2**, donde se define que el encargado de gestionar los beans será el contenedor de spring.

```

6 <application>
7   <el-resolver>org.springframework.web.jsf.el.SpringBeanFacesELResolver</el-resolver>
8 </application>

```

Figura 27-2: Integración de spring y Primefaces

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Y por último en el archivo de configuración de la aplicación denominado web.xml se escriben las siguientes líneas de código ver la **Figura 28-2**, donde se especifica en que ubicación se encuentra el archivo de configuración de spring, de la misma manera se realiza la declaración de ContextLoaderListener.

```

4 <web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema
3 <context-param>
4   <param-name>contextConfigLocation</param-name>
5   <param-value>/WEB-INF/SpringXMLConfig.xml</param-value>
6 </context-param>
7 <listener>
8   <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
9 </listener>
10 <listener>
11   <listener-class>org.springframework.web.context.request.RequestContextListener</listener-class>
12 </listener>

```

Figura 28-2: Configuración archivo web.xml

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Una vez realizadas todas estas configuraciones se puede tener una visión más clara de cómo está estructurada la aplicación, he incluso se puede notar como está el código implementado en la aplicación y como están trabajando los marcos de trabajo.

2.2.1.9 Diseño de Interfaz de Usuario

La sexta historia técnica desarrollada es el diseño de interfaces ver la **Tabla 13-2**.

Tabla 13-2: Historia Técnica 6 del sprint 0

| HISTORIA TECNICA | |
|---|---|
| Número: HUT6 | Usuario: Administrador de Software |
| Nombre de la Historia: diseño de interfaces | |
| Prioridad en el Negocio: Media (Alta / Media / Baja) | Spring asignado: 0 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 10 |
| Descripción: Como administrador de software necesito conocer el diseño de las interfaces de usuario. | |
| Observaciones: presentar las interfaces al usuario final, para ver si está de acuerdo con el diseño (colores). | |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

El diseño de las interfaces de usuario es de gran importancia en el desarrollo del software, ya que por medio de estas el usuario interactúa con la información almacenada en la base de datos. En la **Figura 29-2** se puede observar el formulario de autenticación usuario de la aplicación web, el mismo que consta de los campos usuario y contraseña, mediante los cuales se podrá tener acceso

al sistema luego de su respectiva validación de datos y posteriormente una autorización de funciones dependiendo del rol de cada usuario.

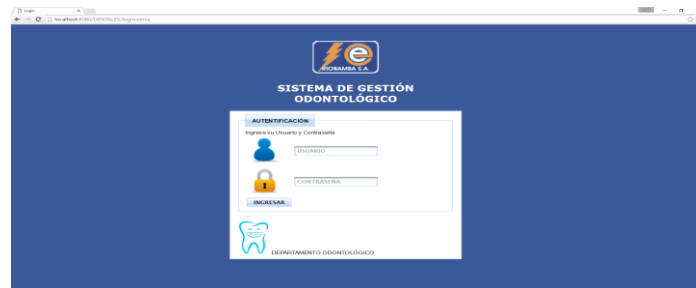


Figura 29-2: Autenticación de Usuarios
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Seguidamente se describen las características de la página principal del sistema web.

Banner: en la parte superior izquierda cuenta con el logo de la empresa, en la derecha cuenta con una imagen del departamento odontológico y en el centro esta las iniciales de la empresa EERSA y DEPARTAMENTO ODONTOLÓGICO y el fondo de todo el banner es de color celeste.

Menú: está ubicado debajo del banner, es horizontal el cual contiene varias opciones de navegación que estas a su vez tienen submenús. También consta de un menú en la parte lateral izquierda en la cual cuenta con las opciones de usuario como principal, cambiar contraseña, acerca de.

Footer: cuenta con un sello la empresa en la parte izquierda, y en la parte derecha cuenta con un logo, en el centro tenemos información de la empresa.

El tema por defecto que se utilizó es *bluesky* de Primefaces para tener un estándar de colores en las ventanas, pero se lo puede cambiar a otro si desea el usuario. Podemos ver la **Figura 30-2** como está diseñada la pantalla principal del sistema web de gestión odontológico.



Figura 30-2: Pantalla principal del sistema web
 Realizado por: Jesús Mesías Ponluiza Horta. 2016

2.2.2 Sprint 1

El Sprint 1, se desarrollaran todas las funcionalidades del rol del dentista, a continuación se especificará una historia de usuario como ejemplo del desarrollo.

2.2.2.1 Gestión de los datos del paciente

La primera historia de usuario del sprint 1 desarrollada es la realización de un CRUD de los datos del paciente podemos ver la **Tabla 14-2**.

Tabla 14-2: Historia de Usuario 1 del sprint 1

| HISTORIA DE USUARIO | |
|---|---|
| Número: HU7 | Usuario: Dentista, Administrador |
| Nombre de la Historia: El sistema permitirá realizar un crud de los datos del paciente | |
| Prioridad en el Negocio: Media (Alta / Media / Baja) | Spring asignado: 1 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 10 |
| Descripción: Como dentista necesito ingresar, editar, eliminar los datos de los pacientes. | |
| Observaciones: | |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Se presenta el formulario de ingreso de pacientes como ejemplo del diseño de las vistas, los campos que tienen (*) son obligatorios. Ver **Figura 31-2**.



Figura 31-2: Pantalla ingresar paciente
Realizado por: Jesús Mesías Ponluiza Horta. 2016

A continuación en la **Figura 32-2** se muestra la pantalla con las opciones ingresar, buscar, modificar, eliminar paciente, las cuales tendrá acceso el rol dentista y el administrador.

LISTA DE PACIENTES

INGRESAR

Datos de la Tabla

BUSCAR: Ingrese el dato a buscar

Pág: 1 de 1 Ir Pág: 1 Filas/Pág: 5

| CEDULA | NOMBRE | SEXO | TIPO DE CONTRATO | DEPARTAMENTO | CARGO | FECHA INICIO CONTRATO | FECHA FIN CONTRATO | EDAD | ESTADO | VER | MODIFICAR | ELIMINAR |
|------------|---------------------------------|-----------|------------------|----------------------|---------------------------|-----------------------|--------------------|------|---------|-----|-----------|----------|
| 0091163462 | GORDILLO VELARDE HECTOR VINICIO | MASCULINO | JUBILADO | FISCALIZACIÓN | ODONTÓLOGO | 2006-01-01 | 2016-05-29 | 59 | PA SIVO | | | |
| 0201562673 | MOREJON DARWIN | MASCULINO | EVENTUAL | INFORMATICA | ING. DE SOFTWARE | 2012-09-09 | 2016-05-28 | 37 | ACTIVO | | | |
| 1723109516 | PONLUIZA HORTA JESUS MESIAS | MASCULINO | EVENTUAL | INFORMATICA | DESARROLLADOR DE SOFTWARE | 2015-05-01 | 2016-09-22 | 27 | ACTIVO | | | |
| 0003243221 | BONIFAZ DAMIAN ROSA CEMIRA | FEMENINO | EMPLEADO EERSA | DIS. Y AL. PUBLICO | ODNTOLOGA | 2016-01-05 | 2016-05-28 | 37 | PA SIVO | | | |
| 1722431028 | ROSERO LÓPEZ MAYRA ALEXANDRA | FEMENINO | EVENTUAL | S. GENERAL Y ARCHIVO | SECRETARIA | 2017-05-02 | 2018-05-28 | 27 | ACTIVO | | | |

TOTAL DE PACIENTES: [5]

Ver todos Los Datos

Figura 32-2: Pantalla crud paciente
Realizado por: Jesús Mesías Ponluiza Horta. 2016

2.3 Fase de post-juego

En esta fase se realizó la respectiva instalación del sistema web, transfiriendo el archivo **DENTALES.war** al servidor web de la EERSA el cual se encuentra funcionando operativamente, de la misma manera se elaboraron los respectivos manuales de usuario del sistema ver ANEXO H, como también se realizó la respectiva capacitación al personal involucrado del manejo de la aplicación web. Posteriormente se realizó la entrega pertinente de la aplicación web.

Gráfico Burndown

En el siguiente gráfico podemos ver el avance de cada uno de las tareas programadas y como se fueron cumpliendo todas las iteraciones del proyecto (los sprints), de la misma manera podemos comparar los avances ideal y real del proyecto. Ver **Figura 33-2**.

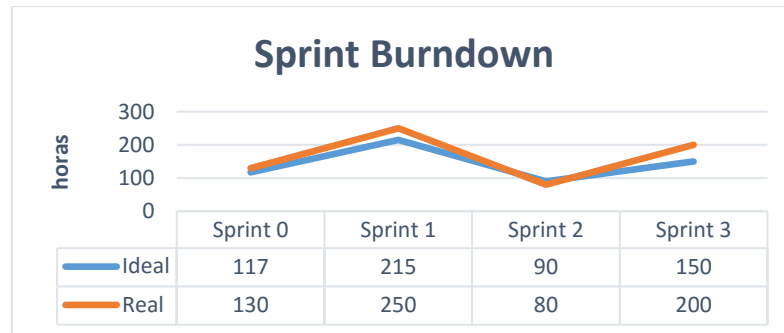


Figura 33-2: Gráfico Burndown del proyecto
Realizado por: Jesús Mesías Ponluiza Horta. 2016

CAPÍTULO III

3 ANÁLISIS Y DISCUSIÓN DE RESULTADOS

En este apartado se realiza el análisis de la eficiencia, mantenibilidad y usabilidad del sistema de gestión odontológico estableciendo indicadores en base a la situación actual del departamento, con el fin de cumplir con los objetivos planteados en este proyecto de tesis.

3.1 Definición de los parámetros

Para evaluar la situación actual del departamento, y la eficiencia del sistema de gestión odontológico como también la mantenibilidad, se definen indicadores en base a la organización internacional de normalización ISO/IEC 9126 estándar para evaluar la calidad de software. Esta norma permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software. Los modelos de calidad para el software se describen así.

Calidad interna y externa: Especifica 6 características para calidad interna y externa, las cuales, están subdivididas. Estas divisiones se manifiestan externamente cuando el software es usado como parte de un sistema Informático, y son el resultado de atributos internos de software.

Calidad en uso: Calidad en uso es el efecto combinado para el usuario final de las 6 características de la calidad interna y externa del software. Especifica 4 características para la calidad en uso (Borbón 2013, p 1).

Eficiencia es la relación entre el nivel de prestaciones de un sistema y el volumen de recursos utilizados en condiciones declaradas [ISO 9126: 1991]. Este parámetro tiene sub características.

- **Comportamiento de tiempos:** son los tiempos adecuados de respuesta ante la ejecución de un procedimiento.
- **Utilización de recursos:** es la capacidad del software para utilizar los recursos adecuados bajo requerimientos específicos.

Mantenibilidad se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad (Abud 2000, p 2).

- **Capacidad de análisis:** Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.
- **Capacidad de modificación:** Mide el esfuerzo necesario para modificar aspectos del software, remover fallas o adaptar el software para que funcione en un ambiente diferente.

En la **Tabla 1-3** se puede observar los variables e indicador establecido que se empleará en este trabajo de titulación.

Tabla 1-3: Parámetro e Indicador

| Parámetro | Indicador | Descripción | Interpretación |
|-----------------------|---------------------------|---------------------------------------|-----------------------------|
| Eficiencia | Tiempo de respuesta | Qué tan rápido responde el sistema | Entre más corto mejor. |
| Mantenibilidad | Capacidad de análisis | Qué fácil es de modificar y verificar | Mientras más rápido mejor. |
| | Capacidad de modificación | | |
| Usabilidad | Aceptación del usuario | Facilidad de uso | Cumplir con la expectativas |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Es importante mencionar que el indicador tiempo de respuesta es muy variable de acuerdo al ambiente que se está realizando las pruebas ejemplo PC utilizada, ubicación del usuario etcétera.

3.2 Criterios de evaluación

Se establece la siguiente escala de valoración para expresar los resultados analizados de una manera cuantitativa y cualitativa. La **Tabla 2-3** especifica la calificación para los parámetros.

Tabla 2-3: Criterios de evaluación

| Calificación Porcentual | Valor Cuantitativo | Valor Cualitativo |
|--------------------------------|---------------------------|--------------------------|
| > 0% y <= 20% | 1 | Malo |
| > 20% y <= 40% | 2 | Regular |
| > 40% y <= 60% | 3 | Bueno |
| > 60% y <= 80% | 4 | Muy Bueno |
| >80% y <= 100% | 5 | Excelente |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Para establecer la calificación de indicador se estable la siguiente tabla de valoración.

Tabla 3-3: Tiempos

| Tiempo | Calificación porcentual | Valor Cualitativo |
|----------------|--------------------------------|--------------------------|
| > 0% y <= 20% | 25 | Regular |
| > 25% y <= 50% | 50 | Bueno |
| > 50% y <= 75% | 75 | Muy Bueno |
| >75% y <= 100% | 100 | Excelente |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

3.3 Tiempo de respuesta

Es el tiempo empleado en completar una acción desde que se envía una solicitud de información hasta que se muestran los resultados. Para analizar este indicador es importante conocer la situación actual del departamento odontológico de cómo se estaba llevando los procesos de gestión de información antes de la implementación de la aplicación y de la misma manera después de la implementación del sistema web.

3.3.1 Análisis actual del proceso de gestión de información en el departamento odontológico

Es importante conocer la situación actual del departamento para analizar el indicador tiempo de respuesta sub-característica de la variable eficiencia.

El departamento odontológico de la EERSA maneja toda la información de sus pacientes de manera manual, ante la falta de un sistema automatizado que gestione dicha información, ha ocasionado una serie de inconvenientes los mismos que fueron descritos en el capítulo I en la parte de justificación aplicativa. Se realizó una Entrevista personal a la odontóloga encargada del departamento con el fin de conocer los tiempos aproximados en la realización de las tareas como el de registrar, modificar, listar datos, diagnóstico, tratamiento y generación de reportes, donde la Dentista proporcionó estos datos como un promedio general de los pacientes que son atendidos diariamente. En el Anexo H podemos observar el registro de los pacientes por día y en el Anexo E podemos ver las preguntas de la Entrevista las cuales se realizará el respectivo análisis.

Para el este estudio solo se especifican las tareas comúnmente utilizadas como la de gestión de pacientes, gestión de diagnóstico, gestión de tratamientos y generación de reportes, en la **Tabla 4-3** se detalla los tiempos en realizar dichas funciones.

Tabla 4-3: Tiempo de respuesta proceso manual

| Nro. | Acción | Tiempo |
|--------------|--|---------------|
| 1 | Registrar los datos del paciente. | 7 |
| 2 | Modificar los datos del paciente. | 1 |
| 3 | Buscar los datos del Paciente. | 6 |
| 4 | Registrar el tratamiento del Paciente. | 5 |
| 5 | Modificar el tratamiento del Paciente. | 5 |
| 6 | Buscar el tratamiento del Paciente. | 6 |
| 7 | Registrar el diagnóstico del Paciente. | 5 |
| 8 | Modificar el diagnóstico del Paciente. | 5 |
| 9 | Buscar el diagnóstico del Paciente. | 6 |
| 10 | Emitir reportes. | 10 |
| Total | | 56 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Para esta prueba se toman los tiempos desde que un paciente ingresa al consultorio y se lo realiza todo el proceso de atención, dando como resultado un tiempo total de **56 minutos**, este tiempo incrementaría si a la dependencia le solicitan otro tipo de reporte.

3.3.2 Tiempo de respuesta con el sistema web de gestión odontológico

Con la implementación de la aplicación se pretende reducir los tiempos empleados en la realización de una cierta tarea de gestión de información en el departamento, estos tiempos fueron cronometrados conjuntamente con la odontóloga y asistente del departamento. En la **Tabla 5-3** se especifica los tiempos obtenidos para esta prueba.

Tabla 5-3: Tiempo de respuesta proceso automatizado

| Nro. | Tarea | Tiempo |
|--------------|-------------------------------------|---------------|
| 1 | Gestión de datos del paciente | 2 |
| 2 | Gestión del tratamiento de paciente | 3 |
| 3 | Gestión de diagnóstico de paciente | 3 |
| 4 | Emitir reportes. | 2 |
| Total | | 10 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Es importante tener en cuenta que el indicador tiempo de respuesta es muy variable y los factores a tener en cuenta para esta prueba es el hardware utilizado y el tiempo en que el usuario realiza una acción en la aplicación (sin distracciones). La PC utilizada para esta prueba es una Intel i7 de 8 GB de RAM.

De la misma manera es importante explicar que todos los datos de pacientes, diagnóstico, tratamiento son validados como obligatorios, cabe mencionar que este proceso se lo realiza cuando el paciente ingresa por primera vez al consultorio, mientras que por segunda vez se omitirán pasos por ende disminuirán los tiempos.

3.3.3 Comparación de resultados

Después de a ver culminado con el análisis y obtenido los tiempos de respuesta sin la aplicación y con la implementación del sistema web. En la **Tabla 6-3** se realiza una comparación de los datos previamente obtenidos.

Tabla 6-4: Comparación de resultados

| Tiempo de Respuesta | | |
|-------------------------------------|-----------------------|-----------------------------|
| Tipo de proceso | Gestión manual | Gestión automatizada |
| Gestión de datos del paciente | 14 | 2 |
| Gestión del tratamiento de paciente | 16 | 3 |
| Gestión de diagnóstico de paciente | 16 | 3 |
| Emitir reportes | 10 | 2 |
| Total | 56 | 10 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

De acuerdo al análisis realizado para tiempo de gestión manual de información se lo realiza en un tiempo promedio de **56** minutos, el cual es un tiempo considerable para cumplir con estas tareas, mientras que con la utilización de la aplicación web se lo realiza en **10** minutos el cual es un tiempo aceptable para realizar dichas tareas. En la **Figura 1-3** se puede interpretar mejor los resultados.

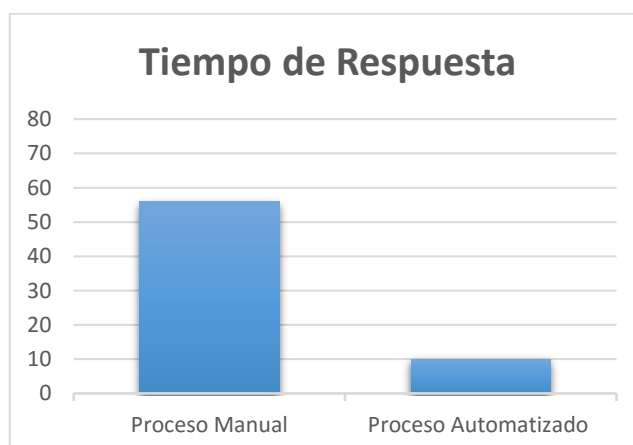


Figura 1-3: Tiempo de Respuesta

Realizado por: Jesús Mesías Ponluiza Horta. 2016

En la **Figura 1-3** se puede apreciar que con la implementación del sistema web de gestión odontológico en el departamento se redujo un tiempo de **56** minutos en realizar las tareas de gestión de información.

Para determinar cuál es la mejor manera de gestionar la información en el departamento odontológico se plantea la siguiente hipótesis: La implementación de la aplicación permitirá mejorar el tiempo utilizado para la gestión de información en el departamento. Lo que se pretende es demostrar el objetivo cuarto de este trabajo de titulación.

La distribución a utilizar es t-student y el nivel de significancia para este análisis es de $\alpha=0.05$.

Para el análisis se plantean la hipótesis nula (H_0) y la hipótesis alternativa (H_1).

H_0 : El tiempo utilizado para la gestión de información manual no mejora utilizando la gestión de información automatizada.

H_1 : El tiempo utilizado para la gestión de información manual mejora utilizando la gestión de información automatizada.

En la siguiente tabla se realizan los cálculos respectivos.

Tabla 7-3: Resultados Generales

| Opción | Con el sistema | Sin el sistema | $x_1 - \bar{x}_1$ | $x_2 - \bar{x}_2$ | $(x_1 - \bar{x}_1)^2$ | $(x_2 - \bar{x}_2)^2$ |
|-------------------------------------|------------------------------------|-------------------------------------|-------------------|-------------------|-----------------------|-----------------------|
| Gestión de datos del paciente | 14 | 2 | 0 | -0,5 | 0 | 0,25 |
| Gestión del tratamiento de paciente | 16 | 3 | 2 | 0,5 | 4 | 0,25 |
| Gestión de diagnóstico de paciente | 16 | 3 | 2 | 0,5 | 4 | 0,25 |
| Emitir reportes | 10 | 2 | -4 | -0,5 | 16 | 0,25 |
| Total | $\bar{x}_1 = 14$ | $\bar{x}_2 = 2,5$ | | | 24 | 1 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Valor de las Varianzas

$$S^2_1 = \frac{24}{14} = 1,71$$

$$S^2_2 = \frac{1}{2,5} = 0,4$$

T-Student

$$t_c = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s^2_1}{n_1} + \frac{s^2_2}{n_2}}}$$

$$t_c = \frac{14 - 2.5}{\sqrt{\frac{1,71}{4} + \frac{0,4}{4}}} = 15,83$$

Grados de libertad

$$gl = \frac{\left(\frac{s^2_1}{n_1} + \frac{s^2_2}{n_2}\right)^2}{\frac{\left(\frac{s^2_1}{n_1 - 1}\right)^2}{n_1} + \frac{\left(\frac{s^2_2}{n_2 - 1}\right)^2}{n_2}} - 2$$

$$gl = \frac{\left(\frac{1,71}{4} + \frac{0,4}{4}\right)^2}{\frac{\left(\frac{1,71}{3}\right)^2}{4} + \frac{\left(\frac{0,4}{3}\right)^2}{4}} - 2 = 1,25$$

De acuerdo a la tabla estadística de la distribución t-student, con un nivel de significancia de 0,05 y 1 grados de libertad, genera un valor de $t_t = 12,70$.

Obteniendo los siguientes resultados:

$$\text{Valor } t_c = 15,83 \text{ y } t_t = 12,70$$

Representando gráficamente.

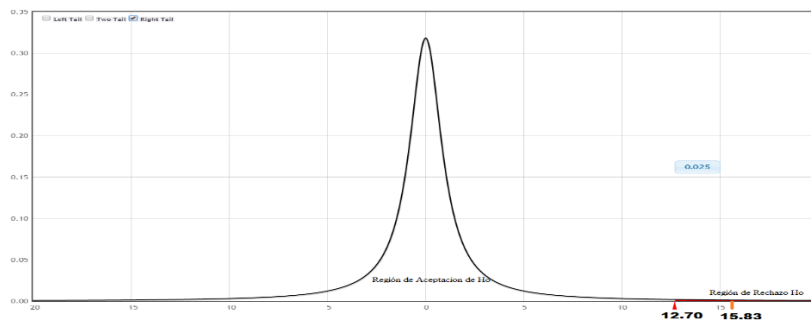


Figura 2-3: Determinación de la Hipótesis
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Se concluye que como $t_c > t_t$ entonces se rechaza la hipótesis nula (H_0) y se acepta la H_1 , es decir que el tiempo empleado para la gestión de información en el departamento si mejora con la utilización de la aplicación.

Una vez confirmado el análisis de la hipótesis se procede a realizar los cálculos correspondientes para determinar el valor cualitativo de la aplicación implementada de acuerdo a la escala de valoración establecida previamente. En la siguiente tabla 8-3 podemos analizar los datos.

Tabla 8-3: Resultados de Tiempos de respuesta

| Aplicación | Tiempo | Valoración Cuantitativa | Valoración Máxima | Valor Porcentual | Valor Porcentual Máximo |
|-------------------------------------|--------|-------------------------|-------------------|------------------|-------------------------|
| Sistema web de Gestión Odontológico | 10 | 5 | 5 | 100% | 100% |
| Gestión manual de información | 56 | 1 | 5 | 17,86% | 100% |
| Total | | | | 82,14% | 100% |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

El tiempo que se redujo con la utilización de la aplicación web es de 46 minutos dando un porcentaje de 82.14% dato muy significativo para la gestión de información que se realiza en el departamento. De acuerdo a la tabla de valoración para el indicador de tiempo de respuesta es equivale al 100% con una valoración cualitativa de excelente.

3.4 Capacidad de Análisis

Es la capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas (Ruiz 2001, p.6).

En la siguiente **Figura 3-3** se muestran la distribución de los tiempos empleados para las tareas de mantenimiento de software según McClure.

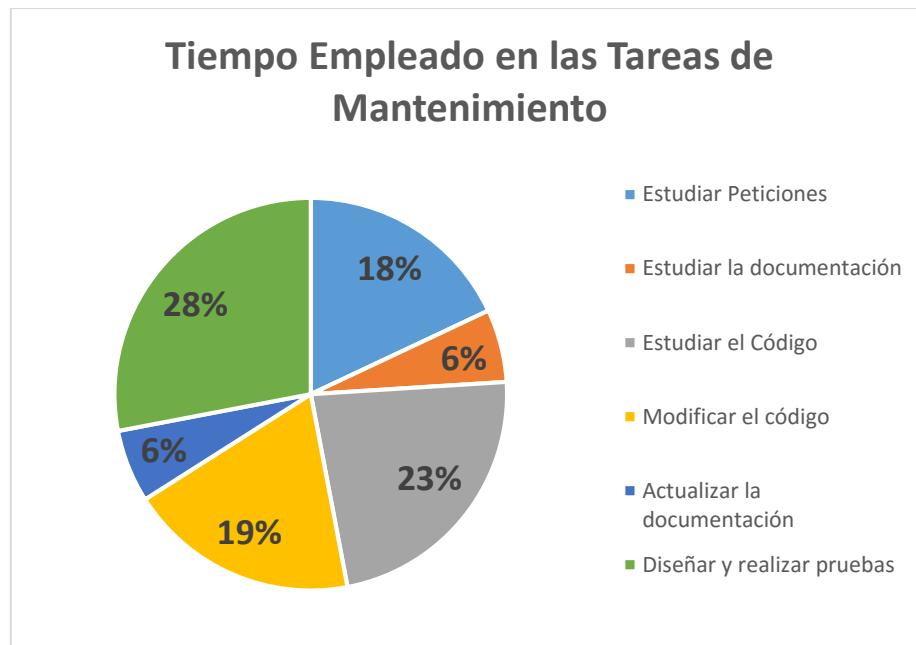


Figura 3-3: Tiempo para las tareas de mantenimiento
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Como se puede observar casi el 50% del tiempo se dedica a comprender (estudiar) el software desarrollado. Este porcentaje puede variar de acuerdo a varios factores como: información incomprensible o insuficiente, complejidad del software, mala interpretación de requerimientos, incomprensión de las herramientas utilizadas, etcétera. Para determinar el indicador capacidad de análisis de la aplicación web se establecerán valores de acuerdo a la estructura y herramientas utilizadas en el sistema.

La aplicación web fue elaborada utilizando los frameworks Spring, Primefaces, Hibernate, obteniendo una estructura ordenada y flexible. Como se puede apreciar en el capítulo III el software fue desarrollado bajo estándares de programación por ende este indicador más bien dependerá del personal encargado del mantenimiento del software, ya que al momento se realizó la capacitación respectiva y posteriormente se entregó toda la documentación relacionada con la aplicación. En base a lo especificado anteriormente se puede deducir que la aplicación web es fácil de comprender y de acuerdo a la tabla denominada capacidad de análisis es excelente.

3.5 Capacidad de Modificación

Es la capacidad del producto software de permitir implementar una modificación especificada previamente. La implementación incluye los cambios en el diseño, el código y la documentación (Ruiz 2001, p.6). Con esta métrica lo que se pretende determinar la facilidad de mantenimiento del código de la aplicación, si el código tiene malo, bueno o excelente grado de dificultad para mantener.

Para obtener la métrica Índice de Mantenibilidad (MI) se utilizó la herramienta llamada JHAWK para analizar los distintos paquetes de la aplicación web como Dao, Bo, Beans, Persistencia. Esta herramienta proporciona valores de indicación para la evaluación de mantenibilidad como podemos ver en la siguiente tabla.

Tabla 9-3: Escala de valoración índice de mantenibilidad

| Valor | Evaluación |
|-----------|------------|
| MI<65 | Malo |
| 65<=MI<85 | Bueno |
| 85>=MI | Excelente |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Para este análisis se tomará en cuenta 5 clases de cada paquete. En la siguiente **Figura 4-3** se puede apreciar las clases que van hacer analizadas.

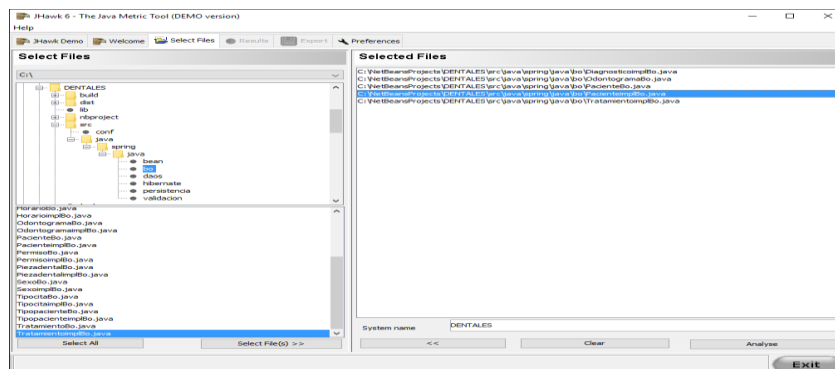


Figura 4-3: Tiempo para las tareas de mantenimiento

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Analizamos las clases del paquete de bo con la herramienta jhawk dando como resultado lo siguiente.

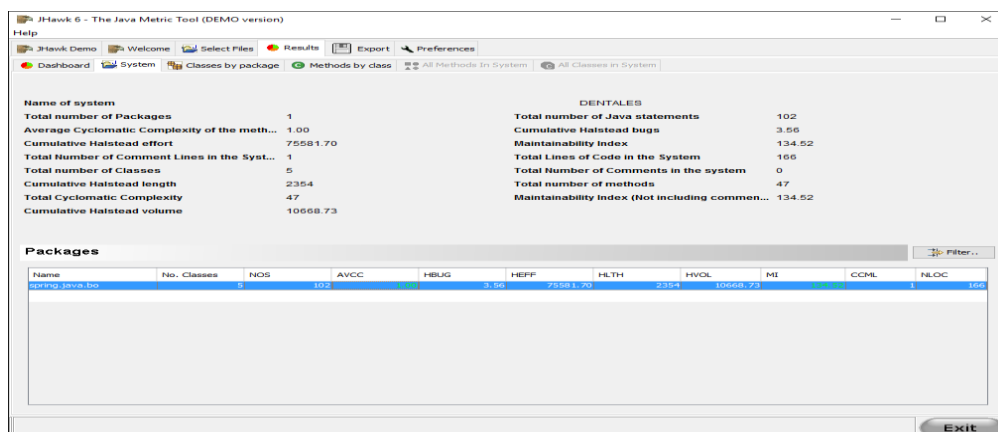


Figura 5-3: Análisis paquete Bo

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Para ver más a detalles los indicadores que nos proporciona la herramienta podemos ver la siguiente **Figura 6-3**.

| Package Information | | | |
|--|--------------------|---|--------|
| Package Name | spring.java.bo | | |
| Classes in package | DiagnosticcompilBo | | |
| Name of package | spring.java.bo | | |
| Total number of Classes | 5 | Total number of Java statements | 102 |
| Average Cyclomatic Complexity for | 1.00 | Cumulative Halstead bugs | 3.56 |
| Cumulative Halstead effort | 75581.70 | Cumulative Halstead length | 2354 |
| Cumulative Halstead volume | 10668.73 | Maintainability Index | 134.52 |
| Total Number of Comment Lines in the package | 1 | Total Lines of Code in the Package | 166 |
| Total Cyclomatic Complexity | 47 | Total Number of Comments in the package | 10 |
| Instability | 0.50 | Distance | 0.10 |
| Fan In | 1 | Total number of methods | 47 |
| Maintainability Index (Not including comments) | 134.52 | Abstractness | 0.40 |
| Maximum Cyclomatic Complexity for methods i... | 1 | Fan Out | 1 |

Figura 6-3: Detalle del análisis paquete Bo
Realizado por: Jesús Mesías Ponluiza Horta. 2016

3.5.1 Interpretación de Resultados

Tenemos un Fan Out con un valor de 1, es bajo lo que se puede deducir que las clases son robustas ante otros cambios en las otras clases. También es importante recalcar el total de líneas de código de las 5 clases analizadas dando un total de 166 es un valor moderado. Para más detalle de los datos obtenidos podemos ver el Anexo F. Por lo tanto para el indicador capacidad de modificación de acuerdo al análisis realizado se puede decir que el índice de mantenibilidad es muy bueno para todas las clases por lo tanto de acuerdo a la **Tabla 2-3** la aplicación web posee un excelente grado de mantenibilidad.

Es importante dejar en claro que las variables mantenibilidad, eficiencia pueden ser evaluadas de acuerdo a las necesidades de cada estudio y es muy probable que los resultados cambien.

3.6 Usabilidad

Según Jakob Nielsen (1999), define a la Usabilidad como el atributo de calidad que mide lo fáciles de usar que son las interfaces web. Es decir un sitio web usable es aquél en el que los usuarios pueden interactuar de la forma más fácil, cómoda, segura e inteligentemente posible.

Para evaluar el diseño del sistema web de gestión odontológico posterior a su implementación y capacitación de usuarios se elaboró un test donde se establecen ciertos parámetros para validar la usabilidad de la aplicación. Para esta prueba se aplicó el test a 24 usuarios, este dato es obtenido como promedio de los pacientes que son atendidos en el transcurso de una semana, en el Anexo

G se puede observar el número de pacientes que asisten al departamento odontológico, obteniendo los siguientes resultados.

Tabla 10-3: Test de Evaluación

| Aspecto evaluado | Si | No |
|----------------------|----|----|
| Funcionalidad | 24 | 0 |
| Navegabilidad | 23 | 1 |
| Rapidez | 23 | 1 |
| Utilidad de la Ayuda | 22 | 2 |
| Contenido | 23 | 1 |
| Guía de usuario | 22 | 2 |
| Accesibilidad | 22 | 2 |
| Satisfacción | 23 | 1 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

En la siguiente figura podemos interpretar mejor los datos.

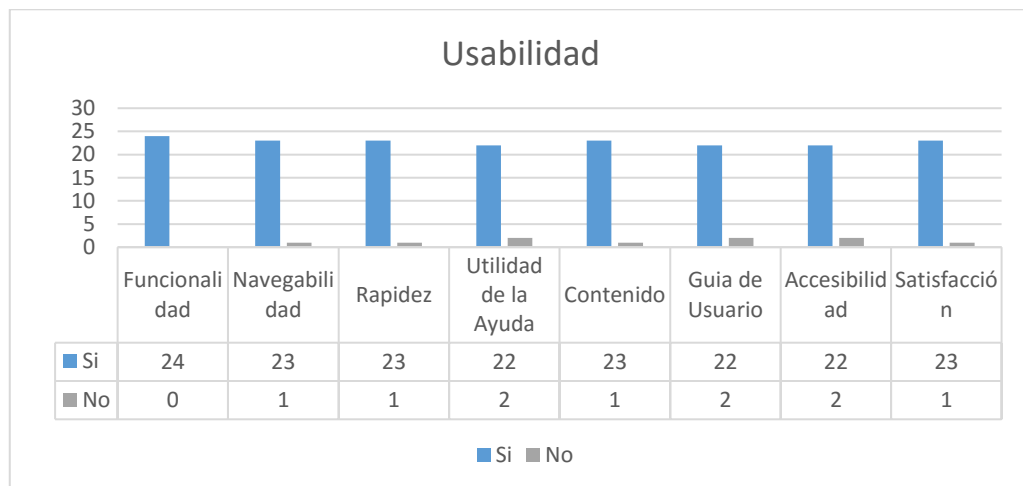


Figura 7-3: Validación de la Usabilidad
Realizado por: Jesús Mesías Ponluiza Horta. 2016

Los resultados que se obtuvieron para la validación de la aplicación por parte de los usuarios del sistema son los siguientes:

Funcionalidad: 100%

Navegabilidad: 95.83%

Rapidez: 95.83%

Utilidad de la ayuda: 91.66%

Contenido: 95.83%

Guía de usuario: 91.66%

Accesibilidad: 91.66%

Satisfacción: 95.83%

Interpretando los resultados se puede concluir que el sistema web de gestión odontológico tiene un 94.79% de aceptación por parte de los usuarios analizados y de acuerdo a la tabla de valoración es excelente.

CONCLUSIONES

- Para el desarrollo de la aplicación web de gestión odontológica se utilizó el lenguaje de programación Java utilizando los Frameworks Spring, Primefaces, Hibernate y el sistema de gestión de base de datos MySQL, generando una aplicación con una estructura sólida, ordenada y flexible, obteniendo una aceptación de un 94.79% en la población estudiada.
- Se realizó el estudio de todos los procesos que se gestionan manualmente en el Departamento de Odontología de la Empresa Eléctrica Riobamba, conjuntamente con la Odontóloga y Asistente de dicho departamento, lo que permitió establecer adecuadamente los requisitos para la elaboración de la aplicación, los mismos que fueron analizados y desarrollados utilizando la metodología ágil Scrum.
- Se ha estudiado las diferentes herramientas involucradas en el desarrollo de la aplicación web, y una de las características principales que adopta el Framework Spring es la inyección de dependencia permitiendo crear aplicaciones débilmente acopladas, Hibernate facilita el mapeo de atributos desde la base de datos y Primefaces se destaca en la creación de interfaces enriquecidas de forma sencilla, por lo tanto la integración entre Spring, Hibernate y Primefaces es ideal para desarrollar aplicaciones web sofisticadas.
- Como se puede evidenciar la aplicación web odontológica está basado en la arquitectura Modelo Vista Controlador, además de la integración de los Frameworks se ha empleado buenas prácticas de programación, obteniendo un sistema dinámico y flexible con una estructura modular facilitando la actualización y modificación de la aplicación ante posibles cambios que se puedan presentar en el departamento.
- Con la implementación de la aplicación web en el Departamento de Odontología se redujo de 56 a 10 minutos disminuyendo en un 82.14% del tiempo utilizado para los diferentes procesos de gestión de información, permitiendo gestionar la información de manera rápida y oportuna para la toma de decisiones.

RECOMENDACIONES

- Para el desarrollo de aplicaciones de software se recomienda especificar claramente los requerimientos y el alcance que tendrá el proyecto a desarrollarse.
- Incorporar aplicaciones integrando Spring con Hibernate pero con versiones actualizadas para aprovechar las nuevas funcionalidades que nos ofrecen como la clase HibernateTemplate que proporciona Spring para la incorporación con Hibernate la misma que provee varios métodos declarados en una interfaz para interactuar entre las dos herramientas.
- Al momento de desarrollar una aplicación con Spring tener en cuenta que el Framework es modular y ofrece diversas características, pero permite utilizar solo los módulos que se necesita para el desarrollo de las aplicaciones, los que resuelvan el problema de la aplicación web a implementarse.
- Se recomienda utilizar Primefaces para la capa de presentación, ya que este Framework facilita la creación de interfaces enriquecidas de una manera sencilla.
- Se recomienda utilizar el sistema web de gestión odontológico ya que disminuye tiempo en la gestión de información en el departamento.
- Se recomienda al personal técnico realizar el mantenimiento de la aplicación ante cualquier cambio que se pueda presentar en el departamento.

BIBLIOGRAFÍA

- Abud, F., & María, A.** (2000). *Calidad en la Industria del Software. La Norma ISO 9126*. Obtenido de repositorio.utp.edu.co:
http://repositorio.utp.edu.co/dspace/bitstream/11059/2737/2/0053L864e_anexo.pdf
- acensTechnologies.** (2014). *Framework para el desarrollo ágil de aplicaciones*.
- Bravo Garcia, E.** (2013). *Google web toolkit*. Obtenido de
<http://www.jtech.ua.es/jornadas/charlas/presentacionGWT.pdf>
- Çağatay, Ç.** (2015). *Primefaces*. Obtenido de www.primefaces.org:
http://www.primefaces.org/docs/guide/primefaces_user_guide_5_0.pdf
- Caliskan, M., Sevindik, K. J., & Holler, J.** (2015). *Beginning Spring*. U.S.E.: Wiley.
- Flohr, M.-A.** (1 de 10 de 2011). *Satoricode*. Obtenido de
<http://satoricode.net/2011/10/01/UnderstandingTheBenefitsOfADependencyInjectionContainerIOC.aspx>
- Freeman, E., & Otros.** (2004). *Head First Design Patterns*. U.S.E.: O'Really Media.
- Gómez, E. J.** (13 de 5 de 2013). *¿QUÉ ES UN FRAMEWORK?* Obtenido de *¿QUÉ ES UN FRAMEWORK?*: <http://edgargomez.es/que-es-un-framework/>
- Gutiérrez, J. J.** (2014). *Que es un Framework*. Obtenido de
http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- javatpoint.** (2014). *Hibernate Architecture Tutorial- javatpoint*. Obtenido de www.javatpoint.com: <http://www.javatpoint.com/hibernate-architecture>
- javatutoriales.** (23 de 12 de 2010). *Tutoriales de Programacion Java: Spring 3 - Parte 2: Contenedores de IoC e Inyección de Dependencias*. Obtenido de [Javatutoriales.com](http://www.javatutoriales.com):
<http://www.javatutoriales.com/2010/12/contenedores-de-ioc-e-inyeccion-de.html>
- Johnson, R., Donald, K., Sampaleanu, C., Harrop, R., Risberg, T., Arendsen, A., & . . . Clozel, B.** (2013). *Spring Framework Reference Documentation*. Obtenido de
<http://docs.spring.io/spring/docs/4.0.1.RELEASE/spring-framework-reference/pdf/spring-framework-reference.pdf>
- Juntadeandalucia.** (2014). *Spring*. Obtenido de [Juntadeandalucia.es](http://www.juntadeandalucia.es):
<http://www.juntadeandalucia.es/servicios/madeja/printpdf/357>
- King, G., Bauer, C., Rydahl, M., & Bernard, E.** (2010). *HIBERNATE - Persistencia relacional para Java idiomático*. Obtenido de [Docs.jboss.org](http://docs.jboss.org):
<https://docs.jboss.org/hibernate/orm/3.5/reference/es-ES/html/index.html>
- Loor, J. M.** (2009). *JSF- Java Server Faces*. Obtenido de www.ndeveloper.com:
http://www.ndeveloper.com/ndeveloperDocuments/documents/nDeveloper_JavaServerFaces.pdf
- Moreira Gibaja, V.** (2009). *Aplicaciones Web. Aplicaciones web en el entorno Empresarial, 1.*

- Netbeans.** (2016). *Netbeans.org*. Obtenido de Netbeans.org:
<https://netbeans.org/kb/docs/web/quickstart-webapps-struts.html>
- Nudia, A.** (12 de 3 de 2013). *NORMA DE EVALUACIÓN ISO/IEC 9126 / EVALUACION DE SOFTWARE*. Obtenido de Actividadreconocimiento-301569-8.blogspot.com:
<http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html>
- Palacio, J.** (2006). *El Modelo Scrum*. Obtenido de www.navwgopolis.net:
http://www.navegapolis.net/files/s/NST-010_01.pdf
- Palacio, J.** (2014). *Gestion de Proyectos Scrum Manager*. Obtenido de www.scrummanager.net:
http://www.scrummanager.net/files/sm_proyecto.pdf
- Reddy, P., & K., S.** (2013). *PrimeFaces Beginner's Guide*.
- Ruiz, F., & M.P.** (2001). *Mantenimiento de Software*. Obtenido de alarcos.esi.uclm.es:
<http://alarcos.esi.uclm.es/per/fruiz/cur/mso/trans/s3.pdf>
- Sergio, L. M.** (2001). *PROGRAMACIÓN EN INTERNET: Cliente Web*. España: Editorial Club Universitario.
- Shimokawa, L.** (2010). *Spring Core*. Obtenido de <http://www.xmind.net/m/cSKA/>
- Such, A.** (26 de 6 de 2014). *Introducción a JavaServer Faces*. Obtenido de Jtech.ua.es:
<http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion01-apuntes.html>
- Tigras, M.** (2013). *Reuniones De Scrum*. Obtenido de www.quimbiotec.gob.ve:
<http://www.quimbiotec.gob.ve/sistem/auditoria/pdf/ciudadano/mtrigasTFC0612memoria.pdf>
- Torras, I.** (2015). *Introducción — Scrum 1 documentation*. Obtenido de Metodologiascrum.readthedocs.org:
<http://metodologiascrum.readthedocs.org/en/latest/Scrum.html>
- Tovar Salazar, J.** (5 de 10 de 2010). *Vaadin, es un framework Java para aplicaciones modernas*. Obtenido de <http://blog.jfexart.com/2010/10/vaadin-un-framework-java-para.html>
- Walls, C.** (2015). *Spring in Action*. U.S.E: Manning Publications Co.

ANEXOS

Anexo A: Factibilidad Técnica

1. HARDWARE

Tabla 1: Hardware Existente

| Cantidad | Descripción | Estado |
|----------|--|--------|
| 1 | Una PC de escritorio, Intel i5(servidor) | Optimo |
| >5 | PC's de escritorio, HP intel i7, i3 (clientes) | Optimo |
| 1 | Impresora Epson tinta continua | Optimo |
| | Red | Optimo |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Tabla 2: Hardware Requerido

| Cantidad | Descripción | Observación |
|----------|-------------------------|---------------|
| 1 | Servidor | |
| >3 | PC's para clientes | SO indistinto |
| 1 | Impresora | Para reportes |
| 1 | Laptop Toshiba Intel i7 | |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

2. SOFTWARE

Tabla 3: Software Existente

| Nombre | Descripción | Licencia |
|---------|-------------------|----------------|
| Windows | Sistema Operativo | Tiene licencia |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Tabla 4: Software Requerido

| Nombre | Descripción | Licencia |
|------------|-------------------------|------------|
| Windows | Sistema Operativo | 1 licencia |
| Netbeans | IDE para el desarrollo | Gratuita |
| MySQL | Gestor de base de datos | Gratuita |
| Spring | Framework | Gratuita |
| Primefaces | Framework | Gratuita |
| Hibernate | Framework | Gratuita |
| GlassFish | Servidor | Gratuita |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

3. PERSONAL TECNICO

Tabla 5: Personal Técnico Existente

| Nombre | Función |
|----------------|---|
| Jesús Ponluiza | Programador, arquitecto, analista y diseñador de base de datos Diseñador de interfaces |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Tabla 6: Personal Técnico Requerido

| Nombre | Función |
|---------------------|---|
| Dr. Julio Santillán | Tutor del trabajo de titulación |
| Jesús Ponluiza | Programador, arquitecto, analista y diseñador de base de datos Diseñador de interfaces |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

FACTIBILIDAD ECONOMICA

1. COSTOS DE DESARROLLO

Tabla 7: Costos de Personal

| Cargo | Cantidad | Tiempo (meses) | Costo (meses) | Total |
|---------------|----------|----------------|---------------|-------|
| Desarrollador | 1 | 4 | 500 | 2000 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Tabla 8: Costos de Hardware

| Cantidad | Descripción | Costo Unitario | Total |
|----------|----------------|----------------|-------|
| 1 | Laptop Toshiba | 800 | 800 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Tabla 9: Costos de Software

| Cantidad | Descripción | Costo Unitario | Total |
|----------|-------------|----------------|-------|
| 1 | Windows | 120 | 120 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Tabla 10: Costos de Suministros

| Cantidad | Descripción | Costo Unitario | Total |
|----------|---------------------------|----------------|-----------|
| 1 | Resma de papel | 4 | 4 |
| 4 | Frascos de tinta continua | 4 | 16 |
| 1 | Suministros de oficina | 10 | 10 |
| 4 | CD's | 0.50 | 2 |
| Total | | | 32 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Tabla 11: Costos de Instalación

| Cantidad | Descripción | Costo Unitario | Total |
|----------|--------------------------|----------------|------------|
| 1 | Instalación del sistema | 60 | 50 |
| 4 | Capacitación de usuarios | 10 | 40 |
| Total | | | 100 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Tabla 12: Resumen de Costos

| Descripción | Total |
|-----------------------|-------------|
| Costos de Personal | 2000 |
| Costos de Hardware | 800 |
| Costos de Software | 120 |
| Costos de Suministros | 32 |
| Costos de Instalación | 100 |
| Total | 3052 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Anexo B: Análisis de Riesgos

Tabla 13: Análisis de riesgos

| Riesgos | | | |
|---------|--|----------|---|
| Id | Descripción | Tipo | Consecuencia |
| R1 | Cambio continuo en los requerimientos | Proyecto | Retraso del proyecto |
| R2 | Mala especificación de requisitos por parte del cliente | Proyecto | Pérdida de tiempo |
| R3 | Incumplimiento con los tiempos establecidos en el proyecto | Proyecto | Retraso de los Hitos del proyecto. |
| R4 | Falta de experiencia en las herramientas de desarrollo. | Técnico | Retraso en la finalización del sistema |
| R5 | Mal diseño de la base de datos | Técnico | Retraso en el desarrollo |
| R6 | Interfaces mal diseñadas | Técnico | Software complejo para el usuario final |
| R7 | Herramientas de desarrollo inadecuadas | Técnico | Software de mala calidad |
| R8 | Cambio del personal de la empresa | Negocio | El proyecto se suspenda |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Tabla de validación

| Rango de Probabilidad | Descripción | Valor |
|-----------------------|-------------|-------|
| 1%- 33% | Baja | 1 |
| 34%-66% | Media | 2 |
| 67%-99% | Alta | 3 |

Tabla 14: Probabilidad de riesgos

| Análisis de Riesgos | | | | | | | | |
|---------------------|--|--------------|--------------|-------|----------|-------|------------|-------|
| Id | Descripción | Probabilidad | | | Impacto | | Exposición | |
| | | Porcentaje | Probabilidad | Valor | Impacto | Valor | Exposición | Valor |
| R1 | Cambio continuo en los requerimientos | 30% | Baja | 1 | Moderado | 2 | Baja | 2 |
| R2 | Mala especificación de requisitos por parte del cliente | 55% | Media | 2 | Alto | 3 | Alta | 6 |
| R3 | Incumplimiento con los tiempos establecidos en el proyecto | 25% | Baja | 1 | Moderado | 2 | Baja | 2 |
| R4 | Falta de experiencia en las herramientas de desarrollo. | 45% | Media | 2 | Moderado | 2 | Media | 4 |
| R5 | Mal diseño de la base de datos | 25% | Baja | 1 | Moderado | 2 | Baja | 2 |
| R6 | Interfaces mal diseñadas | 25% | Baja | 1 | Moderado | 2 | Baja | 2 |
| R7 | Herramientas de desarrollo inadecuadas | 15% | Baja | 1 | Alto | 3 | Media | 3 |
| R8 | Cambio del personal de la empresa | 35% | Baja | 1 | Critico | 4 | Media | 4 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Tabla 15: Prioridad de riesgos

| Análisis de Riesgos | | | | |
|---------------------|--|------------|-------|-----------|
| Id | Descripción | Exposición | Valor | Prioridad |
| R2 | Mala especificación de requisitos por parte del cliente | Alta | 6 | 1 |
| R4 | Falta de experiencia en las herramientas de desarrollo. | Media | 4 | 2 |
| R7 | Herramientas de desarrollo inadecuadas | Media | 3 | 2 |
| R8 | Cambio del personal de la empresa | Media | 4 | 2 |
| R1 | Cambio continuo en los requerimientos | Baja | 2 | 3 |
| R3 | Incumplimiento con los tiempos establecidos en el proyecto | Baja | 2 | 3 |
| R5 | Mal diseño de la base de datos | Baja | 2 | 3 |
| R6 | Interfaces mal diseñadas | Baja | 2 | 3 |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Tabla 16: Gestión del Riesgo 2

| Hoja de Gestión de Riesgo | | | |
|---|----------------------------------|-------------------------------------|---------------------|
| Id del Riesgo: R2 | | Fecha: | |
| Probabilidad: media Valor: 2 | Impacto: alto Valor: 3 | Exposición: alta Valor: 6 | Prioridad: 1 |
| Descripción: Mala especificación de requerimientos por parte del cliente. | | | |
| REFINAMIENTO | | | |
| Causas | | | |
| Falta de comunicación entre el cliente y el equipo de desarrollo. Mala explicación de la lógica del negocio. | | | |
| Consecuencias | | | |
| Retraso en la entrega del proyecto. Mala especificación de requerimientos | | | |
| Reducción | | | |
| Observar, Analizar detalladamente los procesos que se llevan a cabo en el departamento odontológico. | | | |
| Supervisión | | | |
| Estar en constante comunicación con el cliente. | | | |
| Gestión | | | |
| Reuniones con el cliente para cuestionar el funcionamiento del negocio a más detalles. | | | |
| Estado Actual | | | |
| Fase de Reducción iniciada | | x | |
| Fase de Supervisión iniciada | | | |
| Gestión del Riesgo | | | |
| Responsable | | | |
| Jesús Ponluiza | | | |

Realizado por: Jesús Mesías Ponluiza Horta. 2016

Anexo C: Diccionario de Datos

Tabla 17: Diccionario Tabla cita

| Campo | Tipo/Tamaño | Nulo | Enlaces a | PK | FK |
|-------------------|--------------|------|----------------------------|----|----|
| <u>Idcita</u> | int(11) | No | | Si | |
| Cedula | varchar(10) | Sí | paciente -> cedula | | Si |
| Fechacita | date | Sí | | | |
| Motivo | varchar(800) | Sí | | | |
| Idestadocita | int(11) | Sí | estadocita -> idestadocita | | Si |
| Idtipocita | int(11) | Sí | tipocita -> idtipocita | | Si |
| Idhorario | int(11) | Sí | horario -> idhorario | | Si |
| fechacitaatendida | date | Sí | | | |

Tabla 18: Diccionario Tabla departamento

| Columna | Tipo/Tamaño | Null | PK | FK |
|-----------------------|--------------|------|----|----|
| <u>iddepartamento</u> | int(11) | No | Si | |
| Descripción | varchar(800) | Yes | | |

Tabla 19: Diccionario Tabla diagnostico

| Columna | Tipo/Tamaño | Null | PK | FK |
|----------------------|--------------|------|----|----|
| <u>iddiagnostico</u> | int(11) | No | Si | |
| Descripción | varchar(500) | Yes | | |

Tabla 20: Diccionario Tabla estadocita

| Columna | Tipo/Tamaño | Null | PK | FK |
|---------------------|--------------|------|----|----|
| <u>idestadocita</u> | int(11) | No | Si | |
| Descripción | varchar(800) | Yes | | |

Tabla 21: Diccionario Tabla estadopersona

| Columna | Tipo/Tamaño | Null | PK | FK |
|------------------------|--------------|------|----|----|
| <u>idestadopersona</u> | int(11) | No | Si | |
| Descripción | varchar(800) | Yes | | |

Tabla 22: Diccionario Tabla horario

| Columna | Tipo/Tamaño | Null | PK | FK |
|------------------|-------------|------|----|----|
| <u>Idhorario</u> | int(11) | No | Si | |
| descripcion | varchar(50) | Yes | | |

Tabla 23: Diccionario Tabla odontograma

| Campo | Tipo | Nulo | Enlaces a | PK | FK |
|----------------------|-------------|------|--------------------|----|----|
| <u>idodontograma</u> | int(11) | No | | Si | |
| Cedula | varchar(10) | Sí | paciente -> cedula | | Si |
| numeropieza | int(11) | Sí | | | |

| | | | | | |
|-----------------|---------------|----|--|--|--|
| Diagnostico | varchar(2000) | Sí | | | |
| Observacion | varchar(2000) | Sí | | | |
| Estado | int(11) | Sí | | | |
| fechaevaluacion | date | Sí | | | |
| numerohclinica | int(11) | Sí | | | |

Tabla 24: Diccionario Tabla permiso

| Columna | Tipo/Tamaño | Null | PK | FK |
|------------------|-------------|------|----|----|
| <u>Idusuario</u> | int(11) | No | Si | |
| descripcion | varchar(50) | Yes | | |

Tabla 25: Diccionario Tabla piezadental

| Columna | Tipo/Tamaño | Null | PK | FK |
|----------------------|--------------|------|----|----|
| <u>idpiezadental</u> | int(11) | No | Si | |
| numeropieza | int(11) | No | | |
| nombrepieza | varchar(800) | Yes | | |

Tabla 26: Diccionario Tabla sexo

| Columna | Tipo/Tamaño | Null | PK | FK |
|---------------|-------------|------|----|----|
| <u>Idsexo</u> | int(11) | No | Si | |
| descripcion | varchar(50) | Yes | | |

Tabla 27: Diccionario Tabla tipocita

| Columna | Tipo/Tamaño | Null | PK | FK |
|-------------------|--------------|------|----|----|
| <u>Idtipocita</u> | int(11) | No | Si | |
| descripcion | varchar(800) | Yes | | |

Tabla 28: Diccionario Tabla tipopaciente

| Columna | Tipo/Tamaño | Null | PK | FK |
|----------------------|--------------|------|----|----|
| <u>idtipopersona</u> | int(1) | No | Si | |
| Descripción | varchar(100) | Yes | | |

Tabla 29: Diccionario Tabla tratamiento

| Campo | Tipo | Nulo | Enlaces a | PK | FK |
|----------------------|-------------|------|--------------------|----|----|
| <u>idtratamiento</u> | int(11) | No | | Si | |
| Cedula | varchar(10) | Sí | paciente -> cedula | | Si |
| numeropieza | int(11) | Sí | | | |
| restauracion | varchar(20) | Sí | | | |
| Curacion | varchar(20) | Sí | | | |
| endodoncia | varchar(20) | Sí | | | |
| Exodoncia | varchar(20) | Sí | | | |
| Profilaxis | varchar(20) | Sí | | | |

| | | | | | |
|------------------|--------------|----|--|--|--|
| Tflour | varchar(20) | Sí | | | |
| Rx | varchar(20) | Sí | | | |
| Ci | varchar(20) | Sí | | | |
| Otros | varchar(20) | Sí | | | |
| fechatratamiento | date | Sí | | | |
| observaciones | varchar(800) | Sí | | | |

Tabla 30: Diccionario Tabla enfermedadperiodental

| Campo | Tipo | Nulo | Enlaces a | PK | FK |
|-----------------------|--------------|------|--------------------|----|----|
| <u>identfermedadp</u> | int(11) | No | | Si | |
| Cedula | varchar(10) | Sí | paciente -> cedula | | Si |
| materiaalba | varchar(2) | Sí | | | |
| placabacteriana | varchar(2) | Sí | | | |
| csupragingival | varchar(2) | Sí | | | |
| csubgingival | varchar(2) | Sí | | | |
| bolsaperiodental | varchar(2) | Sí | | | |
| movilidaddental | varchar(2) | Sí | | | |
| observaciones | varchar(800) | Sí | | | |
| fechaderegistro | date | Sí | | | |

Tabla 31: Diccionario Tabla oclusión

| Campo | Tipo | Nulo | Enlaces a | PK | Fk |
|--------------------|--------------|------|--------------------|----|----|
| <u>idocclusion</u> | int(11) | No | | Si | |
| Cedula | varchar(10) | Sí | paciente -> cedula | | Si |
| mdentaria | varchar(2) | Sí | | | |
| mmaxilar | varchar(2) | Sí | | | |
| observaciones | varchar(800) | Sí | | | |
| fechaderegistro | date | Sí | | | |

Anexo D: Sprint y Tareas de Ingeniería

Tareas de Ingeniería Sprint 0

| TAREA DE INGENIERÍA | |
|---|---------------------------------|
| Número de Tarea: 2 | Número Historia: HUT2 |
| Nombre de la Tarea: diseño de la estructura, arquitectura, capas de la aplicación. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 6 |
| Fecha de Inicio: 15-10-2015 | Fecha de Fin: 05-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se realizara la estructura de funcionalidades del sistema de acuerdo a los roles. | |

| TAREA DE INGENIERÍA | |
|--|---------------------------------|
| Número de Tarea: 3 | Número Historia: HUT3 |
| Nombre de la Tarea: analizar y diseñar la base de datos. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 7 |
| Fecha de Inicio: 15-10-2015 | Fecha de Fin: 05-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara la base de datos en base a los requerimientos recolectados. | |

| TAREA DE INGENIERÍA | |
|--|---------------------------------|
| Número de Tarea: 4 | Número Historia: HUT5 |
| Nombre de la Tarea: generación de paquetes | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 3 |
| Fecha de Inicio: 15-10-2015 | Fecha de Fin: 05-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se generara los paquetes necesarios para tener organizado el código. | |

| TAREA DE INGENIERÍA | |
|--|---------------------------------|
| Número de Tarea: 5 | Número Historia: HUT4 |
| Nombre de la Tarea: codificación del sistema, preparación de la aplicación. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 10 |
| Fecha de Inicio: 15-10-2015 | Fecha de Fin: 05-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara la aplicación en Netbeans y se dará soporte para los frameworks spring framework, Primefaces, Hibernate. | |

| TAREA DE INGENIERÍA | |
|---|---------------------------------|
| Número de Tarea: 6 | Número Historia: HUT6 |
| Nombre de la Tarea: diseño de interfaces de usuario | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 10 |
| Fecha de Inicio: 15-10-2015 | Fecha de Fin: 05-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se realizara el diseño de las interfaces (colores, temas, logotipos y se especificara una estructura que tendrán las ventanas). | |

Pruebas de Aceptación Sprint 0

| PRUEBAS DE ACEPTACIÓN-SPRINT 0 | |
|---------------------------------------|--|
| Nombre de la prueba | Descripción |
| Test de la base de datos. | Verificar que se ha creado todos los campos de acuerdo a los requerimientos especificados. (administrador de software) |
| Test frameworks | Verificar en la carpeta de librerías la versión de cada uno de los frameworks con los que se va a trabajar. |
| Test conexión | Verificar que estén levantados los servicios para que la aplicación tenga conexión a la base de datos. |

SPRINT 1

Historias de Usuario Sprint 1

| HISTORIA DE USUARIO | |
|---|-----------------------------|
| Número: HU8 | Usuario: Dentista |
| Nombre de la Historia: El sistema permitirá realizar un CRUD de las citas médicas. | |
| Prioridad en el Negocio: Alta (Alta / Media / Baja) | Spring asignado: 1 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 10 |
| Descripción: Como dentista necesito ingresar, editar, eliminar las citas médicas. | |
| Observaciones: las citas médicas deben ser ingresadas en horarios de trabajo. Se listaran de otro color las citas médicas que se han caducado. | |

| HISTORIA DE USUARIO | |
|--|-----------------------------|
| Número: HU9 | Usuario: Dentista |
| Nombre de la Historia: El sistema permitirá ingresar el diagnóstico dental a los pacientes. | |
| Prioridad en el Negocio: Alta (Alta / Media / Baja) | Spring asignado: 1 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 15 |
| Descripción: Como dentista necesito realizar el diagnóstico dental a cada paciente. | |
| Observaciones: el diagnóstico se realizara a todos los pacientes que ingresan al consultorio. | |

| HISTORIA DE USUARIO | |
|--|----------------------------|
| Número: HU10 | Usuario: Dentista |
| Nombre de la Historia: el sistema permitirá consultar el diagnóstico dental de cada paciente. | |
| Prioridad en el Negocio: Alta (Alta / Media / Baja) | Spring asignado: 1 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 7 |
| Descripción: Como dentista necesito consultar el diagnóstico dental de cada paciente. | |
| Observaciones: el rol dentista y paciente pueden consultar el diagnóstico dental. | |

| HISTORIA DE USUARIO | |
|--|----------------------------|
| Número: HU11 | Usuario: Dentista |
| Nombre de la Historia: el sistema permitirá modificar los el diagnóstico dental de cada paciente. | |
| Prioridad en el Negocio: Alta (Alta / Media / Baja) | Spring asignado: 1 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 8 |
| Descripción: Como dentista necesito editar el diagnóstico dental de cada paciente. | |
| Observaciones: solo puede hacer el rol Dentista. | |

| HISTORIA DE USUARIO | |
|---|-----------------------------|
| Número: HU12 | Usuario: Dentista |
| Nombre de la Historia: el sistema permitirá ingresar el tratamiento dental que se realiza cada paciente. | |
| Prioridad en el Negocio: Alta (Alta / Media / Baja) | Spring asignado: 1 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 10 |
| Descripción: Como dentista necesito ingresar el tratamiento dental que el paciente se realizó. | |
| Observaciones: los tratamientos están organizados por fechas. | |

Tareas de Ingeniería Sprint 1

| TAREA DE INGENIERÍA | |
|--|---------------------------------|
| Número de Tarea: 7 | Número Historia: HU1 |
| Nombre de la Tarea: realizar un insertar, modificar, eliminar, listar de los datos de los pacientes. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 7 |
| Fecha de Inicio: 11-01-2016 | Fecha de Fin: 11-02-2016 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara todas las funciones, métodos, beans, controladores, clases para la generación del CRUD de los datos del paciente. Se realizara las validaciones necesarias. | |

| TAREA DE INGENIERÍA | |
|---|---------------------------------|
| Número de Tarea: 8 | Número Historia: HU8 |
| Nombre de la Tarea: creación de las vistas, clases métodos para realizar el crud de las citas médicas. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 6 |
| Fecha de Inicio: 11-01-2016 | Fecha de Fin: 11-02-2016 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara todas las funciones, métodos, beans, controladores, diseño de ventanas, clases para la generación del CRUD de la cita médica. | |

| TAREA DE INGENIERÍA | |
|---|---------------------------------|
| Número de Tarea: 9 | Número Historia: HU9 |
| Nombre de la Tarea: realizar las funciones, métodos, formulario, botones para el ingreso del diagnóstico dental de cada paciente. | |
| Tipo de Tarea: Corrección (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 15 |
| Fecha de Inicio: 11-01-2015 | Fecha de Fin: 11-02-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creará todos los métodos, funciones, que permitan el registro del diagnóstico dental. Se realizara las validaciones necesarias. | |

| TAREA DE INGENIERÍA | |
|--|---------------------------------|
| Número de Tarea: 10 | Número Historia: HU10 |
| Nombre de la Tarea: creación de sentencias HQL, clases, métodos que permitan realizar la consulta del diagnóstico del paciente. | |
| Tipo de Tarea: Corrección (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 6 |
| Fecha de Inicio: 11-01-2016 | Fecha de Fin: 11-02-2016 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creará métodos que permitan realizar la búsqueda a través de un parámetro. Se realizara las validaciones necesarias. | |

| TAREA DE INGENIERÍA | |
|--|---------------------------------|
| Número de Tarea: 11 | Número Historia: HU11 |
| Nombre de la Tarea: realizar funciones de modificación, métodos y el diseño del formulario. | |
| Tipo de Tarea: Corrección (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 7 |
| Fecha de Inicio: 11-01-2016 | Fecha de Fin: 11-02-2016 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara los métodos necesarios para modificar los datos del diagnóstico dental de cada paciente. Se realizara las validaciones necesarias. | |

| TAREA DE INGENIERÍA | |
|---|---------------------------------|
| Número de Tarea: 12 | Número Historia: HU12 |
| Nombre de la Tarea: creación de métodos, formularios, sentencias HQL, beans en el contenedor de spring. | |
| Tipo de Tarea: Corrección (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 4 |
| Fecha de Inicio: 11-01-2016 | Fecha de Fin: 11-01-2016 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara todos los métodos que permitan el ingreso de los tratamientos que se realiza el paciente. Se realizara las validaciones necesarias. | |

Pruebas de aceptación Sprint 1

| PRUEBAS DE ACEPTACIÓN-SPRINT 1 | |
|---------------------------------------|--|
| Nombre de la prueba | Descripción |
| Test ingreso de paciente. | Probar que. Verificar que se encuentren todos los campos necesarios de los pacientes. Todos los campos son obligatorios (*). Campos validados solo números y solo letras de ser el caso. El sistema verificara si la cédula ingresada es válida, caso contrario emitirá un mensaje de cédula incorrecta. Una sola vez se ingresara al paciente. |
| Test cita médica. | Probar que. Todos los campos son obligatorios (*). Para realizar la cita médica se debe ingresar primero al paciente. |
| Test diagnostico dental. | Probar que. Se registren los datos que se ingresan. |
| Test modificar diagnostico dental. | Verificar los datos modificados. |
| Test ingreso tratamiento dental. | Probar que se listen todos los pacientes para posteriormente seleccionar al paciente para ingresar el tratamiento. Probar que se listen todas las patologías. Probar que se listen todas las piezas dentales. Verificar el mensaje de guardado correctamente. |

SPRINT 2

Historias de Usuario Sprint 2

| HISTORIA DE USUARIO | |
|--|----------------------------|
| Número: HU13 | Usuario: Dentista |
| Nombre de la Historia: el sistema permitirá la modificación de los tratamientos de los pacientes. | |
| Prioridad en el Negocio: Alta (Alta / Media / Baja) | Spring asignado: 2 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 4 |
| Descripción: Como dentista necesito editar el tratamiento dental del paciente. | |
| Observaciones: | |

| HISTORIA DE USUARIO | |
|---|----------------------------|
| Número: HU14 | Usuario: Dentista |
| Nombre de la Historia: el sistema permitirá consultar los tratamientos que se realiza cada paciente. | |
| Prioridad en el Negocio: Alta (Alta / Media / Baja) | Spring asignado: 2 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 6 |
| Descripción: Como dentista necesito consultar los tratamientos que se realizó el paciente. | |
| Observaciones: esta opción tendrá acceso el rol dentista y el rol paciente. | |

| HISTORIA DE USUARIO | |
|--|----------------------------|
| Número: HU15 | Usuario: Dentista |
| Nombre de la Historia: el sistema permitirá el ingreso de las patologías. | |
| Prioridad en el Negocio: Media (Alta / Media / Baja) | Spring asignado: 2 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 4 |
| Descripción: Como dentista necesito ingresar las patologías. | |
| Observaciones: CRUD. | |

| HISTORIA DE USUARIO | |
|--|----------------------------|
| Número: HU16 | Usuario: Dentista |
| Nombre de la Historia: el sistema permitirá registrar los tratamientos que se brindan en el departamento. | |
| Prioridad en el Negocio: Baja (Alta / Media / Baja) | Spring asignado: 2 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 4 |
| Descripción: Como dentista necesito ingresar los tratamientos que se brindan en el departamento. | |
| Observaciones: CRUD. | |

| HISTORIA DE USUARIO | |
|---|----------------------------|
| Número: HU17 | Usuario: Dentista |
| Nombre de la Historia: el sistema permitirá listar los pacientes que fueron atendidos. | |
| Prioridad en el Negocio: Media (Alta / Media / Baja) | Spring asignado: 2 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 5 |
| Descripción: Como dentista necesito tener una lista de los pacientes atendidos. | |
| Observaciones: se creara filtros. | |

| HISTORIA DE USUARIO | |
|---|----------------------------|
| Número: HU18 | Usuario: Dentista |
| Nombre de la Historia: el sistema permitirá listar todas las citas médicas pendientes. | |
| Prioridad en el Negocio: Media (Alta / Media / Baja) | Spring asignado: 2 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 5 |
| Descripción: Como dentista necesito tener una lista de las citas médicas pendientes. | |
| Observaciones: | |

| HISTORIA DE USUARIO | |
|--|---------------------------|
| Número: HU19 | Usuario: Dentista |
| Nombre de la Historia: el sistema permitirá conocer los pacientes cuyo contrato de trabajo ha caducado. | |
| Prioridad en el Negocio: Media | Spring asignado: 2 |

| | |
|---|----------------------------|
| (Alta / Media / Baja) | |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 5 |
| Descripción: Como dentista necesito conocer los pacientes que ya han finalizado el contrato. | |
| Observaciones: | |

| HISTORIA DE USUARIO | |
|---|-----------------------------|
| Número: HU20 | Usuario: Dentista |
| Nombre de la Historia: el sistema permitirá listar los pacientes atendidos por primera vez en el departamento. | |
| Prioridad en el Negocio: Media (Alta / Media / Baja) | Spring asignado: 0 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 10 |
| Descripción: Como dentista necesito conocer los pacientes que se atienden por primera vez en el departamento. | |
| Observaciones: | |

Tareas de Ingeniería Sprint 2

| TAREA DE INGENIERÍA | |
|--|---------------------------------|
| Número de Tarea: 13 | Número Historia: HU13 |
| Nombre de la Tarea: creación de métodos, sentencia HQL, formulario, beans necesarios para la modificación del tratamiento dental del paciente. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 6 |
| Fecha de Inicio: 06-11-2015 | Fecha de Fin: 26-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara todo el código en las distintas capas, que permita la modificación del tratamiento dental. Se realizara las validaciones necesarias. | |

| TAREA DE INGENIERÍA | |
|--|---------------------------------|
| Número de Tarea: 14 | Número Historia: HU14 |
| Nombre de la Tarea: creación de la consulta HQL, métodos, funciones, vistas, beans. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 7 |
| Fecha de Inicio: 06-11-2015 | Fecha de Fin: 26-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara los métodos en las distintas capas que permitan realizar la consulta. Se realizaran validaciones, filtros. | |

| TAREA DE INGENIERÍA | |
|--|---------------------------------|
| Número de Tarea: 15 | Número Historia: HU15 |
| Nombre de la Tarea: creación de métodos, funciones, beans, sentencia HQL. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 5 |
| Fecha de Inicio: 06-11-2015 | Fecha de Fin: 26-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: CRUD. Se realizara las validaciones correspondientes. | |

| TAREA DE INGENIERÍA | |
|---|--------------------------|
| Número de Tarea: 16 | Número Historia: HU16 |
| Nombre de la Tarea: creación de métodos, funciones, beans, sentencia HQL. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados:5 |
| Fecha de Inicio: 06-11-2015 | Fecha de Fin: 26-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: CRUD. Validaciones para las vistas. | |

| TAREA DE INGENIERÍA | |
|--|--------------------------|
| Número de Tarea: 17 | Número Historia: HU17 |
| Nombre de la Tarea: creación de la consulta HQL, vistas. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados:4 |
| Fecha de Inicio: 06-11-2015 | Fecha de Fin: 26-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara las funciones en las clases correspondientes para la generación de los pacientes atendidos. | |

| TAREA DE INGENIERÍA | |
|---|--------------------------|
| Número de Tarea: 18 | Número Historia: HU18 |
| Nombre de la Tarea: creación de la consulta HQL, vista, métodos. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados:4 |
| Fecha de Inicio: 06-11-2015 | Fecha de Fin: 26-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara filtros para la consulta. | |

| TAREA DE INGENIERÍA | |
|--|--------------------------|
| Número de Tarea: 19 | Número Historia: HU19 |
| Nombre de la Tarea: crear un método para obtener la lista de pacientes que han finalizado el contrato. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados:4 |
| Fecha de Inicio: 06-11-2015 | Fecha de Fin: 26-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara las sentencias HQL, métodos que permitan la generación de la lista. Filtros | |

| TAREA DE INGENIERÍA | |
|---|--------------------------|
| Número de Tarea: 20 | Número Historia: HU20 |
| Nombre de la Tarea: crear la funciones, HQL, vista | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados:4 |
| Fecha de Inicio: 06-11-2015 | Fecha de Fin: 26-11-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: listado con filtros. | |

Pruebas de aceptación Sprint 2

| PRUEBAS DE ACEPTACIÓN-SPRINT 2 | |
|---------------------------------------|--|
| Nombre de la prueba | Descripción |
| Test modificación tratamiento dental. | Verificar que se encuentren todos los campos a modificar. Los campos obligatorios no se pueden quedar en vacíos (*). Campos validados solo números y solo letras de ser el caso. |

| | |
|---|---|
| | Probar que salga el mensaje de confirmación: datos modificados correctamente. |
| Test consulta de tratamientos del paciente | Probar realizando sub consultas con los filtros que se presenta en la tabla. |
| Test ingreso de patologías. | Probar que todos los campos son obligatorios (*). Probar que se pueda ingresar, modificar, eliminar los datos. |
| Test CRUD tratamientos que brinda el departamento. | Verificar campos obligatorios (*). Probar que salgan los mensajes de confirmación al momento de realizar una acción ejemplo un insertar. |
| Test lista pacientes atendidos. | Probar realizando sub consultas con los filtros que se presenta en la tabla. |
| Test lista citas pendientes. Test lista pacientes contrato caducado. Test lista pacientes atendidos por primera consulta. | Probar realizando sub consultas con los filtros que se presenta en la tabla. |

SPRINT 3

Historias de Usuario Sprint 3

| HISTORIA DE USUARIO | |
|---|---|
| Número: HU21 | Usuario: Administrador de Software |
| Nombre de la Historia: gestión de roles del sistema | |
| Prioridad en el Negocio: Alta (Alta / Media / Baja) | Spring asignado: 3 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 10 |
| Descripción: Como administrador de software necesito gestionar los roles de usuarios en el sistema. | |
| Observaciones: el administrador asignara los perfiles correspondientes dentista, asistente, paciente de ser el caso. | |

| HISTORIA DE USUARIO | |
|--|---|
| Número: HU22 | Usuario: Administrador de Software |
| Nombre de la Historia: gestión de acceso al sistema. | |
| Prioridad en el Negocio: Alta (Alta / Media / Baja) | Spring asignado: 3 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 8 |
| Descripción: Como administrador de software necesito gestionar el acceso al sistema. | |
| Observaciones: al usuario que se le registra, no tiene acceso al sistema mientras no se le asigne la contraseña de usuario, la contraseña asignada será la cédula de identidad, que esta a su vez puede ser modificada por el usuario una vez ingresado al sistema, de la misma manera se le debe asignar un rol correspondiente. | |

| HISTORIA DE USUARIO | |
|---|----------------------------|
| Número: HU23 | Usuario: Paciente |
| Nombre de la Historia: paciente gestión de su cita médica | |
| Prioridad en el Negocio: Media (Alta / Media / Baja) | Spring asignado: 3 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 6 |
| Descripción: Como paciente necesito ingresar la cita médica. | |
| Observaciones: el paciente realizara su cita médica de acuerdo al horario de trabajo del dentista. | |

| HISTORIA DE USUARIO | |
|--|----------------------------|
| Número: HU24 | Usuario: Paciente |
| Nombre de la Historia: modificación cita médica paciente. | |
| Prioridad en el Negocio: Media (Alta / Media / Baja) | Spring asignado: 3 |
| Riesgos en el Desarrollo: Bajo | Puntos Estimados: 5 |

| | |
|--|--|
| (Alto / Media / Bajo) | |
| Descripción: Como paciente necesito modificar la cita médica. | |
| Observaciones: | |

| HISTORIA DE USUARIO | |
|---|----------------------------|
| Número: HU25 | Usuario: Paciente |
| Nombre de la Historia: cancelar cita médica paciente. | |
| Prioridad en el Negocio: Media (Alta / Media / Baja) | Spring asignado: 3 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 3 |
| Descripción: Como paciente necesito cancelar la cita médica. | |
| Observaciones: | |

| HISTORIA DE USUARIO | |
|---|-----------------------------|
| Número: HU26 | Usuario: Paciente |
| Nombre de la Historia: lista de tratamientos aplicados paciente | |
| Prioridad en el Negocio: Media (Alta / Media / Baja) | Spring asignado: 3 |
| Riesgos en el Desarrollo: Bajo (Alto / Media / Bajo) | Puntos Estimados: 10 |
| Descripción: Como paciente necesito ver los tratamientos que me aplicaron. | |
| Observaciones: | |

Tareas de Ingeniería Sprint 3

| TAREA DE INGENIERÍA | |
|---|---------------------------------|
| Número de Tarea: 21 | Número Historia: HU21 |
| Nombre de la Tarea: crear funciones, métodos, vistas para la administración de los roles. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 7 |
| Fecha de Inicio: 27-11-2015 | Fecha de Fin: 22-12-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creará para asignar roles a los usuarios. Se creara la vista para la gestión de asignación de los roles a los usuarios. | |

| TAREA DE INGENIERÍA | |
|---|---------------------------------|
| Número de Tarea: 22 | Número Historia: HU22 |
| Nombre de la Tarea: crear funciones, métodos, vistas, que gestionen el acceso de los usuarios al sistema. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 7 |
| Fecha de Inicio: 27-11-2015 | Fecha de Fin: 22-12-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara las funciones que permitan asignar el acceso al sistema así como también quitar el acceso al sistema. | |

| TAREA DE INGENIERÍA | |
|---|---------------------------------|
| Número de Tarea: 23 | Número Historia: HU23 |
| Nombre de la Tarea: se creara las funciones, métodos, vistas para para que ingrese la cita médica el paciente. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 3 |
| Fecha de Inicio: 27-11-2015 | Fecha de Fin: 22-12-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creará las funciones, métodos, diseño de las vistas para que el paciente ingrese la cita. | |

| TAREA DE INGENIERÍA | |
|--|---------------------------------|
| Número de Tarea: 24 | Número Historia: HU24 |
| Nombre de la Tarea: se crea las funciones, vistas para modificar la cita del paciente. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 4 |
| Fecha de Inicio: 27-11-2015 | Fecha de Fin: 22-12-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara las funciones, vistas para la modificación de la cita médica para el rol paciente. | |

| TAREA DE INGENIERÍA | |
|---|---------------------------------|
| Número de Tarea: 25 | Número Historia: HU25 |
| Nombre de la Tarea: se crea las funciones, métodos para cancelar la cita del paciente. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 4 |
| Fecha de Inicio: 27-11-2015 | Fecha de Fin: 22-12-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creará las funciones para que el rol paciente pueda cancelar su cita médica. | |

| TAREA DE INGENIERÍA | |
|--|---------------------------------|
| Número de Tarea: 26 | Número Historia: HU26 |
| Nombre de la Tarea: crear funciones, métodos, vistas, consulta HQL. | |
| Tipo de Tarea: Desarrollo (Desarrollo / Corrección / Mejora / Actividad inicial) | Puntos estimados: 4 |
| Fecha de Inicio: 27-11-2015 | Fecha de Fin: 22-12-2015 |
| Programador Responsable: Jesus Ponluiza | |
| Descripción: se creara las funciones, métodos y diseño de las vistas, para que el paciente pueda ver los tratamientos que se le han aplicado. | |

Pruebas de aceptación Sprint 3

| PRUEBAS DE ACEPTACIÓN-SPRINT 3 | |
|---------------------------------------|--|
| Nombre de la prueba | Descripción |
| Test gestión de roles de usuario. | Verificar que las funcionalidades (opciones del menú) correspondan al rol asignado. Los campos obligatorios no se pueden quedar en vacíos (*). Verificar que salga el mensaje de confirmación: datos modificados correctamente. Verificar que se listen todos los roles. Verificar que se puede cambiar de rol a los usuarios. |
| Test gestión de acceso al sistema. | Probar que al usuario ingresado se le puede asignar la cédula como contraseña por defecto. Verificar que la contraseña este encriptada. Probar cambiando de contraseña. Verificar que el usuario no tenga acceso al sistema una vez que se le quite el acceso. |
| Test ingreso cita médica. | Probar que todos los campos con (*) son obligatorios. Probar que se registre la cita médica al paciente autenticado. |
| Test modificar cita médica. | Verificar campos obligatorios (*). Verificar que salga el mensaje de confirmación de modificación. |
| Test cancelar cita. | Verificar que se elimina la cita. |
| Test ver tratamientos aplicados. | Probar que se listen los tratamientos del usuario autenticado. |

Anexo E: Encuesta

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTA DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS**

La información obtenida será utilizada para analizar la situación actual del departamento odontológico y realizar el estudio de la variable eficiencia del sistema de gestión odontológico en el departamento de la EERSA.

Nota: Marque con una X la respuesta de su elección.

1. ¿Mediante qué proceso se realiza la gestión de información en el departamento?

Manual Sistema Informático

2. ¿Qué tiempo emplea en la gestión de los datos de los pacientes?

| Tiempo utilizado en minutos | Ingresar | Modificar | Eliminar | Listar |
|-----------------------------|----------|-----------|----------|--------|
| 1 | | | | |
| 3 | | | | |
| 5 | | | | |
| 7 | | | | |
| Más de 8 | | | | |

3. ¿Qué tiempo emplea en la gestión del diagnóstico del paciente?

| Tiempo utilizado en minutos | Ingresar | Modificar | Eliminar | Consultar |
|-----------------------------|----------|-----------|----------|-----------|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |
| Más de 10 | | | | |

4. ¿Qué tiempo emplea en la gestión del tratamiento del paciente?

| Tiempo utilizado en minutos | Ingresar | Modificar | Eliminar | Listar |
|-----------------------------|----------|-----------|----------|--------|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| Más de 10 | | | | |

5. ¿Qué tiempo emplea en la generación de cualquier reporte?

| Tiempo utilizado en minutos | Ingresar | Modificar | Eliminar | Listar |
|-----------------------------|----------|-----------|----------|--------|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| Más de 10 | | | | |

Anexo F: Capturas con Jhawk

Paquete Bo

Jhawk 6 - The Java Metric Tool (DEMO version)

Help | Jhawk Demo | Welcome | Select Files | Results | Export | Preferences

Dashboard | System | Classes by package | Methods by class

Packages

spring.java.bo

Classes

| Name | No. Methods | LCOM | AVCC | NOS | HBUG | HEFF | UWCS | INST | PACK | RPC | CBO | MI | CCML | NLOC |
|------------------|-------------|------|------|-----|------|----------|------|------|------|-----|-----|--------|------|------|
| DiagnosticcompBo | 6 | 0.20 | 1.00 | 14 | 0.11 | 1232.14 | 7 | 1 | 4 | 6 | 0 | 141.56 | 0 | 29 |
| OdontogramaBo | 2 | 0.00 | 1.00 | 13 | 0.11 | 1240.10 | 12 | 1 | 1 | 12 | 0 | 144.49 | 0 | 13 |
| PacienteBo | 10 | 0.00 | 1.00 | 11 | 1.18 | 22730.10 | 10 | 0 | 2 | 10 | 1 | 144.57 | 1 | 12 |
| PacienteimpBo | 12 | 0.09 | 1.00 | 26 | 0.22 | 26772.87 | 13 | 1 | 4 | 12 | 1 | 143.28 | 0 | 55 |
| TratamientoimpBo | 7 | 0.17 | 1.00 | 16 | 0.13 | 1540.84 | 8 | 1 | 4 | 7 | 0 | 141.32 | 0 | 34 |

spring.java.bo

Exit

Jhawk 6 - The Java Metric Tool (DEMO version)

Help | Jhawk Demo | Welcome | Select Files | Results | Export | Preferences

Dashboard | System | Classes by package | Methods by class

Packages

spring.java.bo

Classes

- DiagnosticcompBo
- OdontogramaBo
- PacienteBo
- PacienteimpBo
- TratamientoimpBo

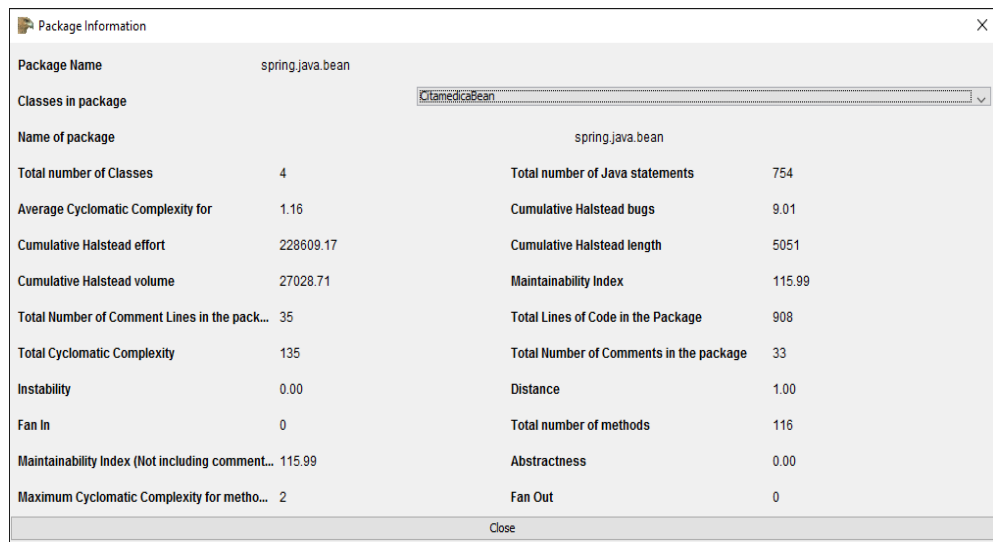
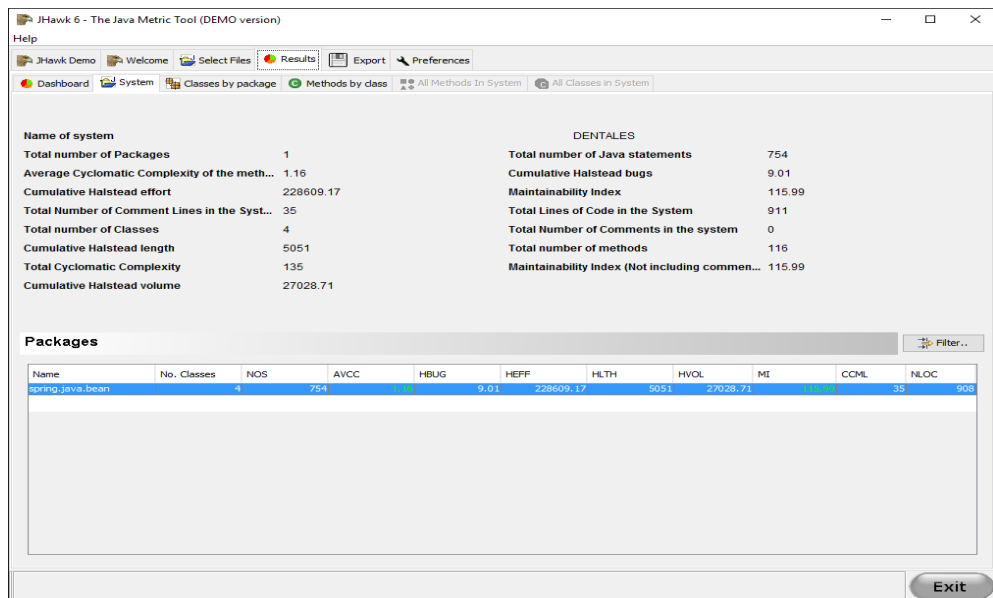
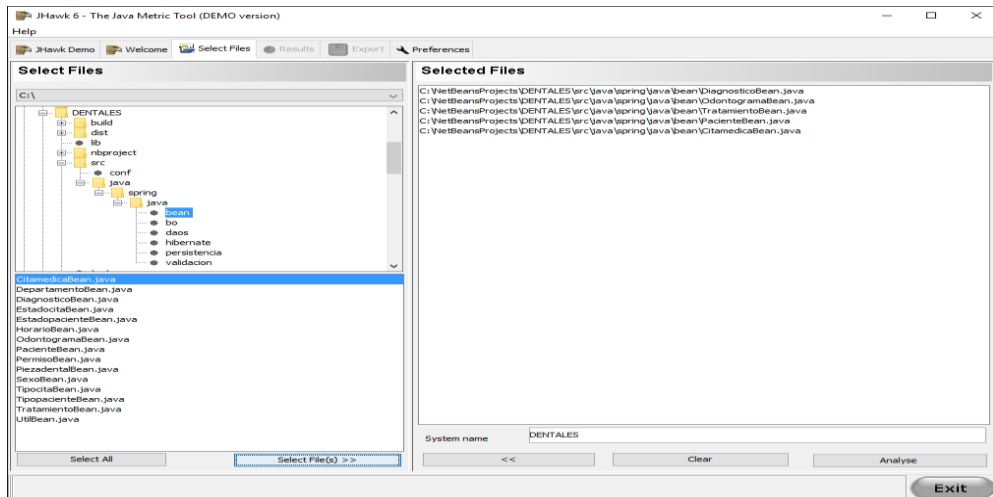
Methods

| Name | COMP | NOCL | NOS | HLTH | HVOC | HEFF | HBUG | CREP | XMET | LMET | NLOC |
|--|------|------|-----|------|------|--------|------|------|------|------|------|
| actualizar (spring.java.persistencia.Persona) (spring.java.bo.PacienteimpBo) | 1 | 0 | 2 | 17 | 13 | 242.64 | 0.02 | 1 | 1 | 0 | 3 |
| eliminar (spring.java.persistencia.Persona) (spring.java.bo.PacienteimpBo) | 1 | 0 | 2 | 17 | 13 | 242.64 | 0.02 | 1 | 1 | 0 | 3 |
| getPacienteDao() (spring.java.bo.PacienteimpBo) | 1 | 0 | 2 | 8 | 8 | 50.00 | 0.01 | 1 | 0 | 0 | 3 |
| insertar (spring.java.persistencia.Persona) (spring.java.bo.PacienteimpBo) | 1 | 0 | 2 | 17 | 13 | 242.64 | 0.02 | 1 | 1 | 0 | 3 |
| listarcompacientesodonto() (spring.java.bo.PacienteimpBo) | 1 | 0 | 2 | 13 | 11 | 196.76 | 0.01 | 1 | 1 | 0 | 3 |
| listar() (spring.java.bo.PacienteimpBo) | 1 | 0 | 2 | 15 | 12 | 225.85 | 0.02 | 1 | 1 | 0 | 3 |
| listarcombo() (spring.java.bo.PacienteimpBo) | 1 | 0 | 2 | 15 | 12 | 225.85 | 0.02 | 1 | 1 | 0 | 3 |
| listarcompacientetratamiento() (spring.java.bo.PacienteimpBo) | 1 | 0 | 2 | 13 | 11 | 196.76 | 0.01 | 1 | 1 | 0 | 3 |
| login (spring.java.persistencia.Persona) (spring.java.bo.PacienteimpBo) | 1 | 0 | 2 | 16 | 12 | 321.21 | 0.02 | 1 | 1 | 0 | 3 |
| obtenerdatos(java.lang.String) (spring.java.bo.PacienteimpBo) | 1 | 0 | 2 | 18 | 14 | 308.40 | 0.02 | 2 | 1 | 0 | 3 |
| setPacienteDao(PacienteimpDao) (spring.java.bo.PacienteimpBo) | 1 | 0 | 2 | 13 | 11 | 188.88 | 0.01 | 1 | 0 | 0 | 3 |
| totalpacientes() (spring.java.bo.PacienteimpBo) | 1 | 0 | 2 | 13 | 11 | 196.76 | 0.01 | 1 | 1 | 0 | 3 |

spring.java.bo.PacienteimpBo

Exit

Paquete Bean



JHawk 6 - The Java Metric Tool (DEMO version)

Help | JHawk Demo | Welcome | Select Files | Results | Export | Preferences

Dashboard | System | Classes by package | Methods by class | All Methods In System | All Classes in System

Packages

spring.java.bean

Classes

| Name | No. Methods | LCOM | AVCC | NOS | HBUG | HEFF | UWCS | INST | PACK | RFC | CBO | MI | CCML | NLOC |
|-----------------|-------------|------|------|-----|------|-----------|------|------|------|-----|-----|--------|------|------|
| CitamedicaBean | 38 | 0.04 | 1.18 | 196 | 2.56 | 57584.58 | 51 | 13 | 31 | 39 | 0 | 119.84 | 6 | 250 |
| DiagnosticoBean | 13 | 0.12 | 1.31 | 79 | 1.20 | 30667.60 | 17 | 4 | 29 | 13 | 0 | 115.35 | 5 | 98 |
| OdontogramaBean | 47 | 0.02 | 1.05 | 260 | 3.95 | 107695.61 | 67 | 20 | 32 | 48 | 0 | 115.85 | 18 | 317 |
| TratamientoBean | 94 | 0.09 | 1.22 | 94 | 1.30 | 32661.38 | 24 | 6 | 29 | 18 | 0 | 119.22 | 6 | 118 |

spring.java.bean

Exit

JHawk 6 - The Java Metric Tool (DEMO version)

Help | JHawk Demo | Welcome | Select Files | Results | Export | Preferences

Dashboard | System | Classes by package | Methods by class | All Methods In System | All Classes in System

Packages

spring.java.bean

Classes

CitamedicaBean
DiagnosticoBean
OdontogramaBean
TratamientoBean

Methods

| Name | COMP | NOCL | NOS | HLTH | HVOC | HEFF | HBUG | CREP | XMET | LMET | NLOC |
|---|------|------|-----|------|------|----------|------|------|------|------|------|
| TratamientoBean() (spring.java.bean.TratamientoBean) | 1 | 0 | 2 | 10 | 9 | 95.10 | 0.01 | 3 | 1 | 0 | 3 |
| actualizar() (spring.java.bean.TratamientoBean) | 2 | 0 | 6 | 51 | 24 | 1344.54 | 0.08 | 3 | 4 | 1 | 9 |
| eliminar() (spring.java.bean.TratamientoBean) | 2 | 0 | 7 | 57 | 26 | 1843.95 | 0.09 | 5 | 4 | 2 | 10 |
| getCombotratamentos() (spring.java.bean.TratamientoBean) | 1 | 0 | 10 | 62 | 27 | 3115.40 | 0.10 | 4 | 6 | 1 | 9 |
| getDescription() (spring.java.bean.TratamientoBean) | 1 | 0 | 2 | 8 | 8 | 60.00 | 0.01 | 1 | 0 | 0 | 3 |
| getIdTratamiento() (spring.java.bean.TratamientoBean) | 1 | 0 | 2 | 8 | 8 | 60.00 | 0.01 | 1 | 0 | 0 | 3 |
| getItratamiento() (spring.java.bean.TratamientoBean) | 1 | 1 | 2 | 11 | 11 | 133.19 | 0.01 | 2 | 0 | 0 | 3 |
| getSelecttratamiento() (spring.java.bean.TratamientoBean) | 1 | 0 | 2 | 8 | 8 | 60.00 | 0.01 | 1 | 0 | 0 | 3 |
| getTratamiento() (spring.java.bean.TratamientoBean) | 1 | 1 | 2 | 8 | 8 | 60.00 | 0.01 | 1 | 0 | 0 | 3 |
| insertar() (spring.java.bean.TratamientoBean) | 2 | 0 | 10 | 84 | 34 | 3679.90 | 0.14 | 5 | 8 | 4 | 13 |
| limpiar() (spring.java.bean.TratamientoBean) | 1 | 0 | 2 | 9 | 8 | 54.00 | 0.01 | 0 | 0 | 1 | 3 |
| listar() (spring.java.bean.TratamientoBean) | 1 | 0 | 2 | 15 | 11 | 186.81 | 0.02 | 0 | 1 | 2 | 3 |
| preIniciador() (java.lang.Object) (spring.java.bean.TratamientoBean) | 1 | 3 | 28 | 315 | 92 | 20549.22 | 0.68 | 22 | 26 | 0 | 29 |
| setDescription() (java.lang.String) (spring.java.bean.TratamientoBean) | 1 | 0 | 2 | 13 | 11 | 188.88 | 0.01 | 1 | 0 | 0 | 3 |
| setIdTratamiento() (java.lang.Integer) (spring.java.bean.TratamientoBean) | 1 | 0 | 2 | 13 | 11 | 188.88 | 0.01 | 1 | 0 | 0 | 3 |
| setItratamiento() (java.util.List) (spring.java.bean.TratamientoBean) | 1 | 0 | 2 | 16 | 14 | 324.89 | 0.02 | 2 | 0 | 0 | 3 |
| setSelecttratamiento() (spring.java.bean.TratamientoBean) | 1 | 0 | 2 | 13 | 11 | 188.88 | 0.01 | 1 | 0 | 0 | 3 |
| setTratamiento() (spring.java.bean.TratamientoBean) | 1 | 0 | 2 | 13 | 11 | 188.88 | 0.01 | 1 | 0 | 0 | 3 |

spring.java.bean.TratamientoBean

Exit

Paquete Dao

JHawk 6 - The Java Metric Tool (DEMO version)

Help | JHawk Demo | Welcome | Select Files | Results | Export | Preferences

Select Files

C:\

- DENTALES
 - build
 - dst
 - lib
 - nbproject
 - conf
 - spring
 - bean
 - bo
 - dao
 - hibernate
 - persistencia
 - validacion
 - test
 - web

Selected Files

- C:\NetBeansProjects\DENTALES\src\java\spring\java\dao\CitamedicaDao.java
- C:\NetBeansProjects\DENTALES\src\java\spring\java\dao\OdontogramaDao.java
- C:\NetBeansProjects\DENTALES\src\java\spring\java\dao\TratamientoDao.java
- C:\NetBeansProjects\DENTALES\src\java\spring\java\dao\DiagnosticoDao.java

System name: DENTALES

<< Clear Analyse

Exit

J-Hawk 6 - The Java Metric Tool (DEMO version)

Help

J-Hawk Demo Welcome Select Files Results Export Preferences

Dashboard System Classes by package Methods by class All Methods In System All Classes in System

Name of system DENTALES

| | | | |
|---|----------|---|--------|
| Total number of Packages | 1 | Total number of Java statements | 111 |
| Average Cyclomatic Complexity of the meth... | 1.02 | Cumulative Halstead bugs | 1.85 |
| Cumulative Halstead effort | 34304.65 | Maintainability Index | 136.67 |
| Total Number of Comment Lines in the Syst... | 2 | Total Lines of Code in the System | 186 |
| Total number of Classes | 5 | Total Number of Comments in the system | 0 |
| Cumulative Halstead length | 1301 | Total number of methods | 40 |
| Total Cyclomatic Complexity | 41 | Maintainability Index (Not including commen... | 136.67 |
| Cumulative Halstead volume | 5555.72 | | |

Packages Filter..

| Name | No. Classes | NOS | AVCC | HBUG | HEFF | HLTH | HVOL | MI | CCML | NLOC |
|------------------|-------------|-----|------|------|----------|------|---------|--------|------|------|
| spring.java.daos | 5 | 111 | 1.02 | 1.85 | 34304.65 | 1301 | 5555.72 | 136.67 | 2 | 186 |

Exit

Package Information

Package Name spring.java.daos

Classes in package CitamedicaDao

Name of package spring.java.daos

| | | | |
|---|----------|--|--------|
| Total number of Classes | 5 | Total number of Java statements | 111 |
| Average Cyclomatic Complexity for | 1.02 | Cumulative Halstead bugs | 1.85 |
| Cumulative Halstead effort | 34304.65 | Cumulative Halstead length | 1301 |
| Cumulative Halstead volume | 5555.72 | Maintainability Index | 136.67 |
| Total Number of Comment Lines in the pack... | 2 | Total Lines of Code in the Package | 186 |
| Total Cyclomatic Complexity | 41 | Total Number of Comments in the package | 10 |
| Instability | 0.00 | Distance | 0.80 |
| Fan In | 0 | Total number of methods | 40 |
| Maintainability Index (Not including commen... | 136.67 | Abstractness | 0.20 |
| Maximum Cyclomatic Complexity for metho... | 2 | Fan Out | 0 |

Close

J-Hawk 6 - The Java Metric Tool (DEMO version)

Help

J-Hawk Demo Welcome Select Files Results Export Preferences

Dashboard System Classes by package Methods by class All Methods In System All Classes in System

Packages Filter..

spring.java.daos

Classes Filter..

| Name | No. Methods | LCOM | AVCC | NOS | HBUG | HEFF | UWCS | INST | PACK | RFC | CBO | MI | CCML | NLOC |
|-----------------------|-------------|------|------|-----|------|----------|------|------|------|-----|-----|--------|------|------|
| CitamedicaDao | 9 | 0.00 | 1.00 | 10 | 0.99 | 21410.08 | 9 | 0 | 2 | 9 | 0 | 144.41 | 1 | 11 |
| DiagnosticcompDao | 4 | 0.00 | 1.00 | 9 | 0.08 | 826.52 | 4 | 0 | 3 | 4 | 0 | 141.54 | 0 | 18 |
| OrdenIntegracionesDao | 12 | 0.00 | 1.10 | 35 | 0.59 | 6506.93 | 12 | 0 | 4 | 12 | 0 | 198.56 | 1 | 27 |
| PacienteimpDao | 10 | 0.00 | 1.10 | 25 | 0.27 | 4088.48 | 10 | 0 | 3 | 10 | 0 | 139.18 | 0 | 47 |
| TratamientoimpDao | 5 | 0.00 | 1.00 | 12 | 0.12 | 1472.63 | 5 | 0 | 3 | 5 | 0 | 140.33 | 0 | 23 |

spring.java.daos

Exit

JHawk 6 - The Java Metric Tool (DEMO version)

Help

JHawk Demo Welcome Select Files Results Export Preferences

Dashboard System Classes by package Methods by class All Methods In System All Classes in System

Packages

spring.java.daos

Classes

CitamedicaDao
DiagnosticImpDao
OdontogramaImpDao
PacienteImpDao
TratamientoImpDao

Methods

| Name | COMP | NOCL | NOS | HLTH | HVOC | HEFF | HBUG | CREF | XMET | LMET | NLOC |
|---|------|------|-----|------|------|---------|------|------|------|------|------|
| Listartratamiento(xpaciente(java.lang.String) (spr... | 1 | 0 | 5 | 38 | 24 | 1400.53 | 0.06 | 3 | 1 | 1 | 6 |
| Listarpaciente(java.lang.String) (spring.java.da... | 1 | 0 | 5 | 38 | 21 | 1037.68 | 0.05 | 2 | 1 | 1 | 6 |
| actualizar(spring.java.persistence.Odontograma... | 1 | 0 | 2 | 16 | 13 | 203.00 | 0.02 | 1 | 1 | 1 | 3 |
| buscarPorId(java.lang.String) (spring.java.daos... | 1 | 0 | 3 | 28 | 19 | 749.33 | 0.04 | 2 | 1 | 1 | 4 |
| eliminar(spring.java.persistence.Odontograma) (... | 1 | 0 | 2 | 16 | 13 | 203.00 | 0.02 | 1 | 1 | 1 | 3 |
| insertar(spring.java.persistence.Odontograma) (... | 1 | 0 | 2 | 16 | 13 | 203.00 | 0.02 | 1 | 1 | 1 | 3 |
| listar() (spring.java.daos.OdontogramaImpDao) | 1 | 0 | 2 | 15 | 13 | 194.27 | 0.02 | 1 | 1 | 1 | 3 |
| listarodontograma() (spring.java.daos.Odontogra... | 1 | 0 | 2 | 15 | 13 | 194.27 | 0.02 | 1 | 1 | 1 | 3 |
| listarodontogramafechas() (spring.java.daos.... | 1 | 0 | 2 | 29 | 21 | 509.51 | 0.04 | 1 | 1 | 1 | 10 |
| listartratamientos() (spring.java.daos.Odontogra... | 1 | 0 | 2 | 15 | 13 | 194.27 | 0.02 | 1 | 1 | 1 | 3 |
| listartratamientospacientes() (spring.java.daos... | 1 | 0 | 2 | 15 | 13 | 194.27 | 0.02 | 1 | 1 | 1 | 3 |
| listarpacientestratamientos(java.lang.String) (s... | 1 | 1 | 5 | 38 | 24 | 1400.53 | 0.06 | 3 | 1 | 1 | 6 |

spring.java.daos.OdontogramaImpDao

Exit

JHawk 6 - The Java Metric Tool (DEMO version)

Help

JHawk Demo Welcome Select Files Results Export Preferences

Dashboard System Classes by package Methods by class All Methods In System All Classes in System

Name of system

DENTALES

| | | | |
|---|----------|---|--------|
| Total number of Packages | 1 | Total number of Java statements | 146 |
| Average Cyclomatic Complexity of the meth... | 1.00 | Cumulative Halstead bugs | 0.93 |
| Cumulative Halstead effort | 11187.78 | Maintainability Index | 134.56 |
| Total Number of Comment Lines in the Syst... | 16 | Total Lines of Code in the System | 199 |
| Total number of Classes | 5 | Total Number of Comments in the system | 4 |
| Cumulative Halstead length | 762 | Total number of methods | 48 |
| Total Cyclomatic Complexity | 48 | Maintainability Index (Not including commen... | 134.56 |
| Cumulative Halstead volume | 2780.35 | | |

Packages


| Name | No. Classes | NOS | AVCC | HBUG | HEFF | HLTH | HVOL | MI | CCML | NLOC |
|-------------------------|-------------|-----|------|------|----------|------|---------|--------|------|------|
| spring.java.persistence | 5 | 146 | 1.00 | 0.93 | 11187.78 | 762 | 2780.35 | 134.56 | 12 | 199 |

Exit

Anexo G: Documentos

Pacientes Atendidos

DISPENSARIO ODONTOLÓGICO
REGISTRO DE PACIENTES MES DE "PRIMO 2016"



| ITEMS | FECHA | NOMBRE - APELLIDO | Nº CEDULA | DIRECCION | CARGO | ADMINISTR. | OPERATIVO | FIRMA |
|-------|------------|--------------------|--------------|-----------|-----------------------|------------|-----------|-------------|
| 1 | 02/05/2016 | Hugo Hagan | 0601632425 | DOT | Chablar #2 | | X | [Signature] |
| 2 | 02/05/2016 | Alfonso Torres | 06054347-1 | DR | Asesor Técnico | | X | [Signature] |
| 3 | 02/05/2016 | Fredy Piquero | 06021690-9 | DR | Asesor Técnico | | X | [Signature] |
| 4 | 02/05/2016 | Fernando Urquiza | 0602791744 | SOS | CRITERIO | | X | [Signature] |
| 5 | 02/05/2016 | Marcel Hernandez T | 0602762846 | SOS | Técnico Fonoaudiólogo | | X | [Signature] |
| 6 | 02/05/2016 | Dario Quiroz | 1904200319 | DON | Ingeniero | | X | [Signature] |
| 7 | 02/05/2016 | Sergio Barrios | 060326786-5 | DIC | Ingeniero SERS | | X | [Signature] |
| 8 | 02/05/2016 | Susa Sagra | 0603727298 | DR | Clasificación | | X | [Signature] |
| 9 | 02/05/2016 | Susa Sagra | 0603727298 | DR | Clasificación | | X | [Signature] |
| 10 | 03/05/16 | Guido Guajardo | 0602025678 | DOT | Chablar #2 | | X | [Signature] |
| 11 | 3-05-2016 | Milceiro Paredes | 060255119-4 | DOT | Chablar #2 | | X | [Signature] |
| 12 | 02/05/2016 | Hugo Pineda | 06017766-4 | GERENTE | Gerente | X | X | [Signature] |
| 13 | 03/05/2016 | Hector Mena | 0602960593 | DR | Hacerlo #1 | | X | [Signature] |
| 14 | 03/05/2016 | Alfonso Ortiz | 0604200321 | DR | Juquiza & Bogot | X | X | [Signature] |
| 15 | 03/05/2016 | Luis Dominguez | 0605448042 | SOS | Técnico Fonoaudiólogo | | X | [Signature] |
| 16 | 03/05/2016 | Galdo Pineda | 060215112-5 | DR | Asesor Técnico | | X | [Signature] |
| 17 | 04/05/2016 | Alfonso Luna | 0601284250 | DR | Asesor Técnico | X | X | [Signature] |
| 18 | 04/05/2016 | Sergio Hernandez | 0605071009-7 | DOT | Asesor Técnico | | X | [Signature] |
| 19 | 04/05/2016 | George Anchunda | 190810982-7 | DIC | Ingeniero | X | X | [Signature] |
| 20 | 04/05/2016 | Dario Quiroz | 1904200319 | DOT | Ingeniero | | X | [Signature] |
| 21 | 04/05/2016 | Alfonso Pineda | 0602776748 | DOT | Chablar #1 | | X | [Signature] |
| 22 | 04/05/2016 | Israel Rojas | 0603216888 | SOS | Chablar #1 | | X | [Signature] |
| 23 | 04/05/2016 | Yaneth Gaitani | 060322330-3 | DR | Asesor Técnico | | X | [Signature] |
| 24 | 04/05/2016 | Hugo Quiroz | 0605448042 | DR | Asesor Técnico | | X | [Signature] |
| 25 | 04/05/2016 | Fredy Piquero | 06021690-9 | DR | Asesor Técnico | | X | [Signature] |
| 26 | 04/05/2016 | Fredy Piquero | 06021690-9 | DR | Asesor Técnico | | X | [Signature] |

Registro de Pacientes

| Fecha de ingreso | Nombre del paciente | Historia clínica | Exámenes de laboratorio | Tratamiento | Evolution |
|------------------|---------------------|------------------|-------------------------|-------------|-----------|
| 15/05/2016 | ... | ... | ... | ... | ... |
| 16/05/2016 | ... | ... | ... | ... | ... |
| 17/05/2016 | ... | ... | ... | ... | ... |
| 18/05/2016 | ... | ... | ... | ... | ... |
| 19/05/2016 | ... | ... | ... | ... | ... |
| 20/05/2016 | ... | ... | ... | ... | ... |
| 21/05/2016 | ... | ... | ... | ... | ... |
| 22/05/2016 | ... | ... | ... | ... | ... |
| 23/05/2016 | ... | ... | ... | ... | ... |
| 24/05/2016 | ... | ... | ... | ... | ... |
| 25/05/2016 | ... | ... | ... | ... | ... |
| 26/05/2016 | ... | ... | ... | ... | ... |
| 27/05/2016 | ... | ... | ... | ... | ... |
| 28/05/2016 | ... | ... | ... | ... | ... |
| 29/05/2016 | ... | ... | ... | ... | ... |
| 30/05/2016 | ... | ... | ... | ... | ... |
| 31/05/2016 | ... | ... | ... | ... | ... |
| 01/06/2016 | ... | ... | ... | ... | ... |
| 02/06/2016 | ... | ... | ... | ... | ... |
| 03/06/2016 | ... | ... | ... | ... | ... |
| 04/06/2016 | ... | ... | ... | ... | ... |
| 05/06/2016 | ... | ... | ... | ... | ... |
| 06/06/2016 | ... | ... | ... | ... | ... |
| 07/06/2016 | ... | ... | ... | ... | ... |
| 08/06/2016 | ... | ... | ... | ... | ... |
| 09/06/2016 | ... | ... | ... | ... | ... |
| 10/06/2016 | ... | ... | ... | ... | ... |
| 11/06/2016 | ... | ... | ... | ... | ... |
| 12/06/2016 | ... | ... | ... | ... | ... |
| 13/06/2016 | ... | ... | ... | ... | ... |
| 14/06/2016 | ... | ... | ... | ... | ... |
| 15/06/2016 | ... | ... | ... | ... | ... |
| 16/06/2016 | ... | ... | ... | ... | ... |
| 17/06/2016 | ... | ... | ... | ... | ... |
| 18/06/2016 | ... | ... | ... | ... | ... |
| 19/06/2016 | ... | ... | ... | ... | ... |
| 20/06/2016 | ... | ... | ... | ... | ... |
| 21/06/2016 | ... | ... | ... | ... | ... |
| 22/06/2016 | ... | ... | ... | ... | ... |
| 23/06/2016 | ... | ... | ... | ... | ... |
| 24/06/2016 | ... | ... | ... | ... | ... |
| 25/06/2016 | ... | ... | ... | ... | ... |
| 26/06/2016 | ... | ... | ... | ... | ... |
| 27/06/2016 | ... | ... | ... | ... | ... |
| 28/06/2016 | ... | ... | ... | ... | ... |
| 29/06/2016 | ... | ... | ... | ... | ... |
| 30/06/2016 | ... | ... | ... | ... | ... |
| 01/07/2016 | ... | ... | ... | ... | ... |
| 02/07/2016 | ... | ... | ... | ... | ... |
| 03/07/2016 | ... | ... | ... | ... | ... |
| 04/07/2016 | ... | ... | ... | ... | ... |
| 05/07/2016 | ... | ... | ... | ... | ... |
| 06/07/2016 | ... | ... | ... | ... | ... |
| 07/07/2016 | ... | ... | ... | ... | ... |
| 08/07/2016 | ... | ... | ... | ... | ... |
| 09/07/2016 | ... | ... | ... | ... | ... |
| 10/07/2016 | ... | ... | ... | ... | ... |
| 11/07/2016 | ... | ... | ... | ... | ... |
| 12/07/2016 | ... | ... | ... | ... | ... |
| 13/07/2016 | ... | ... | ... | ... | ... |
| 14/07/2016 | ... | ... | ... | ... | ... |
| 15/07/2016 | ... | ... | ... | ... | ... |
| 16/07/2016 | ... | ... | ... | ... | ... |
| 17/07/2016 | ... | ... | ... | ... | ... |
| 18/07/2016 | ... | ... | ... | ... | ... |
| 19/07/2016 | ... | ... | ... | ... | ... |
| 20/07/2016 | ... | ... | ... | ... | ... |
| 21/07/2016 | ... | ... | ... | ... | ... |
| 22/07/2016 | ... | ... | ... | ... | ... |
| 23/07/2016 | ... | ... | ... | ... | ... |
| 24/07/2016 | ... | ... | ... | ... | ... |
| 25/07/2016 | ... | ... | ... | ... | ... |
| 26/07/2016 | ... | ... | ... | ... | ... |
| 27/07/2016 | ... | ... | ... | ... | ... |
| 28/07/2016 | ... | ... | ... | ... | ... |
| 29/07/2016 | ... | ... | ... | ... | ... |
| 30/07/2016 | ... | ... | ... | ... | ... |
| 31/07/2016 | ... | ... | ... | ... | ... |
| 01/08/2016 | ... | ... | ... | ... | ... |
| 02/08/2016 | ... | ... | ... | ... | ... |
| 03/08/2016 | ... | ... | ... | ... | ... |
| 04/08/2016 | ... | ... | ... | ... | ... |
| 05/08/2016 | ... | ... | ... | ... | ... |
| 06/08/2016 | ... | ... | ... | ... | ... |
| 07/08/2016 | ... | ... | ... | ... | ... |
| 08/08/2016 | ... | ... | ... | ... | ... |
| 09/08/2016 | ... | ... | ... | ... | ... |
| 10/08/2016 | ... | ... | ... | ... | ... |
| 11/08/2016 | ... | ... | ... | ... | ... |
| 12/08/2016 | ... | ... | ... | ... | ... |
| 13/08/2016 | ... | ... | ... | ... | ... |
| 14/08/2016 | ... | ... | ... | ... | ... |
| 15/08/2016 | ... | ... | ... | ... | ... |
| 16/08/2016 | ... | ... | ... | ... | ... |
| 17/08/2016 | ... | ... | ... | ... | ... |
| 18/08/2016 | ... | ... | ... | ... | ... |
| 19/08/2016 | ... | ... | ... | ... | ... |
| 20/08/2016 | ... | ... | ... | ... | ... |
| 21/08/2016 | ... | ... | ... | ... | ... |
| 22/08/2016 | ... | ... | ... | ... | ... |
| 23/08/2016 | ... | ... | ... | ... | ... |
| 24/08/2016 | ... | ... | ... | ... | ... |
| 25/08/2016 | ... | ... | ... | ... | ... |
| 26/08/2016 | ... | ... | ... | ... | ... |
| 27/08/2016 | ... | ... | ... | ... | ... |
| 28/08/2016 | ... | ... | ... | ... | ... |
| 29/08/2016 | ... | ... | ... | ... | ... |
| 30/08/2016 | ... | ... | ... | ... | ... |
| 31/08/2016 | ... | ... | ... | ... | ... |
| 01/09/2016 | ... | ... | ... | ... | ... |
| 02/09/2016 | ... | ... | ... | ... | ... |
| 03/09/2016 | ... | ... | ... | ... | ... |
| 04/09/2016 | ... | ... | ... | ... | ... |
| 05/09/2016 | ... | ... | ... | ... | ... |
| 06/09/2016 | ... | ... | ... | ... | ... |
| 07/09/2016 | ... | ... | ... | ... | ... |
| 08/09/2016 | ... | ... | ... | ... | ... |
| 09/09/2016 | ... | ... | ... | ... | ... |
| 10/09/2016 | ... | ... | ... | ... | ... |
| 11/09/2016 | ... | ... | ... | ... | ... |
| 12/09/2016 | ... | ... | ... | ... | ... |
| 13/09/2016 | ... | ... | ... | ... | ... |
| 14/09/2016 | ... | ... | ... | ... | ... |
| 15/09/2016 | ... | ... | ... | ... | ... |
| 16/09/2016 | ... | ... | ... | ... | ... |
| 17/09/2016 | ... | ... | ... | ... | ... |
| 18/09/2016 | ... | ... | ... | ... | ... |
| 19/09/2016 | ... | ... | ... | ... | ... |
| 20/09/2016 | ... | ... | ... | ... | ... |
| 21/09/2016 | ... | ... | ... | ... | ... |
| 22/09/2016 | ... | ... | ... | ... | ... |
| 23/09/2016 | ... | ... | ... | ... | ... |
| 24/09/2016 | ... | ... | ... | ... | ... |
| 25/09/2016 | ... | ... | ... | ... | ... |
| 26/09/2016 | ... | ... | ... | ... | ... |
| 27/09/2016 | ... | ... | ... | ... | ... |
| 28/09/2016 | ... | ... | ... | ... | ... |
| 29/09/2016 | ... | ... | ... | ... | ... |
| 30/09/2016 | ... | ... | ... | ... | ... |
| 01/10/2016 | ... | ... | ... | ... | ... |
| 02/10/2016 | ... | ... | ... | ... | ... |
| 03/10/2016 | ... | ... | ... | ... | ... |
| 04/10/2016 | ... | ... | ... | ... | ... |
| 05/10/2016 | ... | ... | ... | ... | ... |
| 06/10/2016 | ... | ... | ... | ... | ... |
| 07/10/2016 | ... | ... | ... | ... | ... |
| 08/10/2016 | ... | ... | ... | ... | ... |
| 09/10/2016 | ... | ... | ... | ... | ... |
| 10/10/2016 | ... | ... | ... | ... | ... |
| 11/10/2016 | ... | ... | ... | ... | ... |
| 12/10/2016 | ... | ... | ... | ... | ... |
| 13/10/2016 | ... | ... | ... | ... | ... |
| 14/10/2016 | ... | ... | ... | ... | ... |
| 15/10/2016 | ... | ... | ... | ... | ... |
| 16/10/2016 | ... | ... | ... | ... | ... |
| 17/10/2016 | ... | ... | ... | ... | ... |
| 18/10/2016 | ... | ... | ... | ... | ... |
| 19/10/2016 | ... | ... | ... | ... | ... |
| 20/10/2016 | ... | ... | ... | ... | ... |
| 21/10/2016 | ... | ... | ... | ... | ... |
| 22/10/2016 | ... | ... | ... | ... | ... |
| 23/10/2016 | ... | ... | ... | ... | ... |
| 24/10/2016 | ... | ... | ... | ... | ... |
| 25/10/2016 | ... | ... | ... | ... | ... |
| 26/10/2016 | ... | ... | ... | ... | ... |
| 27/10/2016 | ... | ... | ... | ... | ... |
| 28/10/2016 | ... | ... | ... | ... | ... |
| 29/10/2016 | ... | ... | ... | ... | ... |
| 30/10/2016 | ... | ... | ... | ... | ... |
| 31/10/2016 | ... | ... | ... | ... | ... |
| 01/11/2016 | ... | ... | ... | ... | ... |
| 02/11/2016 | ... | ... | ... | ... | ... |
| 03/11/2016 | ... | ... | ... | ... | ... |
| 04/11/2016 | ... | ... | ... | ... | ... |
| 05/11/2016 | ... | ... | ... | ... | ... |
| 06/11/2016 | ... | ... | ... | ... | ... |
| 07/11/2016 | ... | ... | ... | ... | ... |
| 08/11/2016 | ... | ... | ... | ... | ... |
| 09/11/2016 | ... | ... | ... | ... | ... |
| 10/11/2016 | ... | ... | ... | ... | ... |
| 11/11/2016 | ... | ... | ... | ... | ... |
| 12/11/2016 | ... | ... | ... | ... | ... |
| 13/11/2016 | ... | ... | ... | ... | ... |
| 14/11/2016 | ... | ... | ... | ... | ... |
| 15/11/2016 | ... | ... | ... | ... | ... |
| 16/11/2016 | ... | ... | ... | ... | ... |
| 17/11/2016 | ... | ... | ... | ... | ... |
| 18/11/2016 | ... | ... | ... | ... | ... |
| 19/11/2016 | ... | ... | ... | ... | ... |
| 20/11/2016 | ... | ... | ... | ... | ... |
| 21/11/2016 | ... | ... | ... | ... | ... |
| 22/11/2016 | ... | ... | ... | ... | ... |
| 23/11/2016 | ... | ... | ... | ... | ... |
| 24/11/2016 | ... | ... | ... | ... | ... |
| 25/11/2016 | ... | ... | ... | ... | ... |
| 26/11/2016 | ... | ... | ... | ... | ... |
| 27/11/2016 | ... | ... | ... | ... | ... |
| 28/11/2016 | ... | ... | ... | ... | ... |
| 29/11/2016 | ... | ... | ... | ... | ... |
| 30/11/2016 | ... | ... | ... | ... | ... |
| 31/11/2016 | ... | ... | ... | ... | ... |
| 01/12/2016 | ... | ... | ... | ... | ... |
| 02/12/2016 | ... | ... | ... | ... | ... |
| 03/12/2016 | ... | ... | ... | ... | ... |
| 04/12/2016 | ... | ... | ... | ... | ... |
| 05/12/2016 | ... | ... | ... | ... | ... |
| 06/12/2016 | ... | ... | ... | ... | ... |
| 07/12/2016 | ... | ... | ... | ... | ... |
| 08/12/2016 | ... | ... | ... | ... | ... |
| 09/12/2016 | ... | ... | ... | ... | ... |
| 10/12/2016 | ... | ... | ... | ... | ... |
| 11/12/2016 | ... | ... | ... | ... | ... |
| 12/12/2016 | ... | ... | ... | ... | ... |
| 13/12/2016 | ... | ... | ... | ... | ... |
| 14/12/2016 | ... | ... | ... | ... | ... |
| 15/12/2016 | ... | ... | ... | ... | ... |
| 16/12/2016 | ... | ... | ... | ... | ... |
| 17/12/2016 | ... | ... | ... | ... | ... |
| 18/12/2016 | ... | ... | ... | ... | ... |
| 19/12/2016 | ... | ... | ... | ... | ... |
| 20/12/2016 | ... | ... | ... | ... | ... |
| 21/12/2016 | ... | ... | ... | ... | ... |
| 22/12/2016 | ... | ... | ... | ... | ... |
| 23/12/2016 | ... | ... | ... | ... | ... |
| 24/12/2016 | ... | ... | ... | ... | ... |
| 25/12/2016 | ... | ... | ... | ... | ... |
| 26/12/2016 | ... | ... | ... | ... | ... |
| 27/12/2016 | ... | ... | ... | ... | ... |
| 28/12/2016 | ... | ... | ... | ... | ... |
| 29/12/2016 | ... | ... | ... | ... | ... |
| 30/12/2016 | ... | ... | ... | ... | |

EERSA
DEPARTAMENTO ODONTOLÓGICO



MANUAL
DE
USUARIO

SISTEMA WEB DE GESTIÓN
ODONTOLÓGICO

Introducción

El Aplicación Web de Gestión Odontológico, es un sistema diseñado para la gestión de información de los pacientes como parte de las actividades diarias que se realizan en el Departamento Odontológico de la EERSA. Este documento tiene como propósito guiar al usuario en las distintas funcionalidades que brinda la aplicación.

Roles de Usuarios

El sistema web consta de los siguientes roles de usuario que se describen a continuación.

- Rol Administrador
- Rol Dentista
- Rol Asistente
- Rol Paciente

Estos perfiles son gestionados por el usuario Administrador del sistema.

Para el acceso al sistema se puede utilizar cualquier navegador de internet de preferencia (Google Chrome).

Ingreso al Sistema

Para la el ingreso al sistema el usuario debe autenticarse ingresando los datos como: Usuario y Contraseña, posteriormente el sistema validará los datos ingresados, de ser correctos el sistema ingresara a la página principal y autorizará las funcionalidades de acuerdo a cada rol, en el caso de ingresar los datos incorrectos el sistema mostrará un mensaje de error “Usuario o contraseña incorrecta”.



Rol Paciente

Posteriormente a la autenticación con el rol paciente se visualizará la pantalla principal con opciones preestablecidas para este rol como se muestra en la siguiente figura.



Cita medica

Para ser atendido por la/el dentista el paciente deberá realizar (ingresar) una cita médica, la misma que especificará la fecha y hora el tipo de consulta: primera consulta o consulta subsecuente de ser el caso de la misma manera podrá especificar el motivo de la consulta. Los campos con (*) son obligatorios.

REALIZAR CITAS

Fecha y Hora

FECHA DE LA CITA: *

| Mayo 2016 | | | | | | | Junio 2016 | | | | | | |
|-----------|----|----|----|----|----|----|------------|----|----|----|----|----|----|
| L | M | Mi | J | V | S | D | L | M | Mi | J | V | S | D |
| | | | | | | 1 | 1 | 2 | 3 | 4 | 5 | | |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 | 27 | 28 | 29 | 30 | | | |
| 30 | 31 | | | | | | | | | | | | |

HORA: * - Seleccione -

Paciente

| | |
|---------------------|---|
| PACIENTE: | ROSERO LÓPEZ MAYRA ALEXANDRA |
| TIPO DE CONSULTA: * | PRIMERA CONSULTA |
| MOTIVO: | <div style="border: 1px solid #ccc; height: 20px; width: 100%;"></div> |

800 Caracteres

GUARDAR
CANCELAR

Citas pendientes

El paciente podrá verificar las citas médicas ingresadas, y de igual manera puede modificar y eliminar dichas citas.

CITAS PENDIENTES

Exportar Datos de la Tabla

BUSCAR:

Pág: 1 de 1 Ir Pág: 1 Filas/Pág: 5

| CEDULA | NOMBRE | FECHA CITA | HORA | TIPO DE CONSULTA | MOTIVO | ESTADO | MODIFICAR | ELIMINAR |
|------------|------------------------------|------------|-------------|----------------------|--------|-----------|-----------|----------|
| 1722431028 | ROSERO LÓPEZ MAYRA ALEXANDRA | 2016-06-02 | 10:30-11:00 | CONSULTA SUBSECUENTE | Dolor | PENDIENTE | | |
| 1722431028 | ROSERO LÓPEZ MAYRA ALEXANDRA | 2016-06-01 | 9:30-10:00 | PRIMERA CONSULTA | | PENDIENTE | | |

DEPARTAMENTO ODONTOLÓGICO

Exportar Todos Los Datos

Citas Atendidas

Al dirigirse a esta opción “**Citas Atendidas**” se desplegará todas las citas fueron atendidas por el dentista, como se muestra en la siguiente figura.

CITAS ATENDIDAS

Exportar Datos de la Tabla

BUSCAR:

Pág: 1 de 1 Ir Pág: 1 Filas/Pág: 5

| CEDULA | PACIENTE | FECHA DE LA CITA | HORA | TIPO CONSULTA | ESTADO | FECHA ATENDIDA | VER |
|------------|------------------------------|------------------|-------------|----------------------|----------|----------------|-----|
| 1722431028 | ROSERO LÓPEZ MAYRA ALEXANDRA | 2016-05-05 | 10:30-11:00 | PRIMERA CONSULTA | ATENDIDA | 2016-05-31 | |
| 1722431028 | ROSERO LÓPEZ MAYRA ALEXANDRA | 2016-04-01 | 16:00-16:30 | CONSULTA SUBSECUENTE | ATENDIDA | 2016-05-31 | |
| 1722431028 | ROSERO LÓPEZ MAYRA ALEXANDRA | 2016-03-02 | 15:30-16:00 | CONSULTA SUBSECUENTE | ATENDIDA | 2016-05-31 | |
| 1722431028 | ROSERO LÓPEZ MAYRA ALEXANDRA | 2016-02-03 | 9:30-10:00 | CONSULTA SUBSECUENTE | ATENDIDA | 2016-05-31 | |

DEPARTAMENTO ODONTOLÓGICO

Exportar Todos Los Datos

Diagnostico dental

En esta opción el paciente podrá tener un detalle del diagnóstico que se le realizó cuando asistió por primera vez al consultorio, podemos observar en la siguiente figura.

DIAGNÓSTICO DENTAL

| PACIENTE | | | | |
|------------|------------------------------|------------|-----------------------|------|
| CÉDULA | NOMBRE | 2016-05-28 | FECHA FIN DE CONTRATO | EDAD |
| 1722431028 | ROSETO LOPEZ MAYRA ALEXANDRA | | | 27 |

| #PIEZA DENTAL | DIAGNOSTICO | OBSERVACIONES | FECHA DEL DIAGNOSTICO |
|---------------|----------------------|---------------|-----------------------|
| 11 | [PROTESIS FIJA] | | 2016-05-29 |
| 25 | [PROTESIS REMOVIBLE] | MAL UBICADA | 2016-05-29 |

ENFERMEDAD PERIODONTAL

| MATERIA ALBA | PLACA BACTERIANA | C. SUPRAGINGIVAL | C. SUBGINGIVAL | BOLSA PERIODONTAL | MOVILIDAD DENTAL | OBSERVACIONES | FECHA DE REGISTRO |
|--------------|------------------|------------------|----------------|-------------------|------------------|---------------|-------------------|
| No | Si | No | Si | Si | Si | | 2016-05-29 |

OCCLUSIÓN

| MALOCCLUSION DENTARIA | MALOCCLUSION MAXILAR | OBSERVACIONES | FECHA DE REGISTRO |
|-----------------------|----------------------|---------------|-------------------|
| No | Si | | 2016-05-29 |

Tratamiento dental

En esta opción el paciente podrá ir observando todos los tratamientos que se le aplicaran conforme asista al consultorio, como se puede observar en la siguiente figura.

TRATAMIENTOS APLICADOS

Exportar Datos de la Tabla

| EVALUACION DENTAL | | | | | | | | | | | |
|-------------------|----|----------|------------|-----------|------------|--------------|----|-------|-------|---------------|----------------|
| #PIEZA DENTAL | CI | CURACION | ENDODONCIA | EXODONCIA | PROFILAXIS | RESTAURACION | RX | TFLOR | OTROS | OBSERVACIONES | FECHA ATENDIDA |
| 11 | | CURACION | | EXODONCIA | | | | | | BEN CURADA | 2016-05-31 |
| 11 | | CURACION | | EXODONCIA | | | | TFLOR | | BEN REALIZADA | 2016-05-31 |
| 11 | | CURACION | ENDODONCIA | | | | | | | BEN REALIZADA | 2016-05-31 |
| 11 | CI | | | | | | | | | | 2016-05-31 |
| 11 | | | | | | | | TFLOR | | | 2016-05-31 |
| 11 | | | ENDODONCIA | | | | | | | | 2016-05-31 |

DEPARTAMENTO ODONTOLÓGICO

Rol Dentista


Posterior a la autenticación con el siguiente rol se mostrará la pantalla principal con las siguientes opciones como se puede observar en la siguiente figura.



Citas médicas

La odontóloga podrá asignar (ingresar) una cita médica a cualquier paciente la misma que tiene como datos obligatorios la fecha de la cita, la hora, el nombre del paciente, el tipo de consulta, y un motivo, como se puede observar en la siguiente figura.

Citas pendientes


La odontóloga podrá listar todas las citas pendientes de todos los pacientes, de igual manera puede modificar, eliminar cualquier cita médica. En la siguiente pantalla se puede observar el listado de las citas pendientes para este rol. Cuando existen citas pendientes en la parte superior izquierda aparecerá un aviso como el siguiente  **Citas Programadas** en el cual se puede dar click para mostrar las citas pendientes.

| CEDULA | NOMBRE | FECHA CITA | HORA | TIPO DE CONSULTA | MOTIVO | ESTADO | MODIFICAR | ELIMINAR |
|------------|-------------------------------|------------|-------------|----------------------|-----------------------|-----------|-----------|----------|
| 020182073 | NEIRE JON DARWIN | 2016-06-01 | 9:00-9:30 | PRIMERA CONSULTA | | PENDIENTE | | |
| 1723109516 | PONLUIZA HORTA JESUS MESIAS | 2016-06-01 | 9:30-10:00 | PRIMERA CONSULTA | | PENDIENTE | | |
| 1722431028 | ROSEIRO LÓPEZ MAYRA ALEXANDRA | 2016-06-01 | 9:30-10:00 | PRIMERA CONSULTA | | PENDIENTE | | |
| 1722431028 | ROSEIRO LÓPEZ MAYRA ALEXANDRA | 2016-06-02 | 10:30-11:00 | CONSULTA SUBSECUENTE | SACARME UNA MUELA | PENDIENTE | | |
| 1723109516 | PONLUIZA HORTA JESUS MESIAS | 2016-06-30 | 9:30-10:00 | CONSULTA SUBSECUENTE | CURARME POR FAVOR DOC | PENDIENTE | | |

Diagnostico dental

En el submenú denominado realizar diagnóstico, se listaran a los pacientes que se registran, y que por primera vez acuden al consultorio, la dentista se encargara de registrar el respectivo diagnóstico para cada paciente. En la siguiente pantalla podemos observar el listado de pacientes con sus respectivas funciones.

| REALIZAR DIAGNÓSTICO | | | | | | |
|----------------------|---------------------------------|------|----------------|--------------------|----------------------|--------------|
| CÉDULA | PACIENTE | EDAD | FECHA CONTRATO | FECHA FIN CONTRATO | DEPARTAMENTO | DIAGNOSTICAR |
| 0601163482 | GORDILLO VELARDE HECTOR VINICIO | 59 | 2006-01-01 | 2016-05-29 | FISCALIZACIÓN | + |
| 0201562873 | MOREJON DARWIN | 37 | 2012-09-09 | 2016-05-28 | INFORMATICA | + |
| 1723109516 | PONLUIZA HORTA JESUS MESIAS | 27 | 2015-05-01 | 2016-09-22 | INFORMATICA | + |
| 0603243221 | BONIFAZ DAMIAN ROSA CEMIRA | 37 | 2016-01-05 | 2016-05-28 | DIS Y AL PUBLICO | + |
| 1722431028 | ROSERO LÓPEZ MAYRA ALEXANDRA | 27 | 2017-05-02 | 2016-05-28 | S. GENERAL Y ARCHIVO | + |

Posteriormente debe presionar sobre la opción  donde aparecerá la siguiente pantalla donde podrá registrar el respectivo diagnóstico.

DIAGNÓSTICO DENTAL

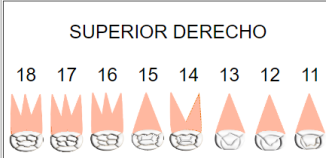
PACIENTE

| CÉDULA | NOMBRE | FECHA FIN DE CONTRATO | EDAD |
|------------|----------------|-----------------------|------|
| 0201562873 | MOREJON DARWIN | 2016-05-28 | 37 |

ODONTOGRAMA

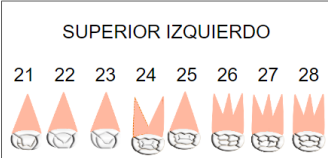
SUPERIOR DERECHO

18 17 16 15 14 13 12 11



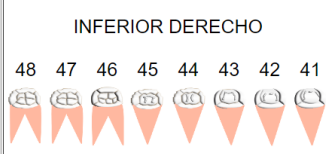
SUPERIOR IZQUIERDO

21 22 23 24 25 26 27 28



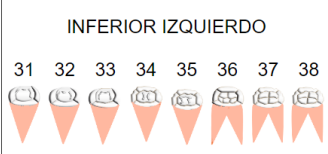
INFERIOR DERECHO


48 47 46 45 44 43 42 41



INFERIOR IZQUIERDO

31 32 33 34 35 36 37 38



Para ingresar el diagnostico debe dar click en cada pieza dental  donde aparecerá una ventana con las siguientes opciones donde podrá especificar el diagnóstico de cada pieza dental de una manera detallada.

PIEZA 11

DIAGNOSTICO: *

ENDODONCIA
 AUSENCIA
 POSTE

PROTESIS FIJA
 FRACTURA
 CORONA

PRÓTESIS REMOVIBLE
 IMPLANTE
 MOVILIDAD

OBSERVACIONES:

2000 Caracteres

Los diagnósticos se irán registrando y se mostrara el listado correspondiente.

| #PIEZA DENTAL | DIAGNÓSTICO | OBSERVACIONES | FECHA DEL DIAGNÓSTICO | ELIMINAR |
|---------------|---------------------------|--------------------------------|-----------------------|----------|
| 11 | [ENDODONCIA] | BIEN REALIZADA | 2016-05-29 | ✗ |
| 12 | [ENDODONCIA] | | 2016-05-31 | ✗ |
| 21 | [FRACTURA] | PERDIÓ EL DIENTE EN UNA PELEA | 2016-05-31 | ✗ |
| 31 | [PROTESIS FIJA, FRACTURA] | | 2016-05-30 | ✗ |
| 33 | [FRACTURA] | EN LA MITAD DE LA PIEZA DENTAL | 2016-05-29 | ✗ |
| 38 | [IMPLANTE] | MAL REALIZADO | 2016-05-29 | ✗ |

De igual manera podrá ingresar el siguiente diagnóstico.

ENFERMEDAD PERIODONTAL

| INGRESAR | | | | | | | | |
|--------------|------------------|-----------------|---------------|-------------------|------------------|---------------|-------------------|----------|
| MATERIA ALBA | PLACA BACTERIANA | C.SUPRAGINGIVAL | C.SUBGINGIVAL | BOLSA PERIODONTAL | MOVILIDAD DENTAL | OBSERVACIONES | FECHA DE REGISTRO | ELIMINAR |
| Si | Si | No | No | No | No | | 2016-05-31 | X |

OCLUSIÓN

| INGRESAR | | | | | |
|----------------------|---------------------|---------------|-------------------|----------|--|
| MALOCLUSION DENTARIA | MALOCLUSION MAXILAR | OBSERVACIONES | FECHA DE REGISTRO | ELIMINAR | |
| No | No | | 2016-05-31 | X | |


Tratamiento dental

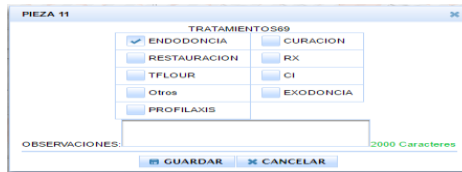
Para registrar los tratamientos que se realizan a cada paciente tendrá que dirigirse hasta el submenú Realizar tratamiento, posteriormente se visualizará una ventana donde se listan todas las citas médicas pendientes para ser atendidas por la odontóloga. Se mostrará la siguiente pantalla.

OPCIONES
Atender Cita Médica

Después debe hacer clic sobre el siguiente botón para poder atender la cita, se visualizara la siguiente ventana.

11

Debe dar click sobre la pieza dental que realizó el tratamiento  y posteriormente se abrirá una ventana como la siguiente.



Luego de realizar el tratamiento o atender la cita médica, esta cita pasará a un estado de aprobado.

De igual manera podrá eliminar el tratamiento aplicado en la opción de eliminar.

Rol Administrador

Este usuario tendrá acceso a todas las opciones detalladas anteriormente, más la administración de usuarios.

Asignación de Roles

El usuario deberá dirigirse hasta el submenú Asignar Roles, tendrá un listado de todos los usuarios del sistema, aquí podrá asignar los perfiles de acuerdo a cada usuario. Como podemos observar en la siguiente ventana.



Luego debe dar click en la siguiente opción donde se visualizará la siguiente ventana modal, donde podrá asignar el perfil correspondiente.



Asignar contraseña

En esta opción se le asignará una contraseña para el acceso al sistema, en este caso será la cédula de identidad de cada paciente, que posteriormente esta puede ser modificada por los usuarios.



De igual manera se podrá quitar la contraseña de acceso al sistema en la siguiente ventana. Donde el usuario ya no podrá ingresar al sistema.

ELIMINAR CONTRASEÑA

BUSCAR: Ingrese el Dato a Buscar

Pág: 1 de 1 Ir Pág: 1 Filas/Pág: 5

| CÉDULA | NOMBRE | APELLIDO | CONTRASEÑA | QUITAR CONTRASEÑA |
|------------|-----------------|------------------|----------------------|-------------------|
| 0601163462 | HECTOR VINICIO | GORDILLO VELARDE | 2336c7750264e08e84 | |
| 0201562873 | DARWIN | MOREJON | 2f71642373106a27334f | |
| 1723109516 | JESUS MESIAS | POHLUIZA HORTA | ec098722001e362a0f5d | |
| 0603243221 | ROSA CEMIRA | BOHIFAZ DAMIAN | 5585b5346b937670b67 | |
| 1722431028 | MAYRA ALEXANDRA | ROSERO LOPEZ | 455d210b818432297cf | |

DEPARTAMENTO ODONTOLÓGICO

La contraseña puede ser modificada después de que el usuario realice su respectivo acceso al sistema. En la siguiente ventana.

CAMBIAR DE CONTRASEÑA

Nueva Contraseña: *

Confirmar Contraseña: *

Fuerte