

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO

**“SISTEMA WEB DE ADMINISTRACIÓN, CONTROL Y GESTIÓN DE HISTORIALES
CLÍNICOS**

CASO: CONSULTORIO DENTAL ORTHO-BRAES”

**Proyecto de Grado para obtener el Título de Licenciatura en Informática Mención Ingeniería
de Sistemas Informáticos**

POR: DIEGO VLADIMIR MAMANI NINA

TUTOR: PHD. JAVIER HUGO GUSTAVO REYES PACHECO

LA PAZ – BOLIVIA

2022

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

Proyecto de grado:

**“SISTEMA WEB DE ADMINISTRACIÓN, CONTROL Y GESTIÓN DE
HISTORIALES CLÍNICOS
CASO: CONSULTORIO DENTAL ORTHO-BRAES”**

Presentado por: Univ. Diego Vladimir Mamani Nina.

Para optar por el grado académico de: Licenciada en Informática mención
Ingeniería de Sistemas Informáticos.

Nota numeral:

Nota literal:

Ha sido:

Director de la carrera de Informática: Lic. Rubén Alcón López

Tutor: Ph.D. Javier Hugo Gustavo Reyes Pacheco

Tribunal:

Tribunal:

Tribunal:



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR

DEDICATORIA

A mi madre quien fue el pilar de mi vida a pesar de todas las dificultades.

Este logro también es suyo.

A mi familia que siempre estuvo unida y me demostró que el plan de Dios es perfecto. Estamos sujetos a su voluntad.

AGRADECIMIENTOS

A Dios por tener un plan de vida para mí y darme lo necesario para poder culminar esta etapa.
A mi madre, por guiarme todos estos años sin dudar de mi capacidad y sacrificando todo para darme un buen futuro.

A mi padre, por tener la firmeza para guiar mi hogar y ser sabio en las decisiones que toma.
A mis Hermano mayor, que es un gran ejemplo en muchos aspectos de mi vida y que me ayuda a mejorar cada día.

A mi Hermana menor, que es una gran compañera y que se siempre se esfuerza para poder salir adelante.

A mi sobrino Noha, por alegrar mi hogar en todo momento.

A mi tío Gregorio Flores, que me dejó muchas enseñanzas y lecciones de vida durante el tiempo que compartimos.

A mis tíos: Ramiro, Orlando, Eva, que siempre me apoyaron con cariño y fueron importantes en este recorrido.

A mis amigos de infancia con los que crecí y fueron cómplices de las aventuras inolvidables en el lugar donde crecimos, les agradezco por la lealtad, la sinceridad, el apoyo por haber estado en momentos tristes y alegres de mi vida, gracias por haber formado una familia inmensa y dejarme hermosos recuerdos.

A mi compañera Katherine Fernández, por los muchos momentos compartidos, por todo el apoyo que me brindó desde el primer momento y por ser la persona que más apoyo me brindó en todas las etapas de mi Proyecto de Grado.

A mis mejores amigos de la universidad Joel, Roger y Horacio que fueron un impulso en muchos momentos cruciales mi carrera universitaria.

A mi tutor PhD. Javier Hugo Gustavo Reyes Pacheco, que tuvo la paciencia y la disponibilidad para ayudarme durante todo el proceso de desarrollo de mi Proyecto de Grado
Al Doctor Brayan Almazán Garate y al Consultorio Dental Ortho-Braes por darme la confianza y haberme permitido realizar el presente Proyecto de Grado.

De todo corazón muchas gracias.

RESUMEN

Con el paso de los años, las tecnologías van avanzando y van transformando los modelos de empresa e instituciones. Las automatizaciones de procesos pretenden ahorrar a las instituciones no solo dinero, sino tiempo y trabajo manual. De esta manera la información se vuelve más eficiente, requiriendo menos mano de obra y proporcionando mucha más exactitud y precisando menos tiempo para proporcionar el mismo volumen de información. Los sistemas de información hoy en día, pueden aportar en la toma de decisiones administrativas para la organización que estarían reflejados en la cadena de abastecimiento, relaciones con el cliente y/o usuarios y la planeación de escenarios de negocios. Por esta razón se propone el Proyecto de Grado titulado: “SISTEMA WEB DE ADMINISTRACIÓN, CONTROL Y GESTIÓN DE HISTORIALES CLÍNICOS CASO: CONSULTORIO DENTAL ORTHO-BRAES”.

Para el diseño del sistema, se adoptó la metodología ágil SCRUM, que es adecuado para el desarrollo con páginas web. Para la implementación del sistema, se utilizaron lenguajes orientados al servidor como: NodeJs, Sequelize y Gestor de la base de datos MySQL, en el desarrollo se utilizará las siguientes tecnologías Angular, PrimeNg, Angular Material, Css, Express y Html5, además de la utilización del servidor Web Apache. La medición de la calidad del Sistema se realizó a través de la NORMAS ISO/IEC 9126, que maneja cinco parámetros como: la usabilidad, funcionalidad, confiabilidad, portabilidad y mantenibilidad.

Con la implementación del sistema se logró obtener una mejor administración de todos los procesos que realiza el consultorio en la atención de un paciente por parte del médico odontólogo

Palabras clave: Programación de citas, odontograma, historia clínica.

godiev320@gmail.com

ABSTRACT

Over the years, technologies are advancing and are transforming business models and institutions. Process automations aim to save institutions not only money, but time and manual work. In this way the information becomes more efficient, requiring less labor and requires much more accuracy and less time to provide the same volume of information.

Information systems today can contribute to administrative decision making for the organization that would be reflected in the supply chain, relationships with customers and/or users, and the planning of business scenarios. For this reason, the Degree Project entitled: "WEB SYSTEM OF ADMINISTRATION, CONTROL AND MANAGEMENT OF CLINICAL RECORDS CASE: ORTHO-BRAES DENTAL OFFICE".

For the design of the system, the agile SCRUM methodology was changed, which is suitable for development with web pages. For the implementation of the system, server-oriented languages were used such as: NodeJs, Sequelize and MySQL database manager, in the development the following technologies were used: Angular, PrimeNg, Angular Material, Css, Express and Html5, in addition to the use of the Apache web server. The measurement of the quality of the System was carried out through the ISO/IEC 9126 STANDARDS, which manages five parameters such as: usability, functionality, reliability, portability and maintainability.

With the implementation of the system, it was modified to obtain a better administration of all the processes carried out by the office in the care of a patient by the dentist.

Keywords: Appointment scheduling, odontogram, clinical history.

godiev320@gmail.com

ÍNDICE DE CONTENIDO

CAPÍTULO I.....	1
1. INTRODUCCIÓN.....	1
1.2 ANTECEDENTES.....	2
1.2.1 ANTECEDENTES INSTITUCIONALES.....	2
1.2.1 ANTECEDENTES PROYECTOS SIMILARES.....	3
1.3 PLANTEAMIENTO DEL PROBLEMA.....	4
1.3.1 PROBLEMA CENTRAL.....	4
1.3.2 PROBLEMAS SECUNDARIOS.....	4
1.4 OBJETIVOS.....	5
1.4.1 OBJETIVO GENERAL.....	5
1.4.2 OBJETIVOS ESPECÍFICOS.....	5
1.5 JUSTIFICACIÓN.....	5
1.5.1 JUSTIFICACIÓN ECONÓMICA.....	6
1.5.2 JUSTIFICACIÓN SOCIAL.....	6
1.5.3 JUSTIFICACIÓN TECNOLÓGICA.....	6
1.6 ALCANCES Y LÍMITES.....	7
1.6.1 ALCANCES.....	7
1.6.2 LÍMITES.....	7
1.7 APORTES.....	8
1.7.1 PRÁCTICO.....	8
1.7.2 TEÓRICO.....	8
1.8 METODOLOGÍA.....	8
1.8.1 METODOLOGÍA DE DESARROLLO.....	8
CAPITULO II.....	9
2. MARCO TEÓRICO.....	9
2.1 INGENIERÍA DEL SOFTWARE.....	9
2.1.1 MARCO CONCEPTUAL DE LA INGENIERÍA DEL SOFTWARE.....	11
2.2 LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN.....	13

2.3 METODOLOGÍA ÁGIL SCRUM.....	14
2.3.1 MARCO CONCEPTUAL DE LA METODOLOGÍA SCRUM	14
2.3.2 EQUIPO DE TRABAJO (ROLES)	16
2.3.2.1 PRODUCT OWNER.....	16
2.3.2.2 SCRUM MASTER.....	17
2.3.2.3 EQUIPO DE DESARROLLO	18
2.3.2.4 STAKEHOLDERS.....	18
2.3.3 REUNIONES.....	19
2.3.3.1 PLANIFICACIÓN DEL BACKLOG	19
2.3.3.2 SEGUIMIENTO DEL SPRINT.....	19
2.3.3.3 REVISIÓN DEL SPRINT	19
2.3.4 ELEMENTOS DE SCRUM.	19
2.3.5 FORMATO DE LA PILA DEL PRODUCTO (PRODUCT BACKLOG).	21
2.3.6 SPRINT BACKLOG.....	21
2.3.6.1 LISTA DE PRIORIDADES	22
2.3.7 INCREMENTO	22
2.3.8 PROCESO DE LA METODOLOGÍA.....	23
2.3.8.1 PRE – GAME.....	23
2.3.2.2 GAME	23
2.3.2.3 POST – GAME	24
2.4 INGENIERÍA WEB.....	24
2.4.1 PROCESO DE INGENIERÍA WEB.....	25
2.5 METODOLOGÍA UWE	26
2.5.1 FASES DE LA METODOLOGÍA UWE.....	26
2.5.1.1 DIAGRAMA DE CASOS DE USO (ANÁLISIS DE REQUISITOS).....	26
2.5.1.2 MODELO NAVEGACIONAL.....	27
2.5.1.3 MODELO CONCEPTUAL.....	29
2.5.1.4 DIAGRAMA DE CLASES	30
2.6 DETERMINACIÓN DE PUNTO FUNCIÓN.....	30
2.6.1 ENTRADA DE DATOS EN LÍNEA.....	30

2.6.2 INTERFACE CON EL USUARIO.....	31
2.6.3 REUSABILIDAD DE CÓDIGO	32
2.7 TÉCNICA ISO 9126.....	33
2.7.1 IMPORTANCIA DE LA NORMA ISO/IEC 9126: 2001.....	34
2.8 MODELO DE ESTIMACIÓN COCOMO II.....	36
CAPÍTULO III	37
3. MARCO APLICATIVO	37
3.1 ANÁLISIS DE LA SITUACIÓN ACTUAL.....	38
3.2 IDENTIFICACIÓN DE ROLES SCRUM.....	39
3.3 FASE DE PRE-GAME	40
3.3.1 CREACIÓN DEL PRODUCT BACKLOG (PILA DE PRODUCTO)	40
3.3.2 IDENTIFICACIÓN DE ROLES DE USUARIO	41
3.4 FASE DEL GAME.....	42
3.4.1 SPRINT 1: MÓDULO DE AUTENTICACIÓN DE USUARIO	42
3.4.1.2 PLANIFICACIÓN DEL SPRINT 1.....	42
3.4.1.2 ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 1	43
3.4.1.3 MODELO NAVEGACIONAL DEL SPRINT 1	45
3.4.1.4 PANTALLAS DEL SPRINT 1.....	45
3.4.1.5 PRUEBA UNITARIA DEL SPRINT 1.....	47
3.4.2 SPRINT 2: MÓDULO DE CONTROL DE PACIENTES.....	47
3.4.2.1 PLANIFICACIÓN DEL SPRINT 2.....	47
3.4.2.2 ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 2	49
3.4.2.3 MODELO NAVEGACIONAL DEL SPRINT 2.....	51
3.4.2.4 PANTALLAS DEL SPRINT 2.....	52
3.4.2.5 PRUEBA UNITARIA DEL SPRINT 2	54
3.4.3 SPRINT 3: MÓDULO DE CONTROL DE TRATAMIENTOS Y PIEZAS DENTALES (ODONTOGRAMA)	55
3.4.3.1 PLANIFICACIÓN DEL SPRINT 3.....	55
3.4.3.2 ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 3	56
3.4.3.3 MODELO NAVEGACIONAL DEL SPRINT 3	58

3.4.3.4 PANTALLAS DEL SPRINT 3.....	59
3.4.3.5 PRUEBA UNITARIA DEL SPRINT 3	60
3.4.4 SPRINT 4: MÓDULO DE MANEJO DE HISTORIALES CLÍNICOS	61
3.4.4.1 PLANIFICACIÓN DEL SPRINT 4.....	62
3.4.4.2 ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 4	62
3.4.4.3 MODELO NAVEGACIONAL DEL SPRINT 4	64
3.4.4.4 PANTALLAS DEL SPRINT 4.....	64
3.4.4.5 PRUEBA UNITARIA DEL SPRINT 4.....	66
3.4.5 SPRINT 5: MÓDULO DE CONTROL COSTOS DE TRATAMIENTOS Y PAGOS	67
3.4.5.1 PLANIFICACIÓN DEL SPRINT 5.....	67
3.4.5.2 ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 5	68
3.4.5.3 MODELO NAVEGACIONAL DEL SPRINT 5	71
3.4.5.4 PANTALLAS DEL SPRINT 5.....	72
3.4.5.5 PRUEBA UNITARIA DEL SPRINT 5.....	73
3.4.6 SPRINT 6: MÓDULO DE CONTROL DE CITAS MÉDICAS.....	73
3.4.6.1 PLANIFICACIÓN DEL SPRINT 6.....	74
3.4.6.2 ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 6	74
3.4.6.3 MODELO NAVEGACIONAL DEL SPRINT 6.....	76
3.4.6.4 PANTALLAS DEL SPRINT 6.....	77
3.4.6.5 PRUEBA UNITARIA DEL SPRINT 6.....	77
3.4.7 DIAGRAMA DE CLASES.	78
3.5 MODELO ENTIDAD RELACIÓN.....	78
3.6 FASE DEL POST – GAME.....	79
3.6.1 PRUEBAS DE STRESS.....	79
CAPITULO IV	80
4. CALIDAD Y SEGURIDAD	80
4.1 FACTORES DE SEGURIDAD.....	80
4.1.1 SEGURIDAD DE BASE DE DATOS.....	80
4.1.2 COPIAS DE SEGURIDAD.....	80

4.1.3 SEGURIDAD DE AUTENTIFICACIÓN DE USUARIO	81
4.1.3.1 BCrypt	81
4.1.4 SEGURIDAD DEL SISTEMA.....	81
4.1.4.1 JSON WEB TOKEN	81
4.1.4.2 ANGULAR GUARDS	81
4.2 CALIDAD DE SOFTWARE	82
4.2.1 TÉCNICA ISO 9126.....	82
4.2.1.1 FUNCIONALIDAD	82
4.2.1.2 USABILIDAD	85
4.2.1.3 CONFIABILIDAD.....	88
4.2.1.4 MANTENIBILIDAD	90
4.2.1.5 PORTABILIDAD	91
4.2.2 RESULTADOS.....	92
CAPÍTULO V.....	93
5. COSTO Y BENEFICIO	93
5.1 INTRODUCCIÓN	93
5.2 COSTO DEL SISTEMA	93
5.2.1 COCOMO II	93
5.2.2 COSTO DEL SOFTWARE DESARROLLADO	94
5.2.3 COSTO DE IMPLEMENTACIÓN DEL SISTEMA.....	97
5.2.4 COSTO DE ELABORACIÓN DEL SISTEMA.....	97
5.2.5 COSTO TOTAL DE SOFTWARE.....	98
5.3 BENEFICIO	98
5.3.1 CÁLCULO DEL VALOR ANULA NETO (VAN)	98
5.3.2 LA TASA INTERNA DE RETORNO TIR	100
5.4 COSTO/BENEFICIO.....	102
CAPÍTULO VI	103
6. CONCLUSIONES Y RECOMENDACIONES	103
6.1 CONCLUSIONES	103
6.2 RECOMENDACIONES	103

7. BIBLIOGRAFÍA	105
8. ANEXOS	108
8.1 ANEXO A – ÁRBOL DE PROBLEMAS	108
8.2 ANEXO B – ÁRBOL DE OBJETIVOS	109
8.3 ANEXO C – CUESTIONARIOS	110
8.4 ANEXO D – CRONOGRAMA DE AVANCE.....	113

ÍNDICE DE FIGURAS

Figura 2.1. Capas de la Ingeniería del Software.....	10
Figura 2.2. Clasificación de la tecnología.....	13
Figura 2.3. Estructura Central de Scrum.....	16
Figura 2.4. Ciclo de desarrollo Scrum.....	20
Figura 2.5. Ejemplo de un Product Backlog.....	21
Figura 2.6. Ejemplo de Sprint Backlog.....	22
Figura 2.7. Visión general del proceso de la metodología Scrum.....	24
Figura 2.8. Análisis de caso de uso.....	27
Figura 2.9. Modelo navegacional UWE.....	29
Figura 2.10. Modelo conceptual.....	29
Figura 3.1. Modelo de procesos del proyecto.....	38
Figura 3.2. Modelo anterior de historia clínica del consultorio dental Ortho-Braes.....	39
Figura 3.3. Caso de uso – Autenticación de usuario.....	43
Figura 3.4. Modelo navegacional – Autenticación de usuario.....	45
Figura 3.5. Página de inicio del sistema.....	46
Figura 3.6. Página inicial administrativa.....	46
Figura 3.7. Caso de uso – Control de pacientes.....	49
Figura 3.8. Modelo Navegacional – Control de Pacientes.....	52
Figura 3.9. Interfaz Lista de paciente.....	53
Figura 3.10. Interfaz Registrar nuevo paciente.....	53
Figura 3.11. Información de un paciente seleccionado.....	54
Figura 3.12. Caso de uso – Control de tratamientos y piezas dentales.....	57
Figura 3.13. Modelo navegacional – Control de tratamientos y piezas dentales.....	59
Figura 3.14. Modelo de Odontograma.....	59
Figura 3.15. Interfaz de tratamientos del consultorio dental.....	60
Figura 3.16. Caso de uso – Manejo de historias clínicas.....	62
Figura 3.17. Modelo navegacional – Manejo de Historias Clínicas.....	64
Figura 3.18. Modelo de historia clínica.....	65

Figura 3.19. Modelo de historia clínica 2.....	65
Figura 3.20. Modelo de historia clínica 3.....	66
Figura 3.21 Caso de uso - Control de pagos.....	68
Figura 3.22 Modelo navegacional – Control de pagos	71
Figura 3.23. Pantalla de costos de tratamientos.....	72
Figura 3.24. Pantalla de pagos realizados	72
Figura 3.25 Caso de uso - Control de citas médicas.....	74
Figura 3.26. Modelo navegacional – Control de citas médicas.....	76
Figura 3.27. Modelo de agenda	77
Figura 3.28. Diagrama de clases.....	78
Figura 3.29. Modelo entidad-relación.....	79
Figura 5.1. Cálculo del TIR.....	101

ÍNDICE DE TABLAS

Tabla 2.1. Elementos de modelo navegacional	28
Tabla 2.2. Entrada de Datos en Línea	31
Tabla 2.3. Interfase con el Usuario	32
Tabla 2.4. Reusabilidad de Código.....	33
Tabla 3.1. Identificación de Roles de Scrum.....	40
Tabla 3.2. Product Backlog	41
Tabla 3.3. Identificación de los roles de usuario	41
Tabla 3.4. Primer Sprint Backlog	43
Tabla 3.5. Especificación de casos de uso - Autenticación de usuario.....	44
Tabla 3.6. Prueba unitaria - Sprint 1	47
Tabla 3.7. Segundo Sprint Backlog	48
Tabla 3.8. Especificación de casos de uso - Registro de Pacientes	50
Tabla 3.9. Especificación de casos de uso - Lista de Pacientes.....	51
Tabla 3.10. Prueba unitaria uno - Sprint 2	54
Tabla 3.11. Prueba unitaria dos - Sprint 2.....	55
Tabla 3.12. Tercer Sprint Backlog.....	56
Tabla 3.13. Especificación de casos de uso - Control de tratamientos y piezas dentales	58
Tabla 3.14. Prueba unitaria - Sprint 3	61
Tabla 3.15. Cuarto Sprint Backlog	62
Tabla 3.16. Especificación casos de uso - Control de tratamientos y piezas dentales.....	63
Tabla 3.17. Prueba unitaria - Sprint 4	67
Tabla 3.18. Quinto Sprint Backlog	68
Tabla 3.19. Especificación de casos de uso - Registrar pago inicial	69
Tabla 3.20. Especificación de casos de uso - Costos de tratamientos	70
Tabla 3.21. Prueba unitaria - Sprint 5	73
Tabla 3.22. Sexto Sprint Backlog	74
Tabla 3.23. Especificación de caso de uso - Registrar cita médica	75
Tabla 3.24. Especificación de casos de uso - Eliminar cita médica	76

Tabla 3.25. Prueba unitaria - Módulo de control de citas	77
Tabla 4.1. Conteo paramétrico – Funcionalidad.....	83
Tabla 4.2. Ajuste de complejidad	84
Tabla 4.3. Ajuste de complejidad	88
Tabla 4.4. Confiabilidad del Sistema.....	89
Tabla 4.5. Porcentaje total de Métricas del Calidad	92
Tabla 5.1. Relación de valores en el modelo COCOMO	94
Tabla 5.2. Conversión de puntos función.....	95
Tabla 5.3. Costos elaboración	97
Tabla 5.4. Costo total	98
Tabla 5.5. Cantidad por año	99
Tabla 5.6. Interpretación del VAN	100

CAPÍTULO I

1. INTRODUCCIÓN

Con el paso de los años, las tecnologías van avanzando y van transformando los modelos de empresa e instituciones y poco a poco, los nuevos requisitos son más exigentes. Las automatizaciones de procesos pretenden ahorrar a las instituciones no solo dinero, sino tiempo y trabajo manual. De esta manera la información se vuelve más eficiente, requiriendo menos mano de obra y proporcionando mucha más exactitud y precisando menos tiempo de funcionamiento para proporcionar el mismo volumen de información. Al usarse automatizaciones, la cantidad de errores posibles en el manejo de datos se ve muy reducida. Los humanos son los que cometen los fallos, pero las máquinas simplemente reproducen una y otra vez de manera incansable, la orden que se les ha asignado. Introduciendo automatizaciones en los procesos se evitan los fallos de los operarios.

Hoy en día no todas las instituciones de salud cuentan con un sistema que les ayude a organizar la información que necesitan al momento de brindar atención a los pacientes, sin embargo, muchas de estas instituciones utilizan distintas herramientas y funcionalidades de distintos softwares para reducir tiempo, costos y aumentar sus beneficios, así de esta forma proporcionar rápidos y buenos servicios a los pacientes, doctores, etc. Los avances tecnológicos también ayudan a la digitalización documental. Lo que antes era muy costoso de archivar se ha vuelto muy sencillo gracias a las nuevas aplicaciones de software que se adaptan a las necesidades de cada institución, de esta manera la información se tiene almacenada de forma digital en muy poco espacio y sin ocupar estanterías. (Hidalgo, 2019)

Los sistemas de información hoy en día, pueden aportar en la toma de decisiones administrativas para la organización que estarían reflejados en la cadena de abastecimiento, relaciones con el cliente y/o usuarios y la planeación de escenarios de negocios. Se debe utilizar procedimientos operativos para maximizar la eficiencia, información precisa de toda la empresa o institución y proporcionar las herramientas para facilitar los procesos de toma de decisiones que ayudarán a la empresa o institución a alcanzar sus objetivos estratégicos. Los sistemas de información modifican profundamente la manera en que una empresa, una

industria, un negocio deba ajustarse al impacto digitalizado y los nuevos flujos de información. Los sistemas de información basados en computadoras, son indispensables para realizar las actividades planificadas en cualquier organización. (Hamidian & Ospino,2015)

El Consultorio Dental ORTHO-BRAES tiene como labor principal brindar un servicio eficaz y eficiente a los pacientes que acuden a solicitar de sus servicios y al igual que muchos establecimientos no cuentan con un sistema de información que les ayude a controlar sus procesos como ser: historial clínico, información de pagos, control de horarios, registro de pacientes y control de piezas dentales, estos procesos hacen que el personal del consultorio demore en la atención de los pacientes, a partir de esto el consultorio busca automatizar sus procesos ya que busca cumplir las expectativas de los pacientes, evitando molestias y facilitando el manejo de distintos tratamientos para un mismo paciente.

1.2 ANTECEDENTES

1.2.1 ANTECEDENTES INSTITUCIONALES

El consultorio dental ORTHO-BRAES, ubicado en la Plaza Arandía, Zona Villa Fátima, calle la Virgen de la Concepción 1378, cuenta con al menos dos médicos odontólogos, este establecimiento tiene 28 años de brindar servicio y tuvo un gran crecimiento en cuanto al número de pacientes por la buena atención que brinda.

Este consultorio dental cuenta con equipamiento de excelente calidad como: Ultrasonido (para hacer limpiezas dentales), Cámara intraoral (para poder observar un antes y un después de los tratamientos realizados). Suctor hemostático (Para realizar cirugías). En el consultorio se realizan servicios de: Endodoncia (tratamiento de conductos), Ortodoncia, Cirugía (extracciones), Operatoria (curaciones), Ortopedia, Periodoncia (limpieza), Odontopediatría, Estética (blanqueamiento) y Prótesis Dental con especialidad en: Ortodoncia y Rehabilitación Oral.

Este establecimiento intentó hacer uso de programas, sistemas web y aplicaciones de internet las cuales no tenían muchas funcionalidades que requería el establecimiento y se hizo muy difícil organizar y manejar los datos del personal y de los pacientes.

Actualmente el establecimiento no cuenta con un sistema de información principal, por lo cual hace uso de distintas funcionalidades y herramientas de distintos sistemas.

1.2.1 ANTECEDENTES PROYECTOS SIMILARES

En la Carrera de Informática de la Universidad Mayor de San Andrés se realizaron varios Proyectos de Grado sobre esta problemática, algunos de los cuales se enumeran a continuación:

- “Sistema web de control de pagos, citas e historiales clínicos Caso: Clínica Dental Lavadent”, el problema radica en el manejo deficiente del control de pagos, citas e historiales clínicos, que provoca una demora al momento de proporcionar la información necesaria a los pacientes y al personal. (Huanca, 2015).
- “Sistema de Administración y Control de Historiales clínicos para los Consultorios Clínicos de la U.M.S.A.”, presenta el Desarrollo de un Sistema de Administración y Control de Historiales Clínicos de los pacientes universitarios, que permite mejorar las tareas de admisión, consulta, búsqueda y elaboración de reportes o informes de manera más rápida y confiable, reduciendo el tiempo en el registro de datos y manejo de la información de los consultorios de la U.M.S.A. (Flores, 2014).
- “Sistema de Información para el Control de Historiales Clínicos (SICHC)”, el problema radica principalmente en la gran cantidad de historiales clínicos del paciente, expedientes y agendas de los doctores con los que cuenta. Es una clínica especializada, los pacientes suelen volver periódicamente, por eso surge la búsqueda del historial, con lo que tomaba mucho tiempo, ya que era en forma manual, el cual logro implementar los módulos de registro de pacientes, agenda, consultas, cirugía, internación y exámenes de laboratorio, desarrollado el año 2010 (Argollo, 2010).
- “Sistema de Seguimiento a Paciente Control-Asignación Proyecciones Hospital La Paz”, el problema radica en manejar grandes volúmenes de información manualmente en el cual se implementó los módulos de control de consultorio externo, laboratorio,

farmacia, seguimiento y control de internación de pacientes, desarrollado el año 2000 (Chirina y Morales, 2000).

Existe una gran variedad de sistemas dedicados al manejo de historiales clínicos en instituciones médicas, diferenciándose por las necesidades específicas de las operaciones y manipulación de datos.

1.3 PLANTEAMIENTO DEL PROBLEMA

Debido a la creciente asistencia de los pacientes que solicitan los servicios médicos en el “Consultorio Médico Ortho-Braes” diariamente se generan una gran cantidad de historiales clínicos que son requeridos como respaldo legal tanto para el paciente como para los médicos odontólogos, debido a esto la labor de los médicos se hace cada vez más ardua, ya que estos informes se los realizan manualmente y se los archiva en estanterías lo que también genera un gran volumen de informes.

Estos informes también contienen información de los pacientes que están siendo tratados. La mayoría de estos tratamientos se hacen en múltiples sesiones, debido a esto, los informes deben estar bien organizados para continuar o reanudar el tratamiento desde la última sesión realizada.

1.3.1 PROBLEMA CENTRAL

Teniendo en cuenta estos antecedentes ahora se plantea el siguiente problema:

¿Cómo realizar la administración, control y gestión de historiales clínicos del Consultorio dental Ortho-Braes?

1.3.2 PROBLEMAS SECUNDARIOS

- El personal médico no tiene información concreta sobre los pacientes antiguos que se tratan en el consultorio, es así que el diagnóstico se hace moroso.
- Los pacientes no tienen la información del costo de los tratamientos que se realizan en el consultorio por esta razón existe una desconfianza con el consultorio médico.

- El consultorio carece de datos para tratamientos que se realizan en varias sesiones lo que ralentiza el tratamiento.
- Para las piezas dentales que se trataron con anterioridad no se tiene información necesaria lo que dificulta su control y seguimiento.
- Las citas médicas agendadas con los pacientes del consultorio solo son anotadas en hojas de papel lo que dificulta al odontólogo en caso de cambiar o cancelar una cita.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar un Sistema web de administración, control y gestión de historiales clínicos para el Consultorio dental Ortho-Braes.

1.4.2 OBJETIVOS ESPECÍFICOS

- Mostrar las citas programadas con fecha y hora para que el personal médico obtenga la información deseada en el momento.
- Definir un proceso que permita controlar la información de las piezas dentales de cada paciente.
- Optimizar el seguimiento y control de los tratamientos de cada paciente mediante fotos y radiografías digitales.
- Controlar los pagos y cuotas de cada tratamiento por cada sesión que se realice.
- Visualizar los costos de cada tipo de tratamiento, mismos que están regidos por el colegio de odontólogos.

1.5 JUSTIFICACIÓN

Con los conocimientos desarrollados durante el ciclo de estudios en la Carrera de Informática y con el objetivo de colaborar al Consultorio dental Ortho-Braes, se propone la implementación de un sistema web de administración, control y gestión de historiales

clínicos, este proyecto será de gran apoyo para un mejor manejo de la información y una mejor atención a los pacientes.

1.5.1 JUSTIFICACIÓN ECONÓMICA

El sistema ahorrará recursos económicos como ser material de escritorio y papelería que son invertidos en el proceso de redacción de historiales clínicos, manejo de agenda y recibos de pago.

El sistema proporcionará beneficios a corto, mediano y largo plazo que se observará a través de sistema web de seguimiento y control de historiales clínicos, agendas, pacientes, tratamientos y pagos. El desarrollo del sistema web no tendrá costo, ya que será desarrollado con herramientas de software libre y estándares abiertos, aprovechando estos recursos para obtener un producto de calidad.

1.5.2 JUSTIFICACIÓN SOCIAL

La implementación del presente proyecto ofrecerá grandes beneficios al consultorio dental, como es el de brindar un mejor seguimiento y control de procesos que proporcione información oportuna y confiable, reduciendo el trabajo excesivo y moroso para un mejor control y seguimiento de sus pacientes, con información centralizada y actualizada.

El sistema apoyará tanto a los médicos como a los pacientes en cuanto al seguimiento de historias clínicas, alimentando la relación entre médicos y pacientes a través de los servicios que otorga el sistema web. Por otro lado, los pacientes podrán tener acceso directo vía internet, para poder ver el horario de su cita programada, poder dejar algún aviso al médico, enterarse de todos los tratamientos que realizan en la clínica dental y el costo de cada tipo de tratamiento, de forma inmediata.

1.5.3 JUSTIFICACIÓN TECNOLÓGICA

En cuanto a los recursos de hardware y software, la institución cuenta con equipos suficientes donde se puede implementar el sistema de información. Para el desarrollo del sistema se

utilizará tecnologías como Node.js, Express.js, Angular y SequelizeORM de Node.js para interactuar con la base de datos MySQL, entre otros. Debido a los muchos beneficios que ofrecen, además que será gratuito, debido al uso de herramientas de software libre.

1.6 ALCANCES Y LÍMITES

1.6.1 ALCANCES

El alcance del sistema web para el consultorio dental Ortho-Braes cubrirá los siguientes que se encarga de la gestión de información específica necesaria para ampliar establecer un vínculo entre el paciente y el consultorio dental, descrita a continuación aspectos:

- Autenticación de usuarios.
- Registro de pacientes.
- Control de tratamientos dentales.
- Control de piezas dentales.
- Generación de historiales clínicos.
- Control de pagos.
- Programación de citas.

1.6.2 LÍMITES

A continuación, se detalla los límites del sistema:

- El uso del sistema será solo para uso del personal del consultorio.
- El sistema no está orientado a procesos contables, como la emisión de facturas.
- El sistema no podrá registrar nuevos usuarios por el hecho de contar con personal limitado.

1.7 APORTES

1.7.1 PRÁCTICO

El aporte informático del proyecto presenta un Sistema web de control de citas, pagos e historiales clínicos, haciendo uso óptimo de recursos de hardware y software, el mismo agilizará el proceso de trabajo del personal del consultorio, además de tener una mejor gestión de la información.

1.7.2 TEÓRICO

El sistema web, es desarrollado con la metodología ágil Scrum el cual es un proceso que se aplica de manera regular un conjunto de buenas prácticas, para trabajar colaborativamente en equipo.

Para el modelado del sistema web, la metodología UWE basada en el modelo UML, proporciona los elementos y diagramas necesarios para el buen desarrollo del modelado del sistema.

1.8 METODOLOGÍA

1.8.1 METODOLOGÍA DE DESARROLLO

El presente proyecto será elaborado bajo algunos métodos que servirán para el requerimiento de datos que nos servirá para el análisis, diseño y posteriormente la implementación y la entrega del sistema web.

Para el desarrollo de este proyecto, en el modelado, diseño y desarrollo del sistema web, se hará uso de una metodología de desarrollo de software ágil en este caso se aplicará la metodología SCRUM y para el modelado se utilizará la metodología UWE.

Las herramientas tecnológicas utilizadas son: Gestor de la base de datos MySQL, en el desarrollo se utilizará las siguientes tecnologías: JavaScript, Express, Bootstrap 4, Sequelize, PrimeNg, Angular Material, Angular y NodeJs.

CAPITULO II

2. MARCO TEÓRICO

2.1 INGENIERÍA DEL SOFTWARE

La introducción del término Ingeniería del Software se produce en la primera conferencia sobre Ingeniería del Software patrocinada por la OTAN, celebrada en Garmisch (Alemania) en octubre de 1968, no obstante, la paternidad del término se le atribuye a Fritz Bauer.

Fue tal la aceptación de esta conferencia que se consiguió un nuevo patrocinio de la OTAN para una segunda conferencia sobre Ingeniería del Software, que tendría lugar un año más tarde en Roma (Italia), con unos resultados menos esperanzadores que los producidos en la primera conferencia. De hecho, no se produjo ninguna petición de que continuara la serie de conferencias de la OTAN, lo cual no influyó para que a partir de entonces se utilizara con gran profusión el nuevo término para describir los trabajos realizados, aunque quizás sin un consenso real sobre su significado. En el contexto educativo, sin duda alguna, lo que más controversia ha levantado es el propio nombre del término, centrándose la discusión en la pregunta ¿es la Ingeniería del Software realmente una Ingeniería? Los argumentos que se dan para sustentar una respuesta negativa se pueden resumir en dos categorías. La primera reuniría a aquellos que se ciñen a la definición literal de ingeniería dada por algunos diccionarios o sociedades profesionales, argumentando que en estas definiciones se hace mención a productos tangibles derivados del uso efectivo de materiales y fuerzas naturales, mientras que el software ni es tangible, ni utiliza materiales y/o fuerzas naturales para su concepción. La segunda categoría estaría formada por los que arguyen que una disciplina ingenieril evoluciona desde una profesión y la profesión relacionada con el software no ha evolucionado lo suficiente para ser considerada una ingeniería. Por el contrario, son muchos los que están a favor de la utilización y difusión del término Ingeniería del Software, tomando como un estándar de facto la utilización reiterada del término en la bibliografía especializada. Quizás la defensa más fuerte y adecuada de la Ingeniería del Software como ingeniería venga de la mano de Mary Shaw que justifica que, si tradicionalmente se ha definido ingeniería

como “la creación de soluciones rentables a problemas prácticos mediante la aplicación del conocimiento científico para la construcción de cosas al servicio de la humanidad”, entonces el desarrollo del software es un problema ingenieril apropiado, porque involucra “la creación de soluciones rentables económicamente para problemas prácticos” (Bolaños, Urrea, Gómez, 2015).



Figura 2.1. Capas de la Ingeniería del Software

Fuente: Estigarríbia, 2015

No solo hace falta decir lo que es la Ingeniería del Software, sino que es conveniente recalcar lo qué no es; así la Ingeniería del Software no es el diseño de programas que se implementan en otras áreas ingenieriles, ni es simplemente una forma de programar más organizada que la que prevalece entre aficionados, principiantes o personas con falta de educación y entrenamiento específico.

El concepto de Ingeniería del Software surge de la distinción entre el desarrollo de pequeños proyectos (programming in the small) y el desarrollo de grandes proyectos (programming in the large), de forma que el reconocimiento de que la Ingeniería del Software está relacionada con esta última. Este primer concepto fue rápidamente ampliado para incorporar a la Ingeniería del Software todas aquellas tareas relacionadas con la automatización de los Sistemas de Información y con la Ingeniería de Sistemas en general. (Rossainz y Baez, 2011)

2.1.1 MARCO CONCEPTUAL DE LA INGENIERÍA DEL SOFTWARE

El estudio de las características comunes de los sistemas se conoce como Teoría General de Sistemas. Los principios de esta teoría, derivados del estudio de otros sistemas, se pueden aplicar a los sistemas automatizados. Algunos de los principios generales de la teoría general de sistemas son los siguientes:

- Cuanto más especializado sea un sistema menos capaz es de adaptarse a circunstancias diferentes.
- Cuanto mayor sea el sistema mayor es el número de recursos que deben dedicarse a su mantenimiento diario.
- Los sistemas siempre forman parte de sistemas mayores y siempre pueden dividirse en sistemas menores.
- Los sistemas crecen.

La Ingeniería de sistemas es uno de los antecedentes de la Ingeniería del Software. En ella las funciones deseadas de un sistema se descubren, analizan y se asignan a elementos de sistemas individuales. El ingeniero de sistemas parte de los objetivos definidos por el usuario y desarrolla una representación de la función, rendimiento, interfaces, restricciones de diseño y estructura de la información que pueden ser asociados a cada uno de los elementos del sistema (documentos, procedimientos, bases de datos, entre otros.).

No existe una única forma de organizar las fases del ciclo de vida del software, la forma más natural de hacerlo sería ordenar sus fases de manera secuencial según ocurren en el tiempo, pero esto se ha demostrado que no es realista por la naturaleza evolutiva del software, por ello se le da un orden lógico, es decir, se representan de una manera comprensible, pero dicho orden no implica que ocurra exactamente de esa forma en el tiempo. A esta manera de representar las fases de un ciclo de vida o, lo que es lo mismo, las fases de un proceso, se denomina modelo de ciclo vida o modelo de proceso. Por tanto, no existe un único modelo de ciclo de vida o de proceso. A lo largo de los años han ido apareciendo modelos que

incluyen diferentes visiones del proceso de desarrollo. Los modelos de proceso tienen su origen en intentar traer orden al caos del desarrollo de software. Históricamente se tiene que estos modelos han aportado una cierta estructura al trabajo de los ingenieros de software debido a la definición de una guía efectiva para los equipos de trabajo. Sin embargo, el trabajo de los ingenieros de software y los artefactos que se producen están al borde del caos. El borde del caos se puede definir como “el estado natural entre el orden y el caos, un gran compromiso entre estructura y sorpresa”. El borde del caos se puede visualizar como un estado inestable y parcialmente estructurado. Es inestable porque se siente constantemente atraído por el caos o el orden absoluto. Se tiene la tendencia a pensar que el orden es el estado ideal de la naturaleza, pero esto no siempre es cierto, el trabajo fuera del equilibrio genera creatividad, procesos autoorganizados y rendimientos crecientes. El orden absoluto significa la ausencia de variabilidad, que podría ser una ventaja en entornos impredecibles. El cambio ocurre cuando hay alguna estructura para que el cambio pueda organizarse, pero sin tanta rígido como para que no pueda ocurrir. Demasiado caos, por otro lado, puede hacer que la coordinación y la coherencia sean imposibles. La falta de estructura no siempre significa desorden. Las implicaciones filosóficas de este argumento son significativas para la Ingeniería de Software. Cada modelo de proceso trata de encontrar un equilibrio entre la necesidad de impartir orden en un mundo caótico y la necesidad de adaptarse cuando las cosas cambian constantemente. (Pinzón, 2017).

En la Figura 2.2 se puede ver una clasificación bidimensional de la tecnología basada en la clasificación de Charles Perrow. En el eje de las ordenadas se presenta la capacidad de análisis del problema, con los valores discretos de bien o mal definido, mientras que en el eje de abscisas se recoge la variabilidad de la tarea, que representa el número de excepciones esperadas. La Ingeniería del Software, a diferencia de otras formas de ingeniería, presenta problemas de toma de decisión en más de un cuadrante. La definición borrosa del problema y un número alto de excepciones caracterizan las fases de análisis de requisitos y de diseño, que deben servir de base para que la generación de código y prueba se puedan incluir en el cuadrante de la ingeniería. El mayor número de actividades van a recaer en el cuadrante de actividades no rutinarias, lo que va a requerir de personal altamente cualificado, una baja

formalización y centralización, una alta demanda de procesamiento de información y una importante necesidad de coordinación del equipo de trabajo. (Pressman, 2010).

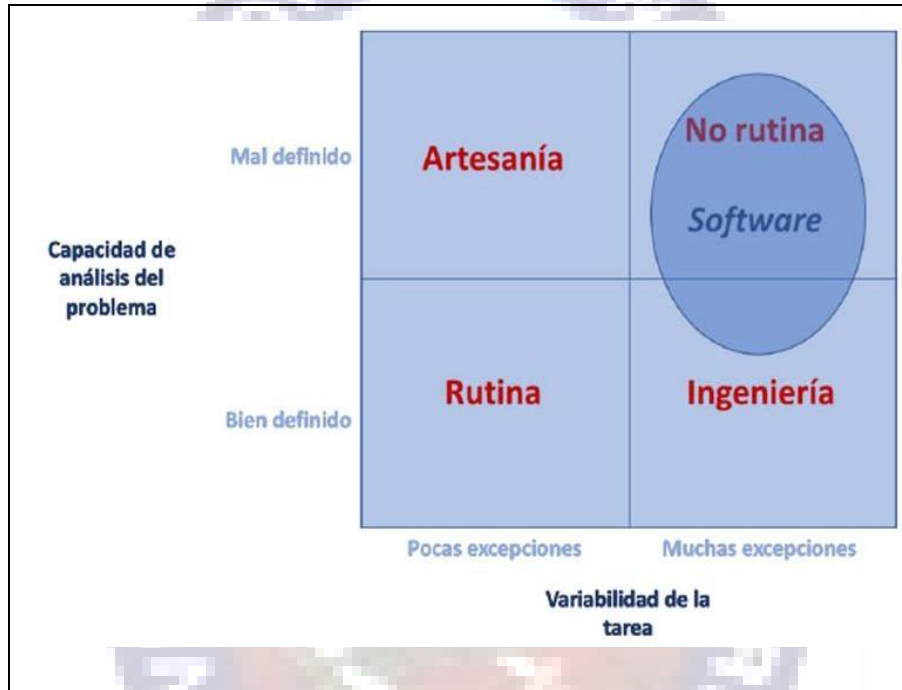


Figura 2.2. Clasificación de la tecnología.

2.2 LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Las TICs son las herramientas necesarias para acceder y manipular datos digitales, a continuación, se explica tres características básicas de las TICs:

- Conjunto de habilidades y competencias, valorándose a las TICs como materia de estudio, procurando determinar los conocimientos y las habilidades que promueven.
- Como herramientas para mejorar el desarrollo de procedimientos ya establecidos.
- Como agentes de cambio capaces de generar nuevas herramientas y técnicas para el desarrollo de actividades, creando nuevos paradigmas a través de la innovación de los procesos. (Pérez, 2002)

Las TICs pueden ser de utilidad para los procesos educativos y de enseñanza, generando transformaciones en la forma en que la educación se imparte y conoce. Ahora, este nuevo escenario exige de la participación de los establecimientos educacionales y de quienes lo regulan, en donde, para empezar, se deben cambiar paradigmas que permitan realizar un cambio que parta desde los cimientos de la educación, creando una nueva que permitan tanto al docente como al estudiante vivir una transición controlada y supervisada.

El desafío de esta transición surge de cómo estas tecnologías permiten innovar el proceso de aprendizaje en sí, y no aferrarse de la tecnología por sí misma. Esto quiere decir: innovar a través de su uso. Las tecnologías pueden moldearse para un usuario en particular, asimismo el usuario puede moldear las tecnologías. Ese es entonces el punto al cual los establecimientos debiesen apuntar; cómo sus propias necesidades moldean un sistema de información que les permita mejorar el proceso de enseñanza y cómo las tecnologías por sí mismas pueden integrarse dada su naturaleza universal. (Fabres, Libuy & Tapia, 2014).

2.3 METODOLOGÍA ÁGIL SCRUM

2.3.1 MARCO CONCEPTUAL DE LA METODOLOGÍA SCRUM

En el año 1986 Takeuchi y Nonaka publicaron el artículo “The New Product Development Game” el cual dará a conocer una nueva forma de gestionar proyectos en la que la agilidad, flexibilidad, y la incertidumbre son los elementos principales.

Nonaka y Takeuchi se fijaron en empresas tecnológicas que, estando en el mismo entorno en el que se encontraban otras empresas, realizaban productos en menos tiempo, de buena calidad y menos costes.

Observando a empresas como Honda, HP, Canon...etc., se dieron cuenta de que el producto no seguía unas fases en las que había un equipo especializado en cada una de ellas, si no que se partía de unos requisitos muy generales y el producto lo realizaba un equipo multidisciplinar que trabajaba desde el comienzo del proyecto hasta el final. (Schwaber y Sutherland, 2013).

Se comparó esta forma de trabajo en equipo, con la colaboración que hacen los jugadores de Rugby y la utilización de una formación denominada SCRUM.

Scrum aparece como una práctica destinada a los productos tecnológicos y será en 1993 cuando realmente Jeff Sutherland aplique un modelo de desarrollo de Software en EASE/Corporation.

En 1996, Jeff Sutherland y Ken Schwaber presentaron las prácticas que se usaban como proceso formal para el desarrollo de software y que pasarían a incluirse en la lista de Agile Alliance.

Scrum es adecuado para aquellas empresas en las que el desarrollo de los productos se realiza en entornos que se caracterizan por tener:

- Incertidumbre: Sobre esta variable se plantea el objetivo que se quiere alcanzar sin proporcionar un plan detallado del producto.
- Esto genera un reto y da una autonomía que sirve para generar una “tensión” adecuada para la motivación de los equipos.
- Autoorganización: Los equipos son capaces de organizarse por sí solos, no necesitan roles para la gestión, pero tienen que reunir las siguientes características:
- Autonomía: Son los encargados de encontrar la solución usando la estrategia que encuentren adecuada.
- Autosuperación: Las soluciones iniciales sufrirán mejoras.
- Auto enriquecimiento: Al ser equipos multidisciplinares se ven enriquecidos de forma mutua, aportando soluciones que puedan complementarse.
- Control moderado: Se establecerá un control suficiente para evitar descontroles. Se basa en crear un escenario de “autocontrol entre iguales” para no impedir la creatividad y espontaneidad de los miembros del equipo.
- Transmisión del conocimiento: Todo el mundo aprende de todo el mundo. Las personas pasan de unos proyectos a otros y así comparten sus conocimientos a lo largo de la organización. (SCRUMstudy, 2010)

Scrum al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones y que en Scrum se llamarán “Sprints”.

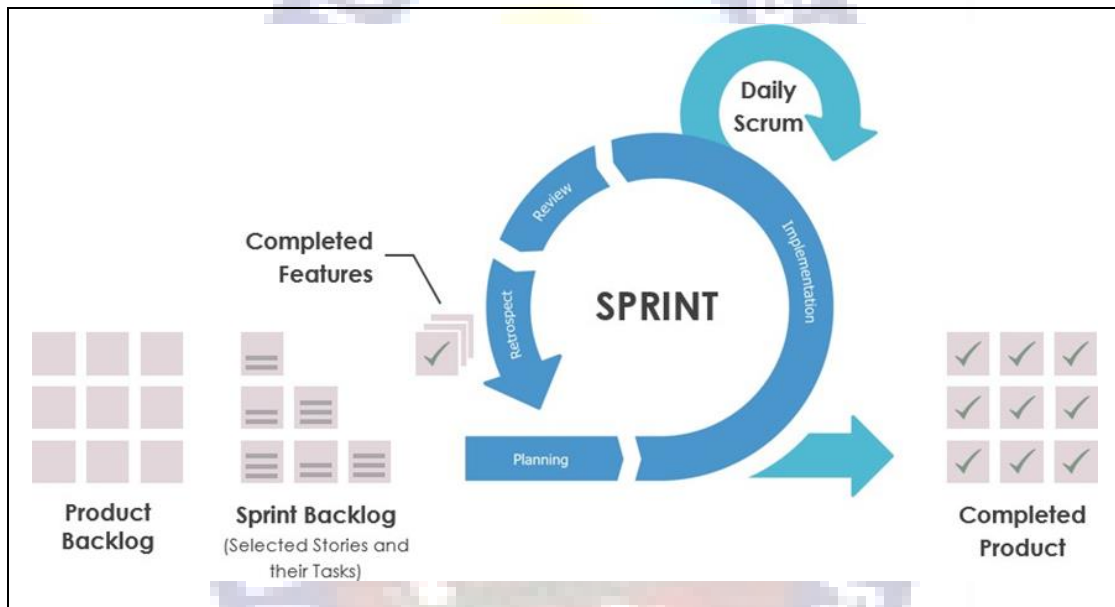


Figura 2.3. Estructura Central de Scrum
Fuente: Visual Paradigm, 2019

2.3.2 EQUIPO DE TRABAJO (ROLES)

2.3.2.1 PRODUCT OWNER

El Product Owner es el encargado de optimizar y maximizar el valor del producto, siendo la persona encargada de gestionar el flujo de valor del producto a través del Product Backlog. Adicionalmente, es fundamental su labor como interlocutor con los stakeholders y sponsors del proyecto, así como su faceta de altavoz de las peticiones y requerimientos de los clientes. Si el Product Owner también juega el rol de representante de negocio, su trabajo también aportará valor al producto.

Tradicionalmente, se ha entendido la labor del Product Owner como un gestor de requisitos o un cliente que se encarga de gestionar el Product Backlog, pero es mucho más que eso. No

solo tiene la responsabilidad de mantener el Product Backlog bien estructurado, detallado y priorizado, sino que además tiene que entender perfectamente cuál es la deriva que se desea para el producto en todo momento, debiendo poder explicar y transmitir a los stakeholders cuál es el valor del producto en el que están invirtiendo.

Con cada Sprint, el Product Owner debe hacer una inversión en desarrollo que tiene que producir valor. Marcar el Sprint Goal de manera clara y acordada con el equipo de desarrollo, hace que el producto vaya incrementando constantemente su valor.

Es fundamental otorgar el poder necesario al Product Owner para que este sea capaz de tomar cualquier decisión que afecte al producto. En el caso de que el Product Owner no pueda tomar estas decisiones sin consultarlas previamente con otra persona, deberá ser investido para tomarlas él mismo, o ser sustituido por esa persona. A su vez, el Product Owner debe convertirse en el altavoz del cliente, en el transmisor de las demandas y del feedback otorgado por los mismos. (Schwaber y Sutherland, 2013).

2.3.2.2 SCRUM MASTER

El Scrum Master tiene dos funciones principales dentro del marco de trabajo: gestionar el proceso Scrum y ayudar a eliminar impedimentos que puedan afectar a la entrega del producto. Además, se encarga de las labores de mentoring y formación, coaching y de facilitar reuniones y eventos si es necesario.

Gestionar el proceso Scrum el Scrum Master se encarga de gestionar y asegurar que el proceso Scrum se lleva a cabo correctamente, así como de facilitar la ejecución del proceso y sus mecánicas. Siempre atendiendo a los tres pilares del control empírico de procesos y haciendo que la metodología sea una fuente de generación de valor.

Eliminar impedimentos esta función del Scrum Master indica la necesidad de ayudar a eliminar progresiva y constantemente impedimentos que van surgiendo en la organización y que afectan a su capacidad para entregar valor, así como a la integridad de esta metodología. El Scrum Master debe ser el responsable de velar porque Scrum se lleve adelante, transmitiendo sus beneficios a la organización facilitando su implementación.

Puede que el Scrum Master esté compartido entre varios equipos, pero su disponibilidad afectará al resultado final del proceso Scrum. (Schwaber y Sutherland, 2013).

2.3.2.3 EQUIPO DE DESARROLLO

El equipo de desarrollo suele estar formado por entre 3 a 9 profesionales que se encargan de desarrollar el producto, autoorganizándose y autogestionándose para conseguir entregar un incremento de software al final del ciclo de desarrollo.

El equipo de desarrollo se encargará de crear un incremento terminado a partir de los elementos del Product Backlog seleccionados (Sprint Backlog) durante el Sprint Planning.

Es importante que en la metodología Scrum todos los miembros del equipo de desarrollo conozcan su rol, siendo solo uno común para todos, independientemente del número de miembros que tenga el equipo y cuales sean sus roles internos. Cómo el equipo de desarrollo decida gestionarse internamente es su propia responsabilidad y tendrá que rendir cuentas por ello como uno solo; hay que evitar intervenir en sus dinámicas.

Habitualmente son equipos ‘cross-funcional’, capaces de generar un incremento terminado de principio a fin, sin otras dependencias externas. (Schwaber y Sutherland, 2013).

2.3.2.4 STAKEHOLDERS

Son todas aquellas personas, organizaciones o empresas cuyo apoyo permiten que una organización exista, sin requerir una relación directa entre estos grupos con el desarrollo de los servicios o productos que la organización desarrolla. (Schwaber y Sutherland, 2013).

Por ejemplo, pueden considerarse como Stakeholders de nuestro producto:

- Clientes finales
- Accionistas
- Proveedores
- Trabajadores de otros departamentos
- Medios de comunicación

2.3.3 REUNIONES

2.3.3.1 PLANIFICACIÓN DEL BACKLOG

Se definirá un documento en el que se reflejarán los requisitos del sistema por prioridades.

En esta fase se definirá también la planificación del Sprint 0, en la que se decidirá cuáles van a ser los objetivos y el trabajo que hay que realizar para esa iteración.

Se obtendrá además en esta reunión un Sprint Backlog, que es la lista de tareas y que es el objetivo más importante del Sprint. (Schwaber y Sutherland, 2013).

2.3.3.2 SEGUIMIENTO DEL SPRINT

En esta fase se hacen reuniones diarias en las que las 3 preguntas principales para evaluar el avance de las tareas serán:

¿Qué trabajo se realizó desde la reunión anterior?

¿Qué trabajo se hará hasta una nueva reunión?

Inconvenientes que han surgido y qué hay que solucionar para poder continuar. (Schwaber y Sutherland, 2013).

2.3.3.3 REVISIÓN DEL SPRINT

Cuando se finaliza el Sprint se realizará una revisión del incremento que se ha generado. Se presentarán los resultados finales y una demo o versión, esto ayudará a mejorar el feedback con el cliente. (SCRUMstudy, 2010).

2.3.4 ELEMENTOS DE SCRUM.

Los elementos que forman a Scrum son:

- Product Backlog: lista de necesidades del cliente.
- Sprint Backlog: lista de tareas que se realizan en un Sprint.
- Incremento: parte añadida o desarrollada en un Sprint, es una parte terminada y totalmente operativa.

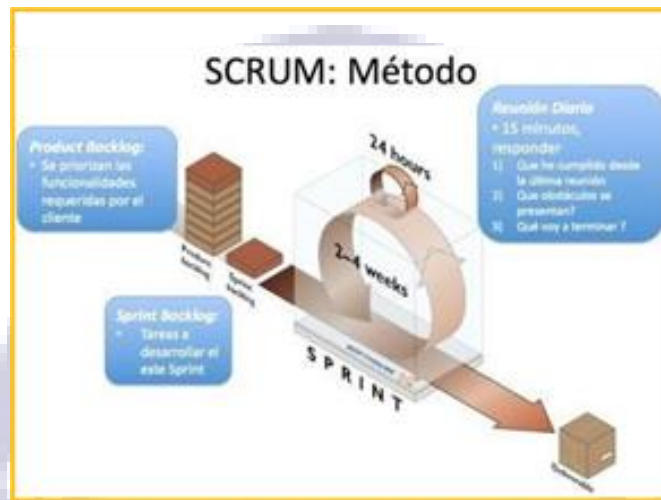


Figura 2.4. Ciclo de desarrollo Scrum.
Fuente: Visual Paradigm, 2019

Es el inventario en el que se almacenan todas las funcionalidades o requisitos en forma de lista priorizada. Estos requisitos serán los que tendrá el producto o los que irá adquiriendo en sucesivas iteraciones.

La lista será gestionada y creada por el cliente con la ayuda del Scrum Master, quien indicará el coste estimado para completar un requisito, y además contendrá todo lo que aporte un valor final al producto. (SCRUMstudy, 2010).

Las características principales de esta lista de objetivos serán:

- Contendrá los objetivos del producto, se suele usar para expresarlos las historias de usuario.
- En cada objetivo, se indicará el valor que le da el cliente y el coste estimado; de esta manera, se realiza la lista, priorizando por valor y coste, se basará en el ROI.
- En la lista se tendrán que indicar las posibles iteraciones y los releases que se han indicado al cliente.
- La lista ha de incluir los posibles riesgos e incluir las tareas necesarias para solventarlos.

2.3.5 FORMATO DE LA PILA DEL PRODUCTO (PRODUCT BACKLOG).

En Scrum, la preferencia por tener documentación en todo momento es menos estricta. Se encuentra más necesario el mantener una comunicación directa con el equipo, por eso se usa como herramienta el Backlog. (García, 2019).

Aunque no hay ningún producto especial a la hora de confeccionar la lista, es conveniente que incluya información relativa a:

- Identificador para la funcionalidad.
- Descripción de la funcionalidad.
- Sistema de priorización u orden.
- Estimación.

Id	Prioridad	Descripción	Est.	Por
1	Muy alta	Plataforma tecnológica	30	AR
2	Muy alta	Interfaz usuario	40	LR
3	Muy alta	Un usuario se registra en el sistema	40	LR
4	Alta	El operador define el flujo y textos de un expediente	60	AR
5	Alta	Etc...	999	XX

Figura 2.5. Ejemplo de un Product Backlog
Fuente: García, 2019

2.3.6 SPRINT BACKLOG

Es la lista de tareas que elabora el equipo durante la planificación de un Sprint. Se asignan las tareas a cada persona y el tiempo que queda para terminarlas.

De esta manera el proyecto se descompone en unidades más pequeñas y se puede determinar o ver en qué tareas no se está avanzando e intentar eliminar el problema.

Requisito	Tarea	Quien	Estado (No iniciada / en progreso / completada)	Día:												
				1	2	3	4	5	6	7	8	9	10			
				Horas	1120	1088	1076	1048	1040	1032	1020	1008	992	972	pendientes	
Requisito A	Tarea 1	Joao	Completada	16	8											
Requisito A	Tarea 4	Laura	Completada	4												
Requisito A	Tarea 5	Laura	Completada	4												
Requisito A	Tarea 3	Gabri	Completada	8												
Requisito A	Tarea 2	Laura	Completada	16	8	4										
Requisito A	Tarea 6	Gabri	Completada	8	8	8										
Requisito A	Tarea 7	Joao	Completada	16	16	16	8									
Requisito A	Tarea 8	Laura	Completada	8	8	8										
Requisito A	Tarea 9	Laura	Completada	8	8	8	8	8								
Requisito A	Tarea 10	Laura	Completada	8	8	8	8	8	8	4						
Requisito A	Tarea 11	Joao	Completada	16	16	16	16	16	16	8						
Requisito B	Tarea 12	Gabri	Completada	16	16	16	16	16	16	16	16	16	16	8		
Requisito B	Tarea 13	Laura	Completada	16	16	16	16	16	16	16	16	16	16	8		
Requisito B	Tarea 14	Joao	En progreso	8	8	8	8	8	8	8	8	8	8	8	4	
Requisito B	Tarea 15	Gabri	En progreso	8	8	8	8	8	8	8	8	8	8	8	8	8
Requisito B	Tarea 16	Laura	En progreso	8	8	8	8	8	8	8	8	8	8	8	8	8
Requisito C	Tarea 17	Joao	No iniciada	4	4	4	4	4	4	4	4	4	4	4	4	4
Requisito C	Tarea 18	Gabri	No iniciada	8	8	8	8	8	8	8	8	8	8	8	8	8
Requisito C	Tarea 19	Laura	No iniciada	16	16	16	16	16	16	16	16	16	16	16	16	16
Requisito C	Tarea 20	Joao	No iniciada	8	8	8	8	8	8	8	8	8	8	8	8	8

Figura 2.6. Ejemplo de Sprint Backlog.
Fuente: García, 2019

2.3.6.1 LISTA DE PRIORIDADES

La lista tiene los siguientes aspectos:

- Es una lista ordenada por prioridades para el cliente.
- Puede haber dependencias entre una tarea y otra, por lo tanto, se tendrá que diferenciar de alguna manera.
- Todas las tareas tienen que tener un coste semejante que será entre 4-16 horas.

2.3.7 INCREMENTO

Representa los requisitos que se han completado en una iteración y que son perfectamente operativos.

Según los resultados que se obtengan, el cliente puede ir haciendo los cambios necesarios y replanteando el proyecto.

2.3.8 PROCESO DE LA METODOLOGÍA

2.3.8.1 PRE – GAME

El proceso comienza con la fase de Pre-game, en la que se realiza de forma conjunta con el cliente una definición sencilla y clara de las características que debe tener el sistema que vaya a ser desarrollado, definiendo las historias de usuario que van a guiar el proceso de desarrollo la cual incluye dos sub-fases:

Planeación

Consiste en la definición del sistema que será construido. Para esto se crea la lista Product Backlog a partir del conocimiento que actualmente se tiene del sistema. En ella se expresan los requerimientos priorizados y a partir de ella se estima el esfuerzo requerido. La Product Backlog List es actualizada constantemente con ítems nuevos y más detallados, con estimaciones más precisas y cambios en la prioridad de los ítems.

Arquitectura

El diseño de alto nivel del sistema se planifica a partir de los elementos existentes en la Product Backlog List. En caso de que el producto a construir sea una mejora a un sistema ya existente, se identifican cambios necesarios para implementar elementos que aparecen en la Product Backlog y el impacto que pueden tener estos cambios. Se sostiene una Design Review Meeting para examinar los objetivos de la implementación y tomar decisiones a partir de la revisión. Se preparan planes preliminares sobre el contenido de cada release.

2.3.2.2 GAME

La fase llamada Game es la parte ágil del Scrum, en esta fase se espera que ocurran cosas impredecibles. Para evitar el caos Scrum define prácticas para controlar las variables técnicas y del entorno, así también como la metodología de desarrollo que hayan sido identificadas y puedan cambiar. Este control se realiza durante los Sprints. Dentro de variables de entorno encontramos: tiempo, calidad, requerimientos, recursos, tecnologías y herramientas de implementación. En lugar de tenerlas en consideración al comienzo del desarrollo, Scrum propone controlarlas constantemente para poder adaptarse a los cambios en forma flexible.

2.3.2.3 POST – GAME

El Post-Game es la fase que contiene el cierre del release. Para ingresar a esta fase se debe llegar a un acuerdo respecto a las variables del entorno por ejemplo que los requerimientos fueron completados. El sistema está listo para ser liberado y es en esta etapa en la que se realiza integración, pruebas del sistema y documentación.

El sprint es por tanto el núcleo central que proporciona la base de desarrollo iterativo e incremental tal como se ve en la Figura 2.1

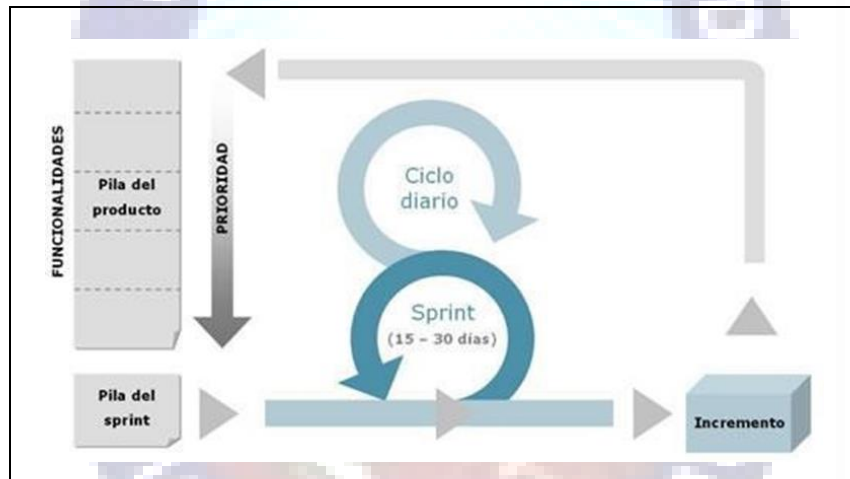


Figura 2.7. Visión general del proceso de la metodología Scrum
Fuente: Schwaber Beedle, 2006

2.4 INGENIERÍA WEB

La ingeniería web se debe al crecimiento desenfrenado que está teniendo la web está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas en que se presenta ha hecho que las personas tiendan a realizar todas sus actividades por esta vía.

El desarrollo de aplicaciones web posee determinadas características que lo hacen diferente del desarrollo de aplicaciones o software tradicional y sistemas de información.

Según S. Murugesan, Y. Deshpande, S., promotores iniciales del establecimiento de la Ingeniería web como nueva disciplina, dan la siguiente definición:

Es el proceso utilizado para crear, implantar y mantener aplicaciones y sistemas web de alta

calidad. Esta breve definición nos lleva a abordar un aspecto clave de cualquier proyecto como es determinar qué tipo de proceso es más adecuado en función de las características del mismo.

El desarrollo de aplicaciones web posee determinadas características que lo hacen diferente del desarrollo de aplicaciones o software tradicional y sistemas de información. La ingeniería de la web es multidisciplinar.

2.4.1 PROCESO DE INGENIERÍA WEB

Según Pressman, las actividades que formarían parte del marco de trabajo incluirían las tareas abajo mencionadas. Dichas tareas serían aplicables a cualquier aplicación web, independientemente del tamaño y complejidad de la misma.

Comunicación con el cliente: La comunicación con el cliente se caracteriza por medio de dos grandes tareas: el análisis del negocio y la formulación. El análisis del negocio define el contexto empresarial-organizativo para las WebApps y otras aplicaciones de negocio. La formulación es una recopilación de requisitos que involucran a todos los participantes.

Planeación: Se crea el plan del proyecto para el incremento de la WebApp. El plan consiste de una definición de tareas y un calendario de plazos respecto al período establecido para el desarrollo del proyecto.

Modelado: Las labores convencionales de análisis diseño de la ingeniería del software se adaptan al desarrollo de las WebApp, se mezclan y luego se funden en una actividad de modelado de la web. El intento es desarrollar análisis rápido y modelos de diseño que definan requisitos y al mismo tiempo representen una WebApp que los satisfará.

Construcción: Las herramientas y la tecnología IWeb se aplican para construir la WebApp que se ha modelado. Una vez que se construye el incremento de WebApp se dirige a una serie de pruebas rápidas para asegurar que se descubran los errores en el diseño. (Guerrero, Ucan, Menendez, 2014)

2.5 METODOLOGÍA UWE

UWE es una metodología basada en el Proceso Unificado y UML para el desarrollo de aplicaciones web, cubre todo el ciclo de vida de las aplicaciones web. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de una meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos. (Gonzales, 2013)

2.5.1 FASES DE LA METODOLOGÍA UWE

Las fases de la metodología UWE, son procesos o actividades que se utilizan y permiten identificar las necesidades de la aplicación o sistema web a desarrollar; estas actividades se describen y representan en cuatro fases que son:

- Diagrama de casos de uso (análisis de requisitos)
- Modelo navegacional
- Modelo conceptual
- Diagrama de clases

2.5.1.1 DIAGRAMA DE CASOS DE USO (ANÁLISIS DE REQUISITOS)

Como en otras metodologías, la primera fase o actividad es la del análisis de requisitos

funcionales, que permite visualizar los procesos y funciones que debe cumplir el sistema web, esta fase se ve reflejada en los casos de uso. (González, 2013)

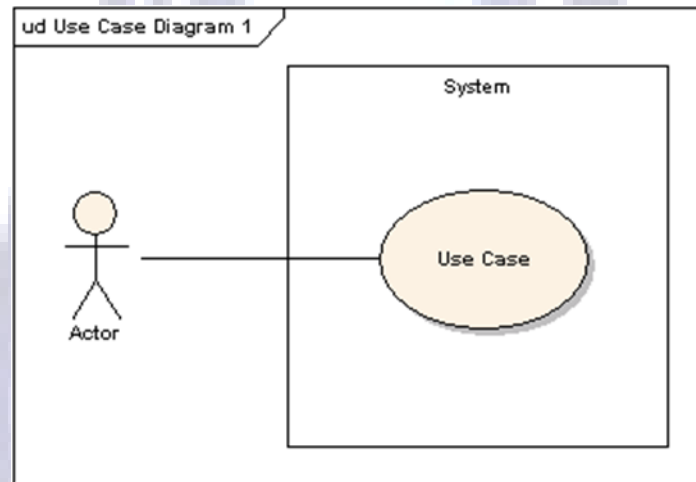


Figura 2.8. Análisis de caso de uso
Fuente: González, 2013

2.5.1.2 MODELO NAVEGACIONAL

El modelo navegacional no es solo útil para la generación de la documentación de la estructura de la aplicación, sino que también permite mejorar la estructura de navegabilidad.

El modelo de la navegación comprende de:

- El modelo de espacio de navegación que especifica qué objetos puede ser visitados a través de la aplicación web.
- El modelo de estructura de navegación que define como se alcanzan estos objetos a través de la web.
- En el proceso de construir el modelo espacial de navegación las decisiones del diseñador están basadas en el modelo conceptual y los requisitos de la aplicación definidos en el modo de caso de uso.

Cuando hablamos de un sistema web, es necesario conocer la relación y los enlaces entre las páginas web, es por eso que en la fase de diseño se describen a través de diagramas la

navegación del sistema cumpliendo con lo que se diseñó en los casos de uso, los elementos que se utilizan para el diseño de diagramas son:










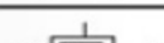
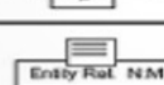
ICONO	DESCRIPCION
	Hiperlance jerarquico inferido
	Hiperlance jerarquico inferido
	Indice
	Visita guiada
	Visita guiada indexada
	Acceso al calorio
	Acceso simultaneo
	Indice Multiple
	Visita guiada multiple
	Visita guiada indexada multiple
	Ejemplo de acceso a partir de una relacion N:M, en este caso utilizando un indice (podria ser cualquier otra primitiva de acceso simple)

Tabla 2.1. Elementos de modelo navegacional

Fuente: Instituto de Informática, 2011.

En la figura a continuación se muestra un ejemplo del modelo navegacional:

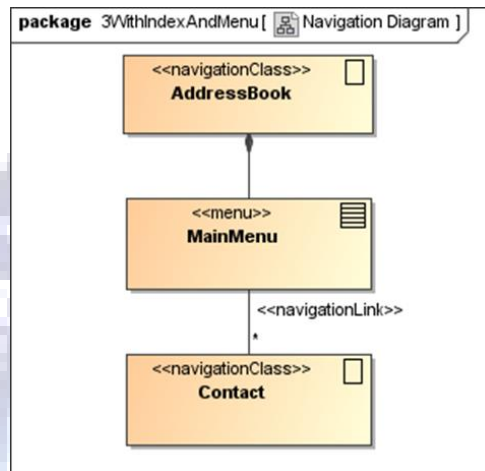


Figura 2.9. Modelo navegacional UWE
Fuente: Instituto de Informática, 2011

2.5.1.3 MODELO CONCEPTUAL

El diagrama conceptual se basa en el análisis de requisitos del paso anterior. Esto incluye los objetos involucrados entre los usuarios y la aplicación. Este modelo propone construir un modelo de clases con estos objetos, ignorando los aspectos de navegación: Presentación e Interacción, que serán tratados posteriormente. Los principales elementos de modelado son; las clases, asociaciones y paquetes. (González, 2013)

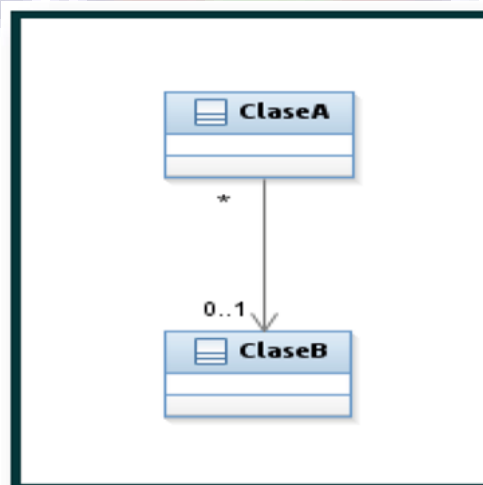


Figura 2.10. Modelo conceptual
Fuente: M. González, 2013

2.5.1.4 DIAGRAMA DE CLASES

Un diagrama de clases es un diagrama estructural utilizado para describir la arquitectura de un sistema. Destaca las distintas clases que puede contener el sistema junto con sus atributos, operaciones específicas o las relaciones que se desarrollan entre los objetos de ese sistema. Los ingenieros de software utilizan habitualmente los diagramas de clases, ya que proporcionan descripciones detalladas de todos los elementos que requiere el sistema recién construido.

Integrar los diagramas de clases UML en el flujo de trabajo de una organización aporta una serie de beneficios que tendrán un impacto positivo en los proyectos a los que se asocian:

- Visualización completa de datos para sistemas de información simples o complejos.
- Buena comprensión de la arquitectura de un sistema.
- Visualización precisa de las particularidades y los requisitos de un sistema, que pueden comunicarse rápidamente a los principales interesados.

2.6 DETERMINACIÓN DE PUNTO FUNCIÓN

La métrica del punto función es un método utilizado en ingeniería del software para medir el tamaño del software. Fue definida por Allan Albrecht, de IBM, en 1979 y pretende medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada para la construcción y explotación del software, y también ser útil en cualquiera de las fases de vida del software, desde el diseño inicial hasta la implementación y mantenimiento. IFPUG (2000)

2.6.1 ENTRADA DE DATOS EN LÍNEA

Esta característica cuantifica la entrada de datos on-line provista por la aplicación.

GRADO	DESCRIPCIÓN
0	Todas las transacciones son procesadas en modo batch
1	De 1% al 7% de las transacciones son entradas de datos on-line
2	De 8% al 15% de las transacciones son entradas de datos on-line
3	De 16% al 23% de las transacciones son entradas de datos on-line
4	De 24% al 30% de las transacciones son entradas de datos on-line
5	Más del 30% de las transacciones son entradas de datos on-line

Tabla 2.2. Entrada de Datos en Línea
Fuente: IFPUG (2000)

2.6.2 INTERFACE CON EL USUARIO

Las funciones on-line del sistema hacen énfasis en la amigabilidad del sistema y su facilidad de uso, buscando aumentar la eficiencia del usuario final. El sistema posee:

- Ayuda para la navegación (teclas de función, accesos directos y menús dinámicos).
- Menús
- Documentación y ayuda on-line.
- Movimiento automático del cursor.
- Scrolling vertical y horizontal.
- Impresión remota (a través de transacciones on-line).
- Teclas de función preestablecidas.
- Ejecución de procesos batch a partir de transacciones on-line.
- Selección de datos vía movimiento del cursor en la pantalla.
- Utilización intensa de campos en video reverso, intensificados, subrayados,

coloridos y otros indicadores.

- Impresión de la documentación de las transacciones on-line por medio de hard copy.
- Utilización del mouse
- Menús pop-up
- El menor número de pantallas posibles para ejecutar las funciones del negocio.
- Soporte bilingüe (el soporte de dos idiomas, cuente como cuatro ítems)
- Soporte multilingüe (el soporte de más de dos idiomas, cuente como seis ítems).

GRADO	DESCRIPCIÓN
0	Ningún de los ítems descritos
1	De uno a tres de los ítems descritos
2	De cuatro a cinco de los ítems descritos
3	Más de cinco de los ítems descritos, no hay requerimientos específicos del usuario en cuanto a amigabilidad del sistema
4	Más de cinco de los ítems descritos, y fueron descritos requerimientos en cuanto a amigabilidad del sistema suficientes para generar actividades específicas incluyendo factores tales como minimización de la digitación
5	Más de cinco de los ítems descritos y fueron establecidos requerimientos en cuanto a la amigabilidad suficientes para utilizar herramientas especiales y procesos especiales para demostrar anticipadamente que los objetivos fueron alcanzados

Tabla 2.3. Interfase con el Usuario

Fuente: IFPUG (2000)

2.6.3 REUSABILIDAD DE CÓDIGO

La aplicación y su código serán o fueron proyectados, desarrollados y mantenidos para ser utilizados en otras aplicaciones.

GRADO	DESCRIPCIÓN
0	No presenta código reutilizable
1	Código reutilizado fue usado solamente dentro de la aplicación
2	Menos del 10% de la aplicación fue proyectada previendo la utilización posterior del código por otra aplicación
3	10% o más de la aplicación fue proyectada previendo la utilización posterior del código por otra aplicación
4	La aplicación fue específicamente proyectada y/o documentada para tener su código fácilmente reutilizable por otra aplicación y la aplicación es configurada por el usuario a nivel de código fuente
5	La aplicación fue específicamente proyectada y/o documentada para tener su código fácilmente reutilizable por otra aplicación y la aplicación es configurada para uso a través de parámetros que pueden ser alterados por el usuario

Tabla 2.4. Reusabilidad de Código
Fuente: IFPUG (2000)

2.7 TÉCNICA ISO 9126

El estándar ISO/IEC 9126: 2001 presenta un marco conceptual para el modelo de calidad y define un conjunto de características y subcaracterísticas, las que debe cumplir todo producto software.

El estándar ISO-9126 establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de las siete características básicas: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad y satisfacción; cada una de ellas se detalla a través de un conjunto de subcaracterísticas que permiten profundizar en la evaluación de la calidad de productos de software.

En relación con el modelo de calidad del producto de software, el estándar ISO/IEC 9126 (2001) indica lo siguiente:

- Presenta un modelo de calidad del software, estructurado en características y subcaracterísticas.
- Proporciona métricas externas para medir los atributos de las seis características de calidad externa definidas en la ISO/IEC 9126-1 (2001) y una explicación de cómo aplicar las métricas de calidad de software.
- Proporciona métricas internas para medir atributos de las seis características de calidad interna definidas en la ISO/IEC 9126-1 (2001).
- Define métricas de calidad en uso para medir los atributos definidos en la ISO/IEC 9126-1 (2001).

2.7.1 IMPORTANCIA DE LA NORMA ISO/IEC 9126: 2001

Previo a la creación de esta norma existían diferentes estándares ISO relacionados con la calidad de software, incluyendo la ISO/IEC 9001:2000 y la ISO/IEC 12207:1995 las cuales buscan la adopción de un enfoque basado en procesos para mejorar la eficacia de un sistema de gestión de calidad y definir los procesos del ciclo de vida del software respectivamente. Sin embargo, estas normas no permitían resolver la dificultad que hay para cuantificar y calificar la mayoría de las características que definen un software.

Es allí donde recae la importancia de la norma ISO/IEC 9126:2001, debido a que al basarse en el modelo de McCall establece una guía acerca de los elementos que deben considerarse al evaluar un software para así generar métricas propias que guíen tanto el desarrollo como su valoración.

Las características de calidad de un software según la ISO/IEC 9126: 2001 son:

- **Funcionalidad:** se evalúa la adecuación, el cumplimiento funcional, idoneidad, corrección, interoperabilidad, conformidad y seguridad de acceso. Por lo que es posible afirmar que la funcionalidad determina la capacidad del software de funcionar en términos de lo que el usuario necesita, de interactuar con otros sistemas y que permita el acceso de diferentes personas pero que cumpla con las regulaciones de las leyes de protección de datos.

- **Confiabilidad:** se tienen en cuenta aspectos como la capacidad y facilidad de recuperación, la mitigación de fallos, cantidad de tiempo que el software está disponible para su uso y la tolerancia. Por lo que esto tiene en cuenta todo lo relacionado a los fallos que podría dar el producto de software.
- **Usabilidad:** mide el grado en que el software es fácil de usar, qué tan intuitivo es, el manejo que el usuario le da al sistema y si este presenta menús sencillos, lectura de textos ágil, cuenta con funciones de forma clara y puntual, entre otros.
- **Eficiencia:** analiza y mide la capacidad en que el software hace óptimo el uso de los recursos del sistema, en términos de tiempo de uso y recursos de los cuales dispone.
- **Facilidad de mantenimiento:** la facilidad con que una modificación puede ser realizada, la capacidad para hacerle pruebas de rendimiento, regresión, accesibilidad, inspección de código y toda la ingeniería de requerimientos. Esto hace que el producto de software sea escalable ya que es posible hacerle mejoras constantes sin que este no tenga ningún problema al querer realizarle alguna modificación o incorporación.
- **Portabilidad:** la facilidad con que el software puede ser llevado de un entorno a otro. Proporciona facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio y otros aspectos que lo hacen un sistema que garantiza portabilidad.
- **Satisfacción:** se tiene en cuenta el cumplimiento de todos los requerimientos para los que fue desarrollado el software, basado en las expectativas del cliente final. Estas expectativas se definen de acuerdo con las necesidades, y estas se basan en la eficiencia y efectividad en coherencia para lo que fue hecho el producto.

Las normas y estándares para proyectos TI están basados en las características de calidad definidas por la norma ISO/IEC 9126 y algunas otras como la ISO-9000 por ejemplo, la cual evalúa todos los recursos humanos técnicos y procedimientos administrativos que requieren del cumplimiento de algunas especificaciones donde se busca la satisfacción del cliente.

Como se puede notar, las normas no solo son requisitos que se deben cumplir para estar alineado con los requerimientos de la industria, sino que son herramientas que ofrecen importantes beneficios para sus clientes. En este sentido, garantizar todas las premisas basadas en las normativas internacionales y nacionales del desarrollo de productos digitales puede hacer parte de sus prácticas de aseguramiento de la calidad del software.

2.8 MODELO DE ESTIMACIÓN COCOMO II

El modelo COCOMO introduce una jerarquía de modelos de estimación, COCOMO original se ha convertido en uno de los modelos de estimación de coste del software más utilizados y estudiados en la industria. El modelo original ha evolucionado a un modelo de estimación más completo llamado COCOMO II y tratan las áreas:

- Modelo de composición de aplicación
- Modelo de fase de diseño previo
- Modelo de fase posterior a la arquitectura

Al igual que todos los modelos de estimación de software. El modelo COCOMO II descrito antes necesita información del tamaño. Dentro de la jerarquía del modelo hay tres opciones de tamaño distintas: puntos objeto, puntos de función, y líneas de código fuente. El modelo de composición de aplicación COCOMO II utiliza los puntos objeto. El punto objeto es una medida indirecta de software que se calcula utilizando el total de pantallas (de la interface de usuario), informes, y componentes que probablemente se necesiten para construir la aplicación. Se clasifican en uno de los tres niveles de complejidad (esto es, básico, intermedio, o avanzado) utilizando los criterios. (Gómez y López, 2010)

CAPÍTULO III

3. MARCO APLICATIVO

Este capítulo tiene como finalidad describir el desarrollo del sistema web de administración, control y gestión de historiales clínicos del Consultorio dental Ortho-Braes. Haciendo uso de la metodología SCRUM como se describió en el anterior capítulo tiene como función la realización del proyecto partiendo de una lista de tareas a realizar, para la gestión de desarrollo del sistema y la metodología UML para el modelado.

Por esta razón el trabajo se ha dividido en una serie de SPRINTS teniendo como las siguientes fases: Pre-Game, Game, Post-Game.

En la fase de Pre-Game se crea el Product Backlog y se identificaran los roles de usuario. Posteriormente en la fase del Game, se realiza la planeación del Sprint la cual tendrá 6 iteraciones “Sprint”, cada sprint estará desarrollado con la metodología UWE la cual está compuesta por 4 fases:

- Diagrama de casos de uso respectivos
- Modelo navegacional que muestra las opciones de navegación y procesos partiendo de un proceso básico
- Diagrama conceptual.
- Diagrama de clases.

Estos dos últimos diagramas se muestran después del desarrollo de los 6 sprints

En la fase del Post-Game, se tiene el sistema integrado completamente, se muestran las pantallas del sistema con sus características.

En la figura 3.1 se puede observar gráficamente el modelo del proceso que se plantea utilizar para el presente proyecto de grado.

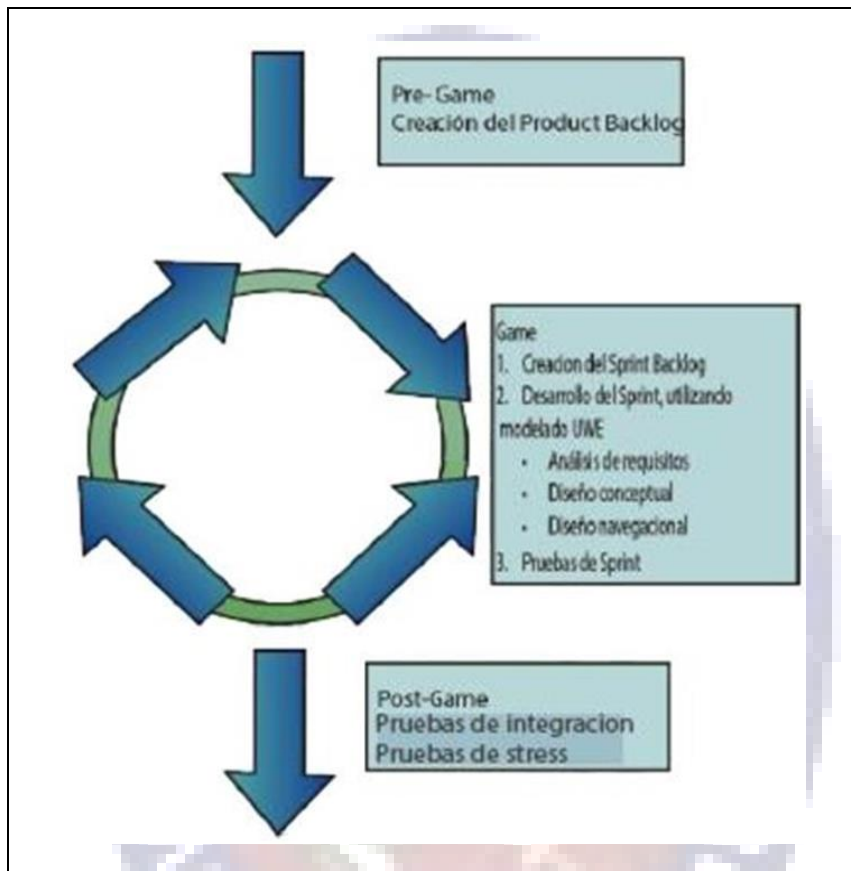


Figura 3.1. Modelo de procesos del proyecto

Fuente: Maximilians, 2015

3.1 ANÁLISIS DE LA SITUACIÓN ACTUAL

El consultorio dental Ortho-Braes actualmente no cuenta con un sistema de información en el cual pueda almacenar y consultar los datos de sus pacientes lo cual es un problema al momento de ofrecer el servicio.

El registro de citas se realiza en hojas de papel o documentos Excel lo cual dificulta a la organización del personal médico del consultorio médico. Optimizar el seguimiento y control de los tratamientos de cada paciente mediante fotos y radiografías digitales.

Los tratamientos que se realizan en múltiples sesiones generan un exceso de recordatorios para el odontólogo que trata a varios pacientes.

Como en cualquier institución relacionada con la salud en el consultorio se tiene los historiales clínicos de cada paciente los cuales son manejados en hojas de papel lo que genera gran cantidad de papelería.

HISTORIA CLÍNICA ODONTOLÓGICA

HC: _____ Fecha: _____ Hora: _____

I.- ANAMNESIS:

FILIACIÓN:
 Nombre del Paciente: _____ Edad: _____ Sexo: _____
 Lugar y fecha de nacimiento: _____
 Dirección: _____
 Procedencia: _____ Ocupación: _____
 Viajes en el último año: _____
 Teléfono: _____
 En caso de Emergencia comunicarse a: _____

RIESGOS: _____

MOTIVO DE CONSULTA:

ENFERMEDAD ACTUAL:
 Tiempo de enfermedad: _____ Inicio: _____ Curso: _____
 Signos y síntomas principales: _____
 Relato de la enfermedad: _____

FUNCIONES BIOLÓGICAS

- Apetito: _____
- Sed: _____
- Sueño: _____
- Sudor: _____
- Peso: _____
- Orina: _____
- Estado de ánimo: _____

ANTECEDENTES PERSONALES:
 Si la respuesta es positiva en alguno de los siguientes items colocar (SI), caso contrario se consigna (NO):

a.- Generales:

- Residencias anteriores: _____
- Ocupaciones anteriores: _____
- Vivienda: Material: _____ Agua potable: _____ Electricidad: _____ Desagüe: _____
- Crianza de animales: _____
- Ventilación: _____ N° Habitaciones: _____ N° de Habitantes: _____
- Alimentación: _____

b.- Fisiológicos

- Parto: _____ Eutócico: _____ Distócico: _____ Domiciliario: _____ Hospitalario: _____
- Desarrollo físico: Normal: _____ Anormal: _____
- Desarrollo psíquico: Normal: _____ Anormal: _____

Gineco-Obstétricos:

- Menarquia: _____ FUM: _____
- Dismenorrea: _____ Lactancia: _____
- Int. Quirúrgicas: _____

c.- A. Patológicos:

Hipertensión Arterial ()	Otra enfermedad Cardiovascular ()
Diabetes ()	Otra enfermedad endocrinológica ()
Asma ()	Hepatitis ()
Enfermedad hepática ()	Enfermedad renal ()
TBC ()	Otra enfermedad infecciosa ()
Enfermedad hemorrágica ()	Otras enfermedades ()
Ingesta de medicamentos ()	

Reacción alérgica a:

Fármacos ()	Alimentos ()
Cirugías previas ()	
Hospitalizaciones previas ()	
Hábitos nocivos ()	Tabaco () Alcohol () Drogas ()

Alteraciones de la esfera sexual: RSP: ... RSH: ... N° parejas en el último año: _____
 Está gestando () Mes ()
 Ampliación: (detallar diagnósticos, lugar, fechas, tratamientos)

d.- Estomatológicos:
 Si la respuesta es positiva en alguno de los siguientes items colocar (SI), caso contrario se consigna (NO):
 Última visita a dentista: _____
 Tratamientos odontológicos previos: _____
 Infiltraciones de anestesia previas: ()
 Reacciones adversas a la infiltración de anestesia () Cuál: _____
 Exodoncias y Cirugías bucales ó maxilofaciales previas ()
 Ha tenido complicaciones después de la exodoncia () Cuál: _____

Figura 3.2. Modelo anterior de historia clínica del consultorio dental Ortho-Braes

3.2 IDENTIFICACIÓN DE ROLES SCRUM

El grupo de trabajo para el presente proyecto estará conformado de la siguiente manera. (Ver tabla 3.1)

ROL		NOMBRE
PRODUCT OWNER		Dr. Brayan Almazán Garate
SCRUM MASTER		Phd. Javier Hugo Gustavo Reyes Pacheco
SCRUM TEAM	Analista	Diego Vladimir Mamani Nina
	Diseñador	
	Desarrollador	
STAKEHOLDERS		Consultorio Dental Ortho-Braes

Tabla 3.1. Identificación de Roles de Scrum

3.3 FASE DE PRE-GAME

Esta fase comprende el inicio del proyecto teniendo como propósito identificar las necesidades del sistema y la documentación. También podríamos decir que esta etapa es para la recolección de tareas que requiere el consultorio dental y la importancia que tiene cada una de ellas.

Es resultado de esta etapa fue el Product Backlog que es una etapa del scrum en el cual se registran las tareas, se las divide en módulos y se las ordena por prioridad según lo que considere prioritario el SCRUM Team.

Se elabora la Product Backlog (Pila del Producto) en la cual se fijarán los requerimientos funcionales, y se identificarán los roles de las personas que harán uso del sistema.

3.3.1 CREACIÓN DEL PRODUCT BACKLOG (PILA DE PRODUCTO)

En la siguiente tabla se presenta el Product Backlog con sus respectivos módulos, la estimación en horas y ordenada por prioridad.

	MÓDULOS	ESTIMACIÓN (HORAS)	PRIORIDAD
1	Módulo de autenticación de usuarios.	128	1
2	Módulo de control de pacientes.	160	2
3	Módulo de control de tratamientos y piezas dentales (odontograma).	136	3
4	Módulo de manejo de historiales clínicos.	120	4
5	Módulo de control de costo de tratamientos, pagos entre otros.	112	5
6	Módulo de control de citas médicas.	168	6

Tabla 3.2. Product Backlog

3.3.2 IDENTIFICACIÓN DE ROLES DE USUARIO

En la siguiente tabla se observa los roles de usuario.

ROLES	TIPO
Médico odontólogo	Usuario encargado de gestionar información de los pacientes

Tabla 3.3. Identificación de los roles de usuario

3.4 FASE DEL GAME

En esta fase se realizó el desarrollo de los Sprints ya definidos anteriormente en el Product Backlog, para el presente proyecto tenemos 6 iteraciones.

Se utilizó los modelos y diagramas de la metodología UWE. En el desarrollo de los sprints se detallan los siguientes diagramas: Diagrama de Casos de uso, Modelo Navegacional, Diagrama Conceptual y Diagrama de clases.

3.4.1 SPRINT 1: MÓDULO DE AUTENTICACIÓN DE USUARIO

Basándonos en la metodología scrum, lo primero que se concretó fue la reunión de planificación del primer sprint. Esta reunión se la efectuó con el usuario o Product Owner de manera presencial. El cuál tenía todo el conocimiento necesario para indicar las tareas que se realizarán y con el que se trabajará durante el resto del proyecto de manera presencial.

También es importante resaltar que el avance de cada sprint se le presentará al Product Owner de manera presencial esperando el visto bueno para continuar con los demás Sprints que generalmente dependen de los Sprints anteriores para su correcto funcionamiento.

3.4.1.2 PLANIFICACIÓN DEL SPRINT 1

En el presente Sprint se detallará los requerimientos y las tareas iniciales que pertenecen al módulo de inicio de sesión.

En la siguiente tabla se muestran las tareas planificadas para este Sprint y que fueron concluidas como se muestra a continuación.

SPRINT 1: MÓDULO DE AUTENTICACIÓN DE USUARIO		
Sprint	Duración en horas	Días de trabajo
1	128	16 días hábiles
SPRINT BACKLOG		
Tareas	Tipo	Estado
Planificación y análisis de los requerimientos del sprint	Planificación	Terminado
Diseño de la base de datos para el sistema web	Desarrollo	Terminado
Modelo conceptual	Diseño	Terminado
Modelo navegacional	Modelado	Terminado
Diseño de la interfaz gráfica para la autenticación de usuario	Desarrollo	Terminado
Seguridad y control de acceso al sistema	Desarrollo	Terminado

Tabla 3.4. Primer Sprint Backlog

3.4.1.2 ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 1

A continuación, los casos de uso del primer sprint.

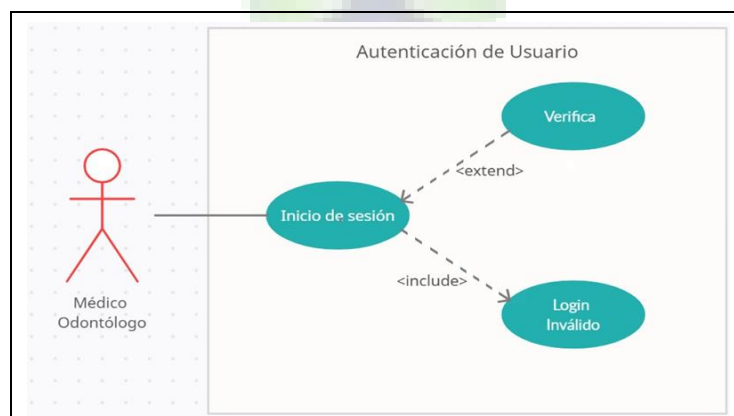


Figura 3.3. Caso de uso – Autenticación de usuario

En este sprint se observa la autenticación de usuario en este caso el médico odontólogo en el momento de ingresar al sistema.

Nombre		Autenticación de usuario
Actores		Médico odontólogo
Descripción		El caso de uso comienza cuando el médico odontólogo quiere ingresar al sistema.
Flujo principal	Evento actor	Evento sistema
	1.- El médico odontólogo quiere ingresar al sistema. 3.- El médico odontólogo introduce los datos.	2.- El sistema despliega el formulario de login. 4.- El sistema valida los datos ingresados.
		5.- El sistema despliega mensaje de confirmación.
Escenarios alternativos		
Alternativa 1	Si en el escenario 4 el sistema encuentra datos que no concuerdan con los requisitos entonces el sistema despliega el mensaje de error.	
Post condición	Los datos del registro deben ser almacenados en la base de datos.	

Tabla 3.5. Especificación de casos de uso - Autenticación de usuario

3.4.1.3 MODELO NAVEGACIONAL DEL SPRINT 1

El modelo navegacional muestra las opciones de navegación y procesos.

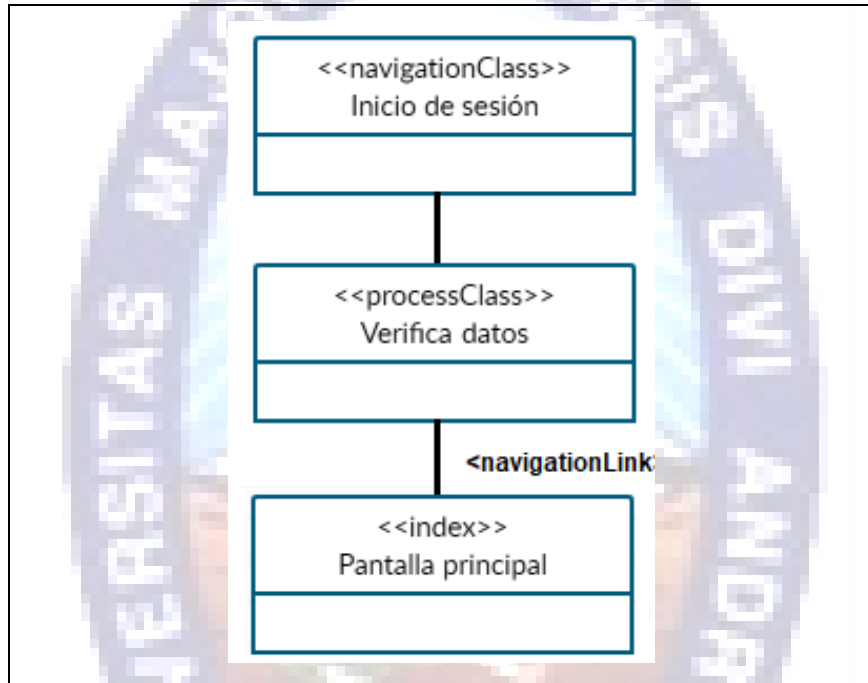


Figura 3.4. Modelo navegacional – Autenticación de usuario

3.4.1.4 PANTALLAS DEL SPRINT 1

A continuación, se puede observar la página de inicio de sesión en la cual el médico odontólogo ingresa los datos de usuario para poder interactuar con los procesos que ofrece el sistema web, ver figura 3.5.

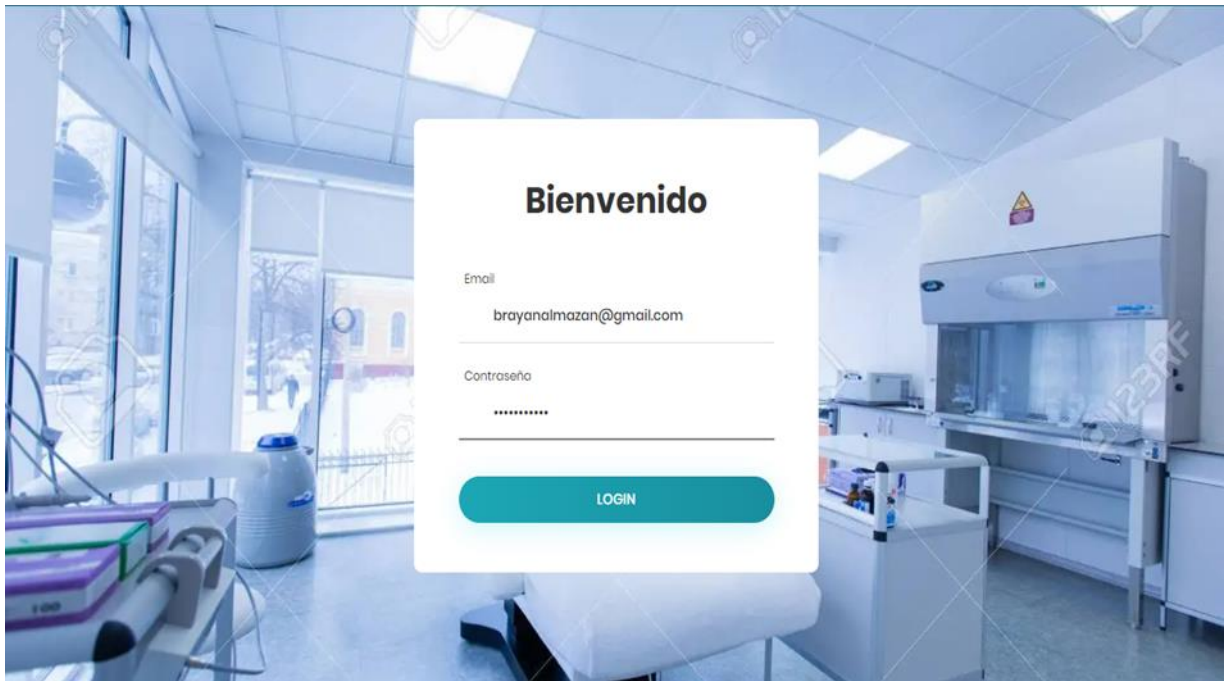


Figura 3.5. Página de inicio del sistema

A continuación, se observará la página inicial administrativa.

Almadent Odontólogo

Brayan Almazan Garate

Agenda

Mostrar Fines de Semana

14 - 20 mar 2022

	Jun 14/3	mar 15/3	mié 16/3	jue 17/3	vie 18/3	sáb 19/3	dom 20/3
all-day							
8			8:00 - 9:00 Brayan Almazan				
9	9:00 - 10:30 Carlos			9:00 - 10:00 Sergio	9:00 - 10:00 Roger		
10	10:30 - 12:00 José		10:00 - 12:00 Choco				
11							
12		12:00 - 13:00 Joe					
13						12:00 - 13:00 Diego Mamani	

Figura 3.6. Página inicial administrativa

3.4.1.5 PRUEBA UNITARIA DEL SPRINT 1

La prueba unitaria a continuación sirve para comprobar que el módulo funcione correctamente.

PRUEBA 1 - MÓDULO DE AUTENTICACIÓN DE USUARIO	
Descripción	Al presionar ingresar, el médico ingresa al sistema.
Objetivos	1. Que el médico ingrese al sistema al presionar ingresar. 2. Abrir la pantalla principal del sistema.
Condiciones	Conexión a internet desde cualquier medio (Pc, Celular, iPhone, Tablet).
Resultado esperado	Que se abra satisfactoriamente el sistema
Resultados obtenidos	El sistema inicia correctamente.

Tabla 3.6. Prueba unitaria - Sprint 1

3.4.2 SPRINT 2: MÓDULO DE CONTROL DE PACIENTES

Una vez concluido el primer sprint se dio inicio al sprint 2. En el desarrollo de este sprint no se presentaron complicaciones por lo cual se avanzó conforme a lo planeado en la lista priorizada.

La comunicación entre el usuario y el desarrollador se incrementó y se realizaron pruebas entre ambas partes para resolver cualquier tipo de cuestión y aclarar algunas dudas sobre el funcionamiento de este sprint.

También se hizo evidente la satisfacción por parte del médico odontólogo con la conclusión del primer sprint.

3.4.2.1 PLANIFICACIÓN DEL SPRINT 2

En este sprint se procedió a realizar un formulario para el registro de los datos de los pacientes como ser: nombre, apellidos, teléfono, lugar fecha de nacimiento, entre otros.

También se desarrolló una tabla donde se muestran los datos principales del paciente, en esta tabla el médico odontólogo puede realizar ciertas acciones como ordenar la tabla por nombre, apellido, carnet y edad, también se pueden filtrar o buscar datos de los pacientes escribiendo palabras o números y en dicha tabla se desplegarán los resultados más aproximados, por otro lado en la tabla se muestran las opciones de editar los datos de un paciente y eliminar los registros y la información de un paciente. Con estas acciones el médico tendrá un fácil acceso a la información actual de paciente como ser, próximas citas, tratamientos realizados, información detallada de todos los datos proporcionados, entre otros.

En la siguiente tabla se muestra el Sprint Backlog para este módulo.

SPRINT 2: MÓDULO DE CONTROL DE PACIENTES		
Sprint	Duración en horas	Días de trabajo
2	160	20 días hábiles
SPRINT BACKLOG		
Tareas	Tipo	Estado
Planificación y análisis de los requerimientos del sprint	Planificación	Terminado
Modelo conceptual	Diseño	Terminado
Modelo navegacional	Modelado	Terminado
Diseño de la interfaz gráfica para el formulario de registro de pacientes y tabla de información de pacientes	Desarrollo	Terminado

Tabla 3.7. Segundo Sprint Backlog

3.4.2.2 ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 2

A continuación, los casos de uso del segundo sprint.

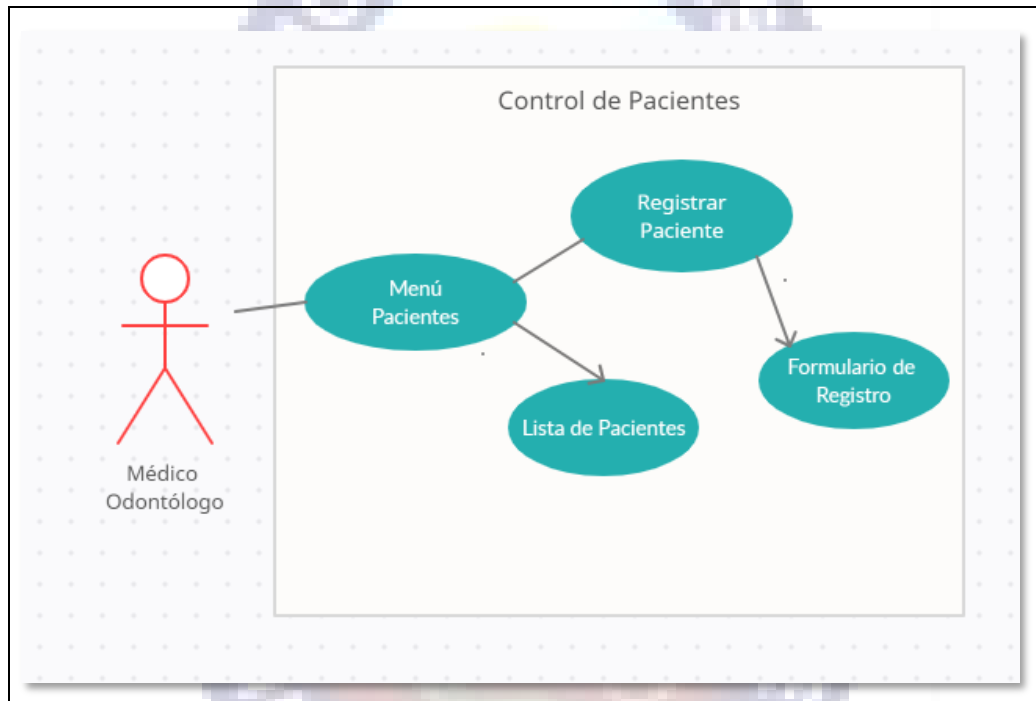


Figura 3.7. Caso de uso – Control de pacientes

En este sprint se abarcó los procesos que realiza el médico odontólogo para registrar los datos necesarios del paciente, es importante mencionar que el paciente es registrado por haber realizado una consulta o un primer tratamiento, cualquiera de estas dos acciones son tomadas como un tratamiento en el consultorio. También se desarrolló la tabla de información de pacientes y un componente de angular en el cual se desplegará la información de un paciente seleccionado

A continuación, se describe la especificación de cada caso de uso.

Nombre		Registro de pacientes
Actores		Médico odontólogo.
Precondición		El usuario debe estar autenticado.
Descripción		El caso de uso comienza cuando el médico odontólogo quiere realizar el registro de paciente.
Flujo principal	Evento actor	Evento sistema
	1.- El médico odontólogo quiere introducir los datos del paciente. 3.- El médico introduce los datos	2.- El sistema despliega el formulario de registro de la hoja que contendrá todos los datos del paciente. 4.- El sistema valida los datos introducidos 5.- El sistema despliega mensaje de confirmación
Escenarios alternativos		
Alternativa 1		Si en el escenario 4 el sistema encuentra datos que no concuerdan con los requisitos entonces el sistema despliega alertas.
Post condición		Los datos del registro deben ser almacenados en la base de datos.

Tabla 3.8. Especificación de casos de uso - Registro de Pacientes

Nombre		Información de pacientes
Actores		Médico odontólogo.
Precondición		El usuario debe estar autenticado.
Descripción		El caso de uso comienza cuando el médico odontólogo quiere ver la información de los pacientes.

Flujo principal	Evento actor	Evento sistema
	1.- El médico odontólogo quiere ver la información del paciente. 3.- El médico odontólogo selecciona un paciente de la tabla.	2.- El sistema despliega la tabla que contendrá información de todos los pacientes del consultorio. 4.- El sistema valida los datos seleccionados. 5.- El sistema despliega la información del paciente.
Escenarios alternativos		
Alternativa 1	Si en el escenario 2 el médico no encuentra la información de algún paciente deberá proceder a un registro desde el formulario.	
Post condición	Los datos seleccionados deben ser obtenidos de la base de datos.	

Tabla 3.9. Especificación de casos de uso - Lista de Pacientes

3.4.2.3 MODELO NAVEGACIONAL DEL SPRINT 2

El modelo navegacional muestra las opciones de navegación y procesos para el control de pacientes del consultorio dental.

A continuación, se puede observar el modelo navegacional del módulo de control de pacientes de la clínica dental.

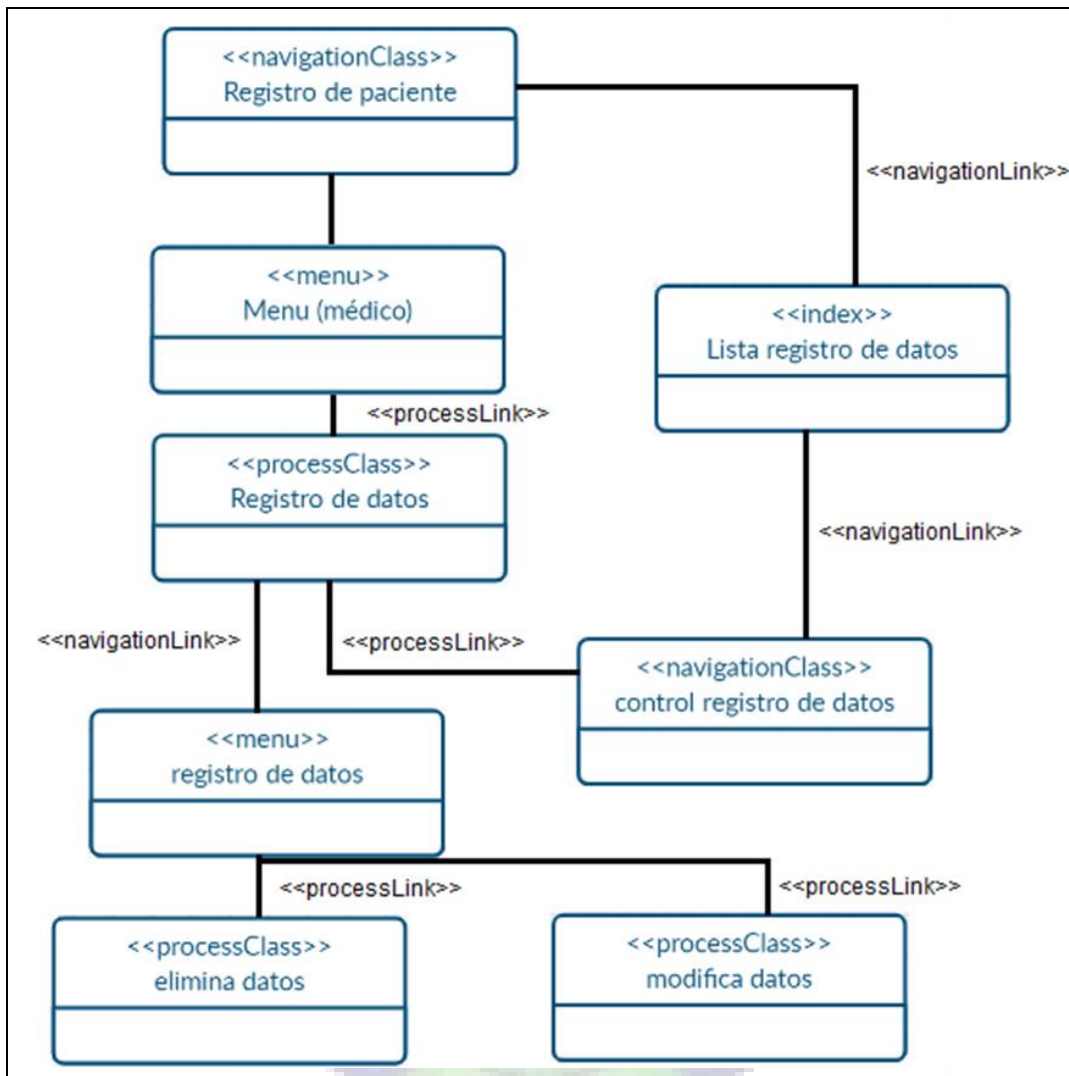


Figura 3.8. Modelo Navegacional – Control de Pacientes

3.4.2.4 PANTALLAS DEL SPRINT 2

Se muestra la lista de pacientes ya registrados en el consultorio dental con los datos más relevantes y algunas opciones para ver más detalles del paciente. Ver Figura 3.9.

CI	Nombre	Apellido	Teléfono	Fecha de Nacimiento	Edad	Sexo		
1434577	Stela	Nina Allaga	78963632	20 feb. 1998	24	Femenino		
123456	Sergio	Mamani Nina	77766655	2 feb. 1996	26	Masculino		
53323	Roger	Paxipati Aruni	72737322	2 nov. 1998	23	Masculino		
1234577	PruebaNomb	PruebaApel	77766655	24 feb. 2022	0	Femenino		
1231221	Jose	Aruni Santalla	78911095	17 mar. 1995	27	Masculino		
199821	Jorge	Blanco Calle	78916616	18 mar. 2022	0	Masculino		
1694577	Jazmin	Sunagua Gamboa	77766655	23 feb. 2001	21	Femenino		

Figura 3.9. Interfaz Lista de paciente

A continuación, se muestra el formulario de registro de pacientes con los datos que el consultorio médico requiere para la atención del paciente. Ver Figura 3.10

Crear Usuario
Complete los Campos

Nombres Apellidos

Fecha de Nacimiento: C.I.

Nro. Celular Teléfono de emergencia

Género
 Femenino
 Masculino

Foto de Perfil:
 Ninguno archivo selec.

Figura 3.10. Interfaz Registrar nuevo paciente

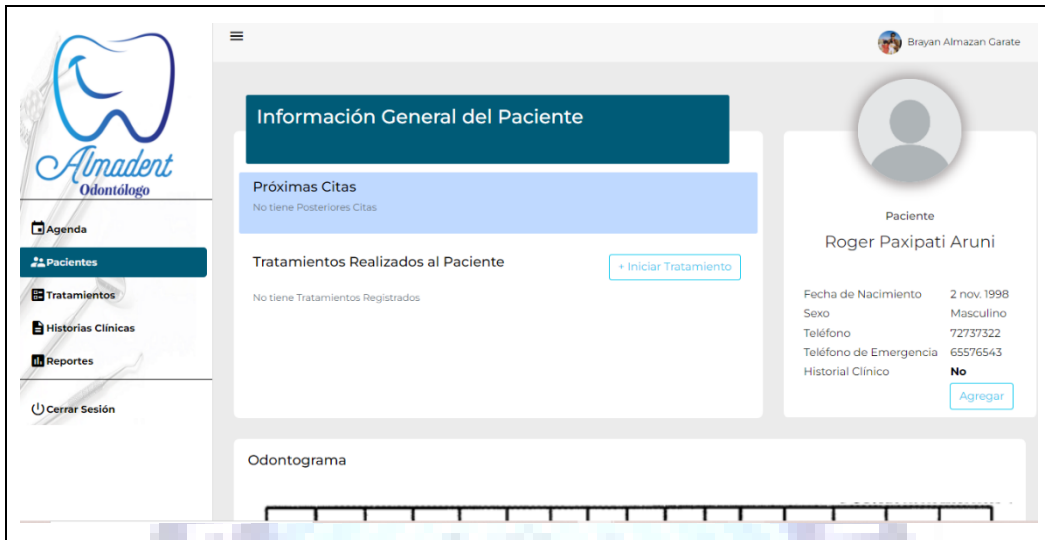


Figura 3.11. Información de un paciente seleccionado

3.4.2.5 PRUEBA UNITARIA DEL SPRINT 2

La prueba unitaria a continuación sirve para comprobar que el módulo funcione correctamente.

PRUEBA 1 - MÓDULO DE CONTROL DE PACIENTES	
Descripción	Al presionar registrar paciente nuevo, el médico ingresa al formulario de registro de paciente.
Objetivos	<ol style="list-style-type: none"> 1. Abrir la interfaz. 2. Introducir los datos requeridos del paciente. 3. Guardar correctamente los datos del paciente.
Condiciones	El médico odontólogo debe estar autenticado.
Resultado esperado	Que se registre satisfactoriamente los datos del paciente.
Resultados obtenidos	Se registró correctamente los datos del paciente.

Tabla 3.10. Prueba unitaria uno - Sprint 2

PRUEBA 2 - MÓDULO DE CONTROL DE PACIENTES	
Descripción	Al presionar en el carnet de un paciente, el médico ingresa al componente de información general del paciente.
Objetivos	<ol style="list-style-type: none"> 1. Abrir el componente. 2. Ver la información general del paciente.
Condiciones	El médico odontólogo debe estar autenticado.
Resultado esperado	Que se muestre satisfactoriamente la información del paciente.
Resultados obtenidos	Se desplegó correctamente la información del paciente.

Tabla 3.11. Prueba unitaria dos - Sprint 2

3.4.3 SPRINT 3: MÓDULO DE CONTROL DE TRATAMIENTOS Y PIEZAS DENTALES (ODONTOGRAMA)

En este módulo se registraron todas las piezas dentales con sus respectivas características, a toda esta estructura se la llama Odontograma, cada pieza dental tiene información de los tratamientos que se realizaron en el mismo. Por otro lado, se registraron los tratamientos que ofrece el consultorio dental según el médico odontólogo y su especialidad. Este sprint también nos ayudó a que cada paciente tenga un seguimiento a cada pieza dental y todos los tratamientos que tiene cada una de ellas.

3.4.3.1 PLANIFICACIÓN DEL SPRINT 3

Durante este sprint se establecieron los requerimientos que pertenecen al control de piezas dentales y tratamientos.

A continuación, se puede observar en una tabla la planificación para este sprint.

SPRINT 3: MÓDULO DE CONTROL DE TRATAMIENTOS Y PIEZAS DENTALES		
Sprint	Duración en horas	Días de trabajo
3	136	17 días hábiles
SPRINT BACKLOG		
Tareas	Tipo	Estado
Planificación y análisis de los requerimientos del sprint	Planificación	Terminado
Modelo conceptual	Diseño	Terminado
Modelo navegacional	Diseño	Terminado
Diseño de la interfaz gráfica para el control de tratamientos y piezas dentales.	Desarrollo	Terminado

Tabla 3.12. Tercer Sprint Backlog

3.4.3.2 ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 3

A continuación, se tiene el caso de uso del módulo de control de tratamientos y piezas dentales, con las especificaciones respectivas.

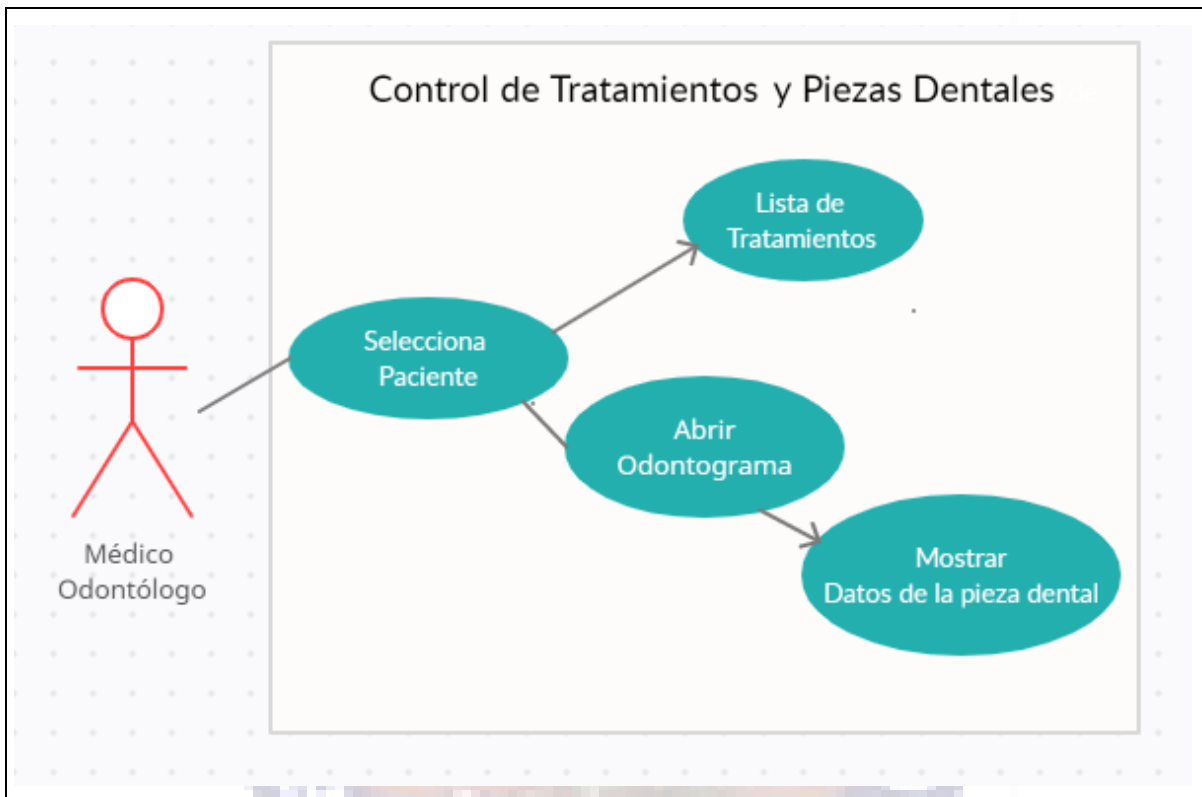


Figura 3.12. Caso de uso – Control de tratamientos y piezas dentales

En este sprint se registró toda la estructura del odontograma digital, con esta estructura se hará un control de los tratamientos que tiene la pieza dental de cada paciente que este registrado en el consultorio dental, después de cada tratamiento se actualiza el estado de la pieza dental en el odontograma y en la historia clínica.

A continuación, se describe la especificación del caso de uso.

Nombre		Control de tratamientos y piezas dentales
Actores		Médico odontólogo
Precondición		El usuario debe estar autenticado
Descripción		El caso de uso comienza cuando el médico odontólogo quiere verificar los tratamientos de la pieza dental.
Flujo principal	Evento actor	Evento sistema
	1.- El médico odontólogo quiere verificar el estado de la pieza dental. 3.- El médico introduce los datos nuevos después de un tratamiento	2.- El sistema despliega el odontograma. 4.- El sistema valida los datos introducidos 5.- El sistema despliega mensaje de confirmación
Escenarios alternativos		
Alternativa 1		Si en el escenario 4 el sistema encuentra datos que no concuerdan con los requisitos entonces el sistema despliega alertas
Post condición		Los datos del registro deben ser almacenados en la base de datos.

Tabla 3.13. Especificación de casos de uso - Control de tratamientos y piezas dentales

3.4.3.3 MODELO NAVEGACIONAL DEL SPRINT 3

A continuación, se puede observar el modelo navegacional del módulo de control de piezas dentales.

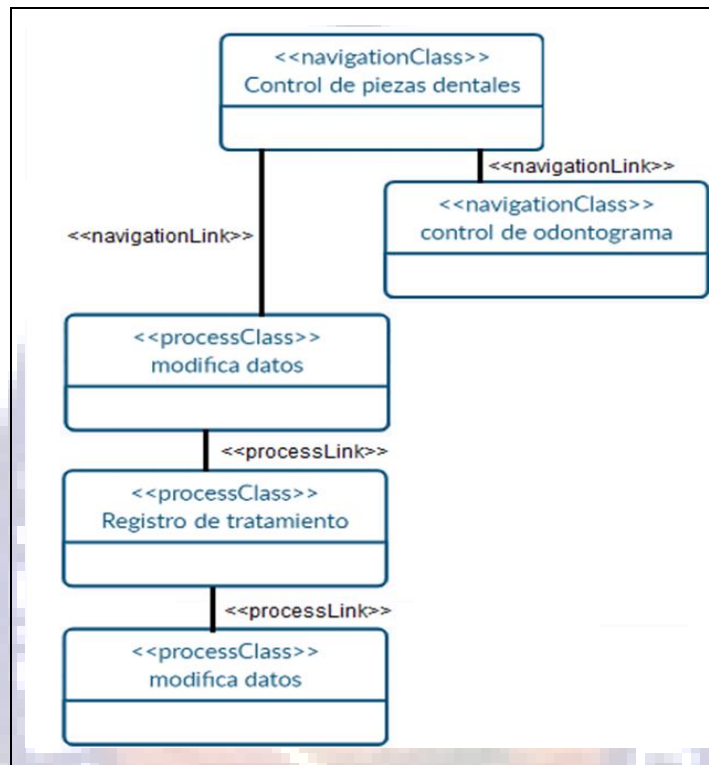


Figura 3.13. Modelo navegacional – Control de tratamientos y piezas dentales

3.4.3.4 PANTALLAS DEL SPRINT 3

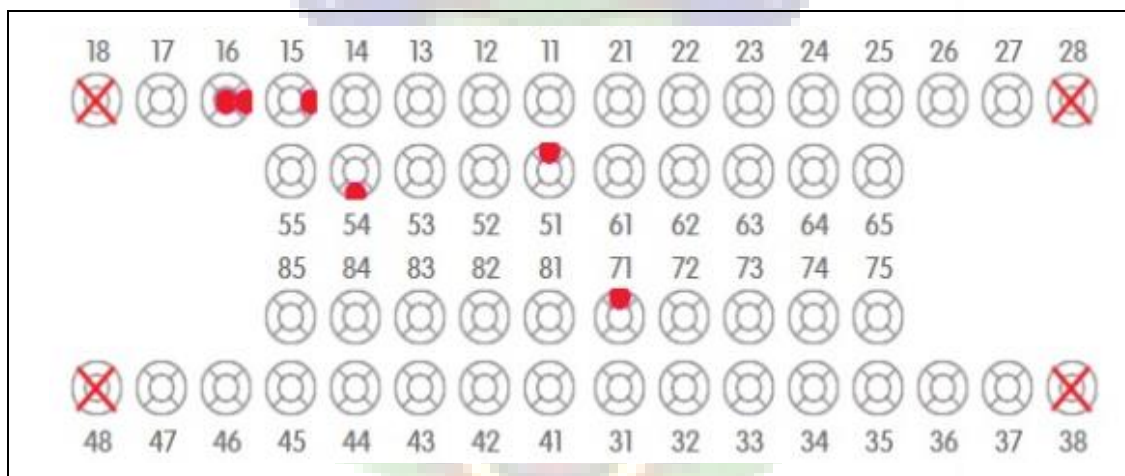


Figura 3.14. Modelo de Odontograma

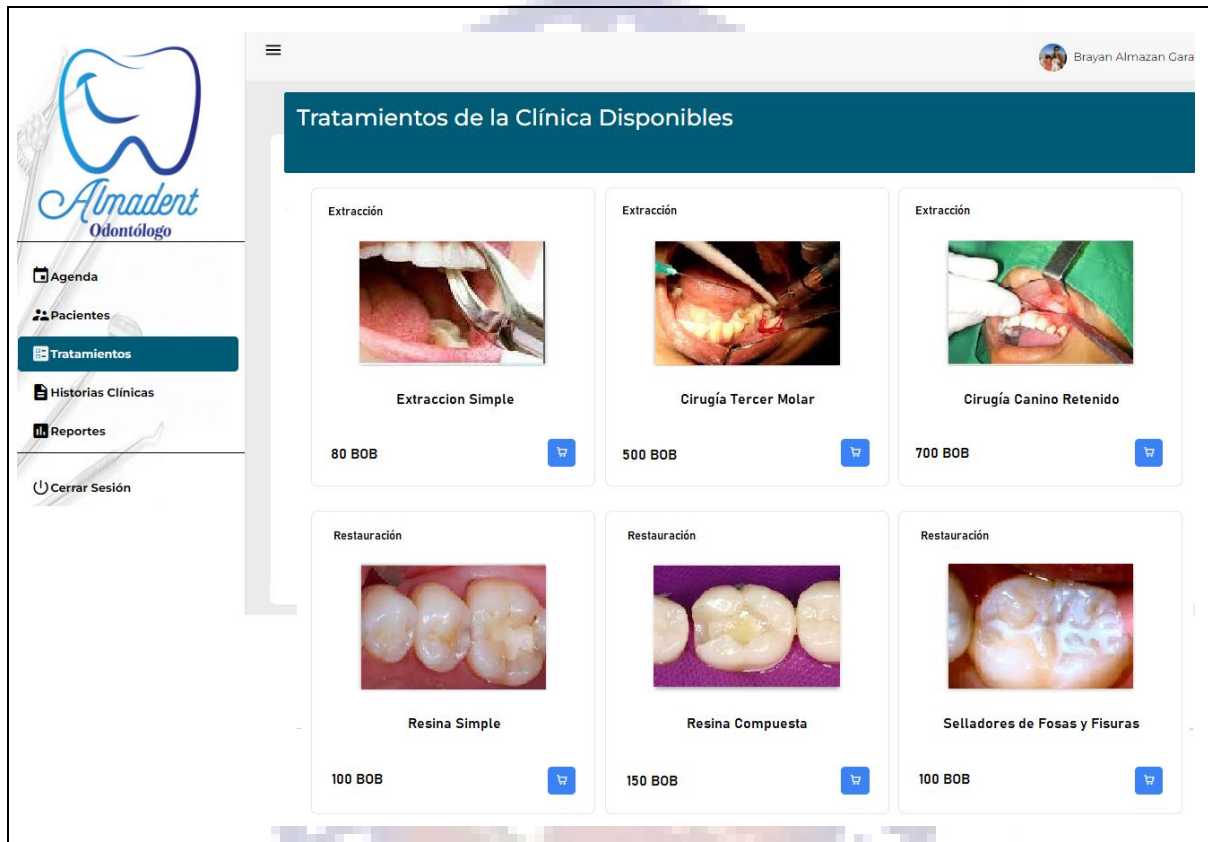


Figura 3.15. Interfaz de tratamientos del consultorio dental

3.4.3.5 PRUEBA UNITARIA DEL SPRINT 3

La prueba unitaria a continuación sirve para comprobar que el módulo funcione correctamente.

PRUEBA 1 - MÓDULO DE CONTROL DE TRATAMIENTOS Y PIEZAS DENTALES	
Descripción	Al ingresar al odontograma el médico visualiza todas las piezas dentales.
Objetivos	Abrir la interfaz Visualizar las piezas dentales Seleccionar una pieza dental Ver los tratamientos de la pieza dental
Condiciones	El médico odontólogo debe estar autenticado.
Resultado esperado	Que se muestre la información completa de la pieza dental.
Resultados obtenidos	Se mostró correctamente la información de la pieza dental

Tabla 3.14. Prueba unitaria - Sprint 3

3.4.4 SPRINT 4: MÓDULO DE MANEJO DE HISTORIALES CLÍNICOS

El trabajo de este sprint avanzó a buen ritmo ajustándose de un modo más aproximado a la planificación planteada, hecho que acrecentó la motivación.

Este es uno de los sprints más importantes ya que la información de los historiales clínicos es solicitada por gran parte de los módulos del sistema.

3.4.4.1 PLANIFICACIÓN DEL SPRINT 4

SPRINT 4: MÓDULO DE MANEJO DE HISTORIALES CLÍNICOS		
Sprint	Duración en horas	Días de trabajo
4	120	15 días hábiles
SPRINT BACKLOG		
Tareas	Tipo	Estado
Planificación y análisis de los requerimientos del sprint	Planificación	Terminado
Modelo conceptual	Diseño	Terminado
Modelo navegacional	Modelado	Terminado
Diseño de la interfaz gráfica para la redacción de historias clínicas.	Desarrollo	Terminado

Tabla 3.15. Cuarto Sprint Backlog

3.4.4.2 ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 4

A continuación, se tiene el caso de uso del módulo de control de historias clínicas, cuarto sprint, con las especificaciones respectivas.

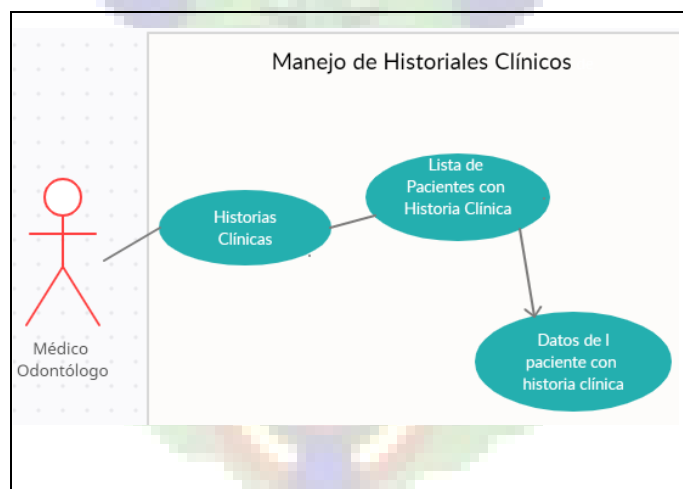


Figura 3.16. Caso de uso – Manejo de historias clínicas

En este sprint se abarca los procesos que realiza la clínica dental para redactar la historia clínica de cada paciente, para cumplir con estos requerimientos, se tiene el siguiente caso de uso general.

A continuación, se describe la especificación de cada caso de uso.

Nombre		Control de historia clínica
Actores		Médico odontólogo
Precondición		El usuario debe estar autenticado
Descripción		El caso de uso comienza cuando el médico odontólogo quiere redactar el historial clínico del paciente.
Flujo principal	Evento actor	Evento sistema
	1.- El médico odontólogo quiere redactar el historial clínico del paciente. 3.- El médico redacta el historial clínico.	2.- El sistema despliega el formulario de redacción de la hoja que contendrá el historial clínico del paciente. 4.- El sistema valida los datos introducidos. 5.- El sistema despliega mensaje de confirmación.
Escenarios alternativos		
Alternativa 1	Si en el escenario 4 el sistema encuentra datos que no concuerdan con los requisitos entonces el sistema despliega alertas	
Post condición	Los datos del registro deben ser almacenados en la base de datos.	

Tabla 3.16. Especificación casos de uso - Control de tratamientos y piezas dentales

3.4.4.3 MODELO NAVEGACIONAL DEL SPRINT 4

A continuación, se puede observar el modelo navegacional del módulo de control de historias clínicas.



Figura 3.17. Modelo navegacional – Manejo de Historias Clínicas

3.4.4.4 PANTALLAS DEL SPRINT 4

HISTORIA CLÍNICA	
EMISIÓN : 15-12-2021	
Nombre y Apellidos: Asbel Añamuro Miranda	Edad: 8 años
Fecha de Nacimiento: 25-03-2013	
Domicilio: 7 de octubre #3515 ZonaVilla Yunguyo	Teléfono: 0
DATOS FAMILIARES	
Nombre Padre: Hector Añamuro Colquehuanca	Celular: 0
Nombre Madre: Celia Vivian Miranda Hulliri	Celular: 0
Nombre Tutor:	Celular: 0
ANTECEDENTES PERSONALES	
¿Fue sometido a intervenciones quirúrgicas?	NO
¿Está actualmente en tratamiento médico?	SÍ
¿Toma algún medicamento actualmente?	SÍ
¿Tuvo reacción alérgica a la anestesia local?	SÍ
¿Posterior a heridas o extracciones tuvo hemorragias?	SÍ
¿Los cortes y heridas tardan en cicatrizar?	SÍ
¿Presentó alergia a algún medicamento?	SÍ
¿Sufre desmayos?	SÍ
ENFERMEDADES SISTÉMICAS	

Figura 3.18. Modelo de historia clínica

antefa	
HÁBITOS O COSTUMBRES	
Respiración bucal	NO
Usa biberón	NO
Usa chupón	NO
Succión de dedo	NO
Onicofagia	NO
Mastica algún objeto	SÍ
OTROS	
otro	
FECHA ÚLTIMA VISITA AL ODONTÓLOGO	
fech	
TIPO DE TRATAMIENTO RECIBIDO	
tipo	
OBSERVACIONES	
obser	
PACIENTE COLABORADOR	NO
EXAMEN DE TEJIDOS BLANDOS	

Figura 3.19. Modelo de historia clínica 2

22	j	62	j	42	j	82	j
23	k	63	k	43	k	83	k
24	l	64	l	44	l	84	l
25	m	65	m	45	m	85	m
26	n			46	n		
27	o			47	o		
28	p			48	p		

HIGIENE ORAL	MALA
Técnica de cepillado: asdasdas	
Frecuencia: asddawww	
Usa pasta dental	SÍ
Usa hilo dental	NO
Usa enjuague bucal	SÍ

RECOMENDACIONES

reco

Figura 3.20. Modelo de historia clínica 3

3.4.4.5 PRUEBA UNITARIA DEL SPRINT 4

La prueba unitaria a continuación sirve para comprobar que el módulo funcione correctamente.

PRUEBA 1 - MÓDULO DE CONTROL DE HISTORIA CLÍNICA	
Descripción	Al presionar historias clínicas, el médico odontólogo ingresa a la interfaz de pacientes con historia clínica.
Objetivos	<ol style="list-style-type: none"> 1. Que ingrese a la interfaz de historia clínica 2. Abrir la interfaz 3. Seleccionar un paciente 4. Desplegar el historial del paciente
Condiciones	El médico odontólogo debe estar autenticado.

Resultado esperado	Que se muestre satisfactoriamente la historia clínica de los pacientes.
Resultados obtenidos	La interfaz muestra los datos correctamente.

Tabla 3.17. Prueba unitaria - Sprint 4

3.4.5 SPRINT 5: MÓDULO DE CONTROL COSTOS DE TRATAMIENTOS Y PAGOS

En este módulo gran parte del sistema está desarrollado, este sprint fue realizado como los sprints anteriores y el avance fue óptimo.

El desarrollo siguió el orden lógico de las tareas establecidas en la planificación, comenzando a familiarizarse con el proyecto.

3.4.5.1 PLANIFICACIÓN DEL SPRINT 5

Durante este sprint se establecieron los requerimientos que pertenecen al módulo de control de pagos y costos de tratamientos, tomando en cuenta que los pagos se realizan dependiendo del tratamiento y pueden realizar los pagos en una solo cuota o en varias para tratamientos de múltiples sesiones.

A continuación, se puede observar en una tabla la planificación para este sprint.

SPRINT 5: MÓDULO DE CONTROL COSTOS DE TRATAMIENTOS Y PAGOS		
Sprint	Duración en horas	Días de trabajo
5	112	14 días hábiles
SPRINT BACKLOG		
Tareas	Tipo	Estado
Planificación y análisis de los requerimientos del sprint	Planificación	Terminado
Diseño de la base de datos para el sistema web	Desarrollo	Terminado
Modelo conceptual	Diseño	Terminado
Modelo navegacional	Modelado	Terminado
Diseño de la interfaz gráfica del control de costos de tratamientos.	Desarrollo	Terminado

Tabla 3.18. Quinto Sprint Backlog

3.4.5.2 ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 5

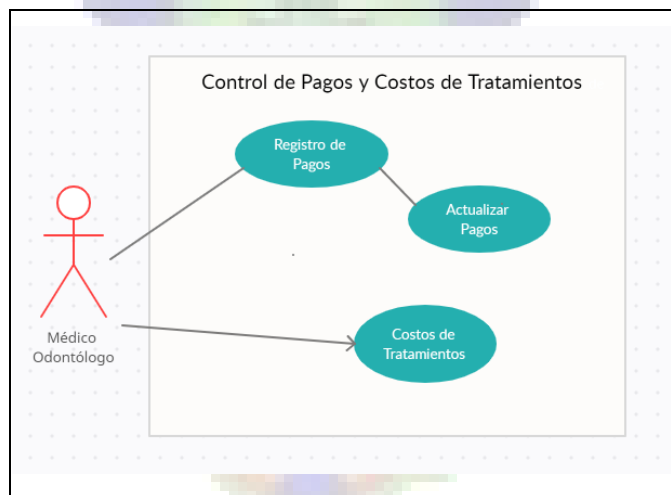


Figura 3.21 Caso de uso - Control de pagos

A continuación, se muestra la tabla de caso de uso de registro de pago inicial.

Nombre		Registro de pago inicial
Actores		Médico odontólogo
Precondición		El usuario debe estar autenticado.
Descripción		El caso de uso comienza cuando el médico odontólogo quiere registrar el pago inicial del tratamiento.
Flujo principal	Evento actor	Evento sistema
	1.- El médico odontólogo selecciona la opción de registrar pago. 3.- El médico odontólogo introduce los datos del pago.	2.-El sistema despliega el formulario de registro de pagos. 4.- El sistema valida los datos introducidos 5.- El sistema despliega mensaje de confirmación.
Escenarios alternativos		
Alternativa 1		Si en el escenario 4 el sistema encuentra datos que no concuerdan con los requisitos entonces el sistema despliega alertas.
Post condición		Los datos del registro deben ser almacenados en la base de datos.

Tabla 3.19. Especificación de casos de uso - Registrar pago inicial

Se puede observar a continuación el caso de uso de costos de tratamientos.

Nombre		Costos de tratamientos
Actores		Médico odontólogo
Precondición		El usuario debe estar autenticado
Descripción		El caso de uso comienza cuando el médico odontólogo quiere ver los costos de tratamientos del consultorio.
Flujo principal	Evento actor	Evento sistema
	1.- El médico odontólogo selecciona la opción de tratamientos la opción de registrar pago.	2.- El sistema despliega la interfaz de costos de tratamientos.
	3.- El médico odontólogo introduce los datos del pago que se actualiza en el monto total del tratamiento.	4.- El sistema valida los datos introducidos 5.- El sistema despliega mensaje de confirmación
Escenarios alternativos		
Alternativa 1	Si en el escenario 4 el sistema encuentra datos que no concuerdan con los requisitos entonces el sistema despliega alertas	
Post condición	Los datos del registro deben ser almacenados en la base de datos.	

Tabla 3.20. Especificación de casos de uso - Costos de tratamientos

3.4.5.3 MODELO NAVEGACIONAL DEL SPRINT 5

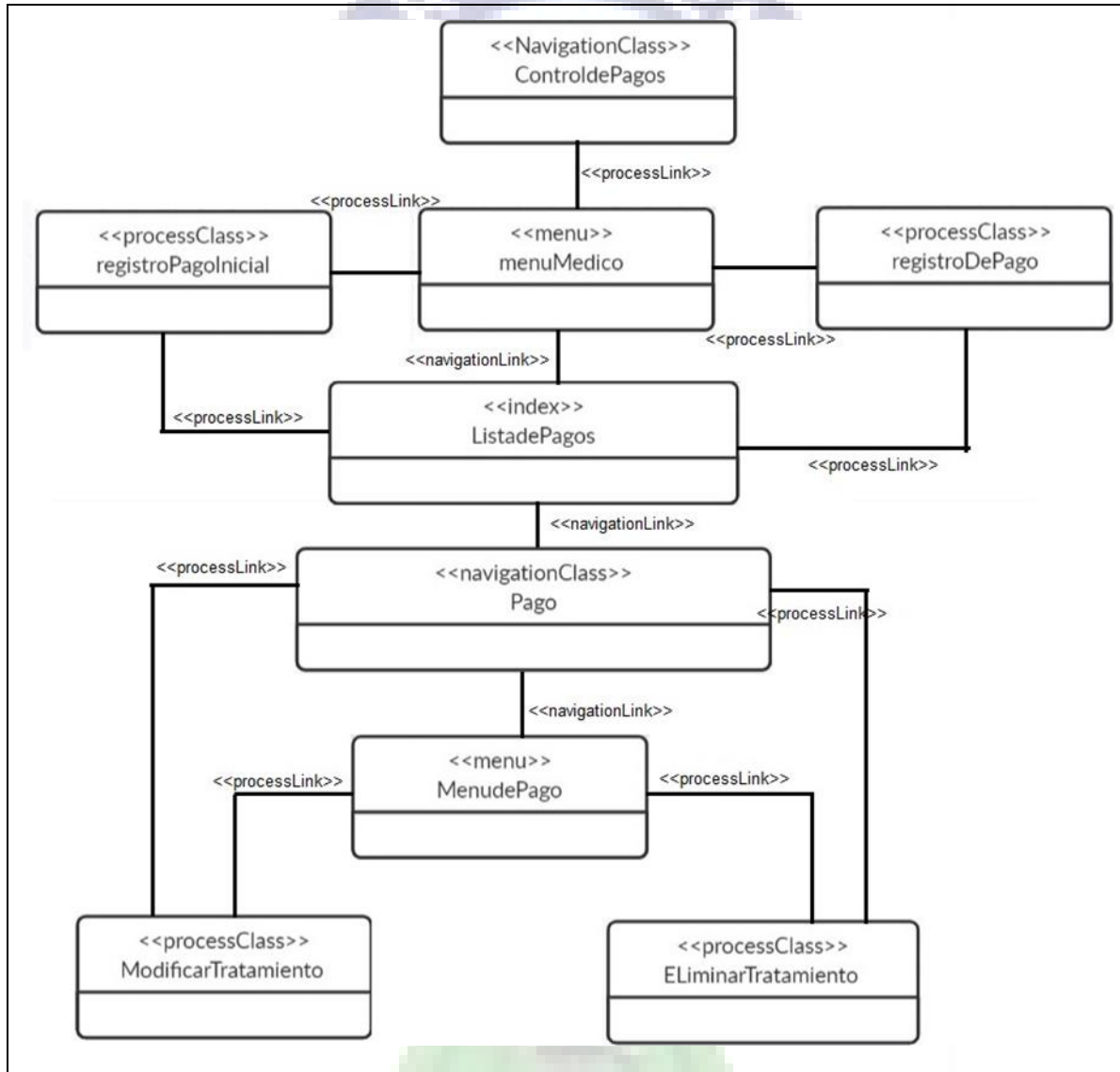


Figura 3.22 Modelo navegacional – Control de pagos

3.4.5.4 PANTALLAS DEL SPRINT 5



Figura 3.23. Pantalla de costos de tratamientos

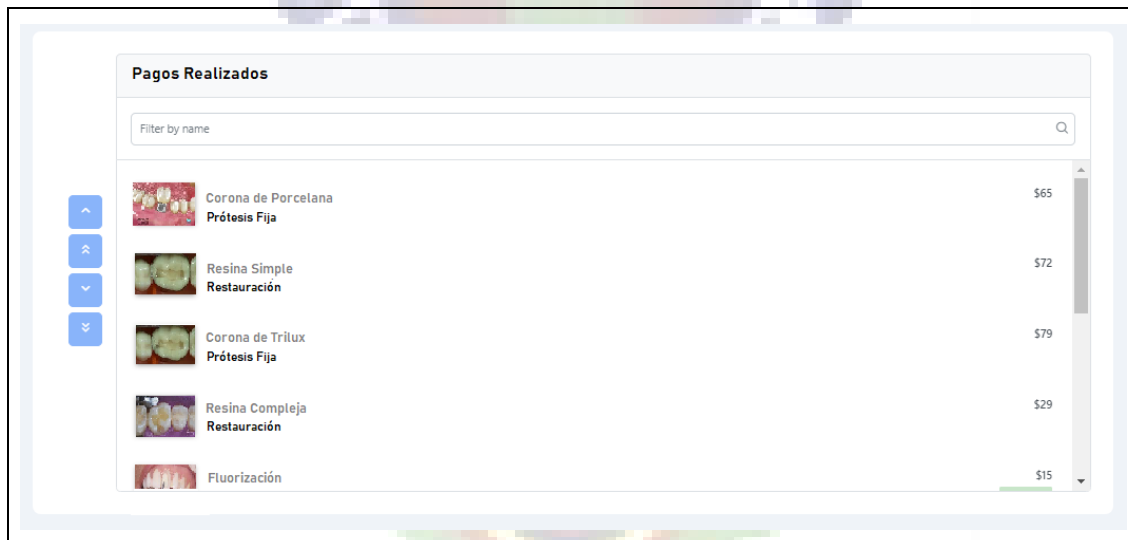


Figura 3.24. Pantalla de pagos realizados

3.4.5.5 PRUEBA UNITARIA DEL SPRINT 5

La prueba unitaria a continuación sirve para comprobar que el módulo funcione correctamente.

PRUEBA 1 - MÓDULO DE CONTROL DE COSTOS DE TRATAMIENTOS Y PAGOS	
Descripción	Al presionar pagos, el médico ingresa a la interfaz de pagos
Objetivos	<ol style="list-style-type: none">1. Que ingrese a la interfaz de pagos2. Abrir los pagos realizados por pacientes de acuerdo al tratamiento3. Registrar los pagos que hace el paciente.4. Guardar correctamente los pagos realizados
Condiciones	El médico odontólogo debe estar autenticado.
Resultado esperado	Que se abra satisfactoriamente la interfaz de pagos
Resultados obtenidos	La interfaz de pagos guarda los datos correctamente.

Tabla 3.21. Prueba unitaria - Sprint 5

3.4.6 SPRINT 6: MÓDULO DE CONTROL DE CITAS MÉDICAS

Para este sprint gran parte del sistema está completo y funcional por lo cual los resultados obtenidos durante el desarrollo fueron muy buenos y se tuvo un avance muy fluido.

La planificación propuesta para el sistema fue cumplida paso a paso, aunque el trabajo resultó menos complejo, no se logró cumplir la planificación en su totalidad y quedó un pequeño porcentaje de este trabajo pendiente en el siguiente sprint.

3.4.6.1 PLANIFICACIÓN DEL SPRINT 6

A continuación, se tiene la tabla de tareas planificadas para este sprint.

SPRINT 6: MÓDULO DE CONTROL DE CITAS MÉDICAS		
Sprint	Duración en horas	Días de trabajo
6	168	21 días hábiles
SPRINT BACKLOG		
Tareas	Tipo	Estado
Planificación y análisis de los requerimientos del sprint	Planificación	Terminado
Modelo conceptual	Diseño	Terminado
Modelo navegacional	Modelado	Terminado
Diseño de la interfaz gráfica para la reserva de citas.	Desarrollo	Terminado
Diseño de la interfaz gráfica de agenda de citas	Desarrollo	Terminado

Tabla 3.22. Sexto Sprint Backlog

3.4.6.2 ESPECIFICACIÓN DE CASO DE USO DEL SPRINT 6

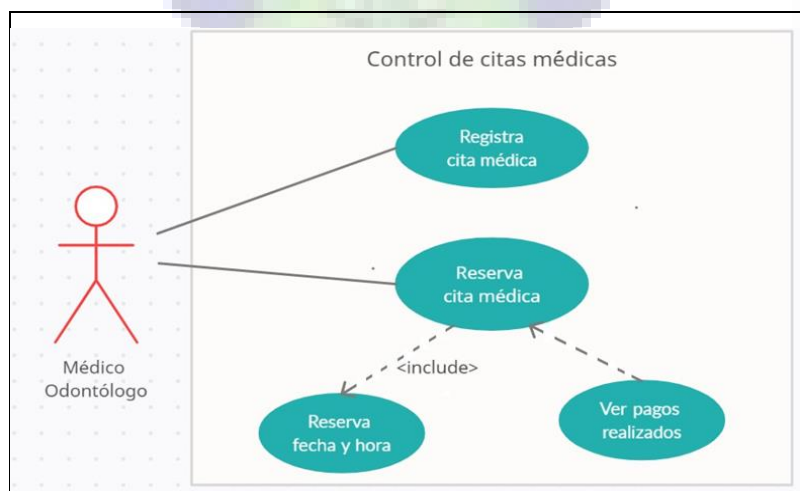


Figura 3.25 Caso de uso - Control de citas médicas

Nombre		Registro de cita médica
Actores		Médico odontólogo
Precondición		El usuario debe estar autenticado
Descripción		El caso de uso comienza cuando el médico odontólogo quiere registrar una nueva cita médica
Flujo principal	Evento actor	Evento sistema
	1.- El médico odontólogo selecciona la opción agenda 3.- El médico odontólogo selecciona el día y la hora.	2.- El sistema despliega una agenda con los horarios ocupados y disponibles. 4.- El sistema despliega un formulario pequeño que requiere el nombre y carnet del paciente
Escenarios alternativos		
Alternativa 1	Si en el escenario 4 el sistema no encuentra el número de carnet del paciente el sistema despliega alertas.	
Post condición	Los datos de la cita deben ser almacenados en la base de datos.	

Tabla 3.23. Especificación de caso de uso - Registrar cita médica

Nombre		Eliminar cita
Actores		Médico odontólogo
Precondición		El usuario debe estar autenticado
Descripción		El caso de uso comienza cuando el médico odontólogo quiere cancelar una cita médica

Flujo principal	Evento actor	Evento sistema
	1.- El médico odontólogo seleccionan la cita que desea cancelar. 3.- El médico odontólogo selecciona la opción de cancelar.	2.- El sistema despliega las opciones de la cita. 4.- El sistema da elimina la cita.

Tabla 3.24. Especificación de casos de uso - Eliminar cita médica

3.4.6.3 MODELO NAVEGACIONAL DEL SPRINT 6

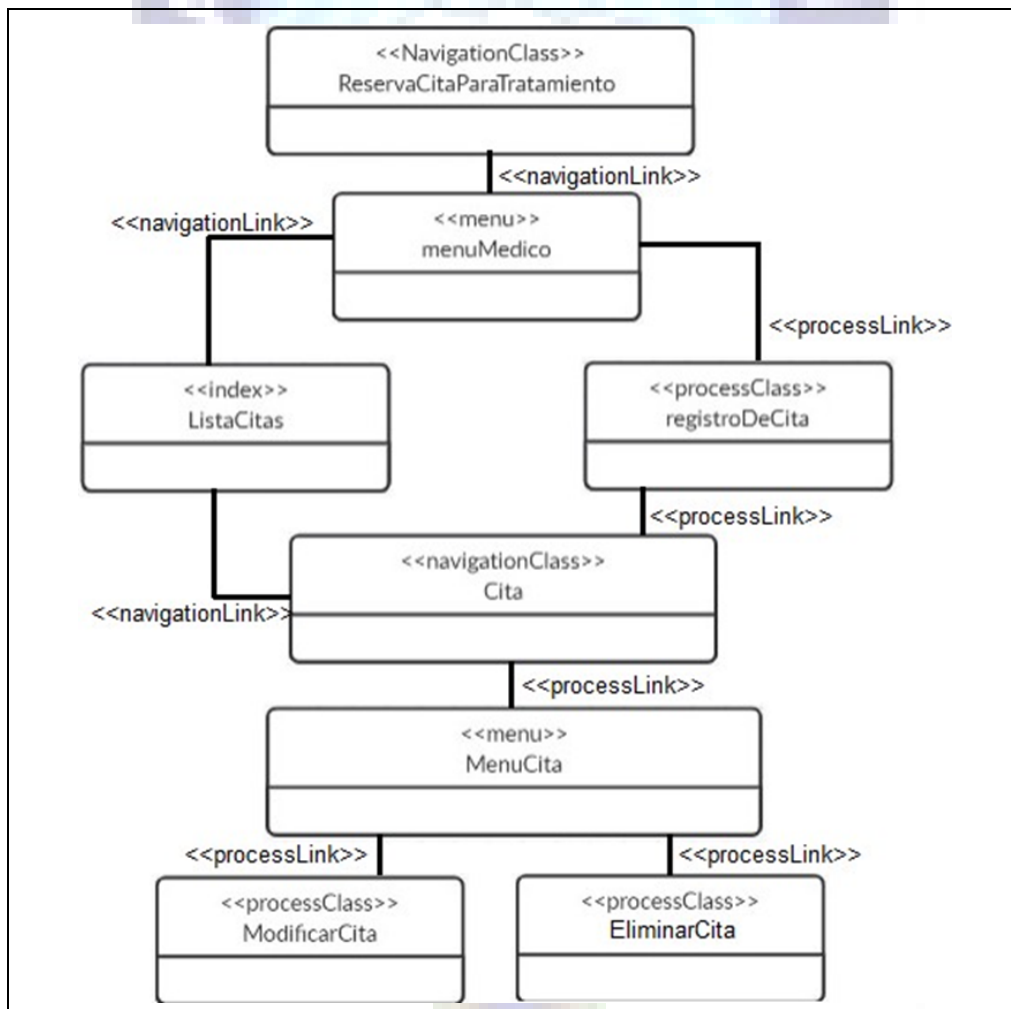


Figura 3.26. Modelo navegacional – Control de citas médicas

3.4.6.4 PANTALLAS DEL SPRINT 6

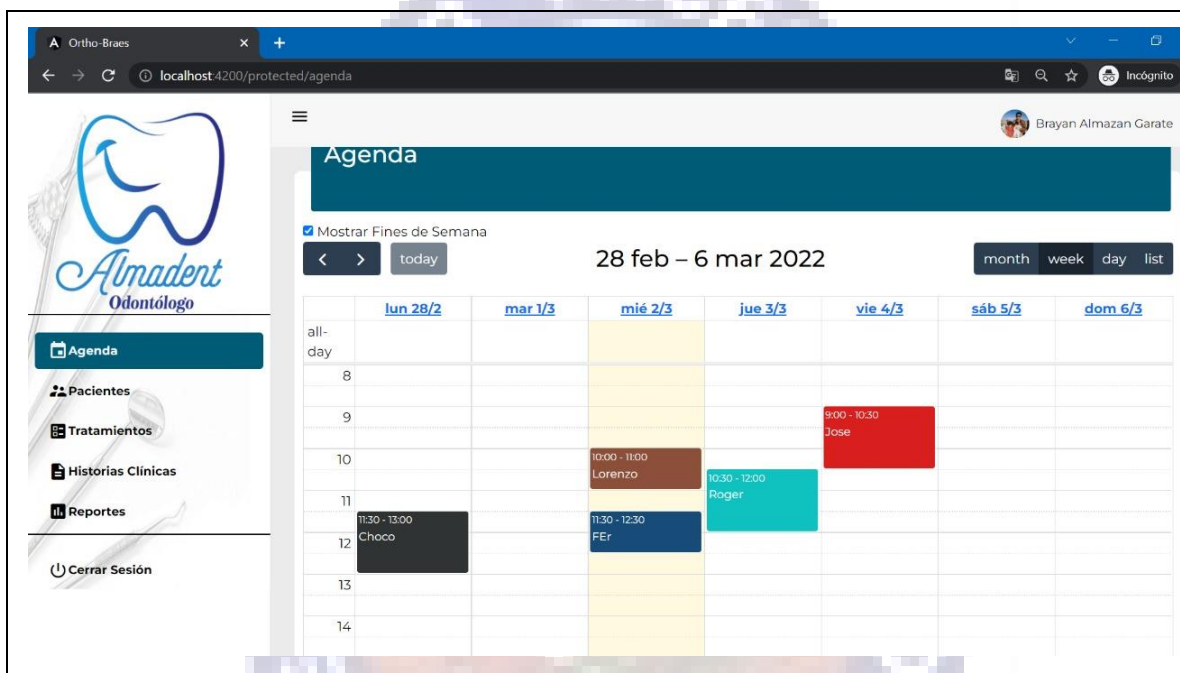


Figura 3.27. Modelo de agenda

3.4.6.5 PRUEBA UNITARIA DEL SPRINT 6

PRUEBA 1 - MÓDULO DE CONTROL DE CITAS MÉDICAS	
Descripción	Al presionar agenda, el médico ingresa a la interfaz de citas
Objetivos	<ol style="list-style-type: none"> 1. Que ingrese a la interfaz citas 2. Abrir la interfaz 3. Reservar cita médica para el paciente 4. Guardar los registros
Condiciones	El médico odontólogo debe estar autenticado.
Resultado esperado	Que se abra satisfactoriamente la interfaz de citas
Resultados obtenidos	El sistema registre correctamente las citas reservadas hechas por el médico.

Tabla 3.25. Prueba unitaria - Módulo de control de citas

3.4.7 DIAGRAMA DE CLASES.

Este diagrama es de estructura estática y muestra las clases del sistema y sus interrelaciones.

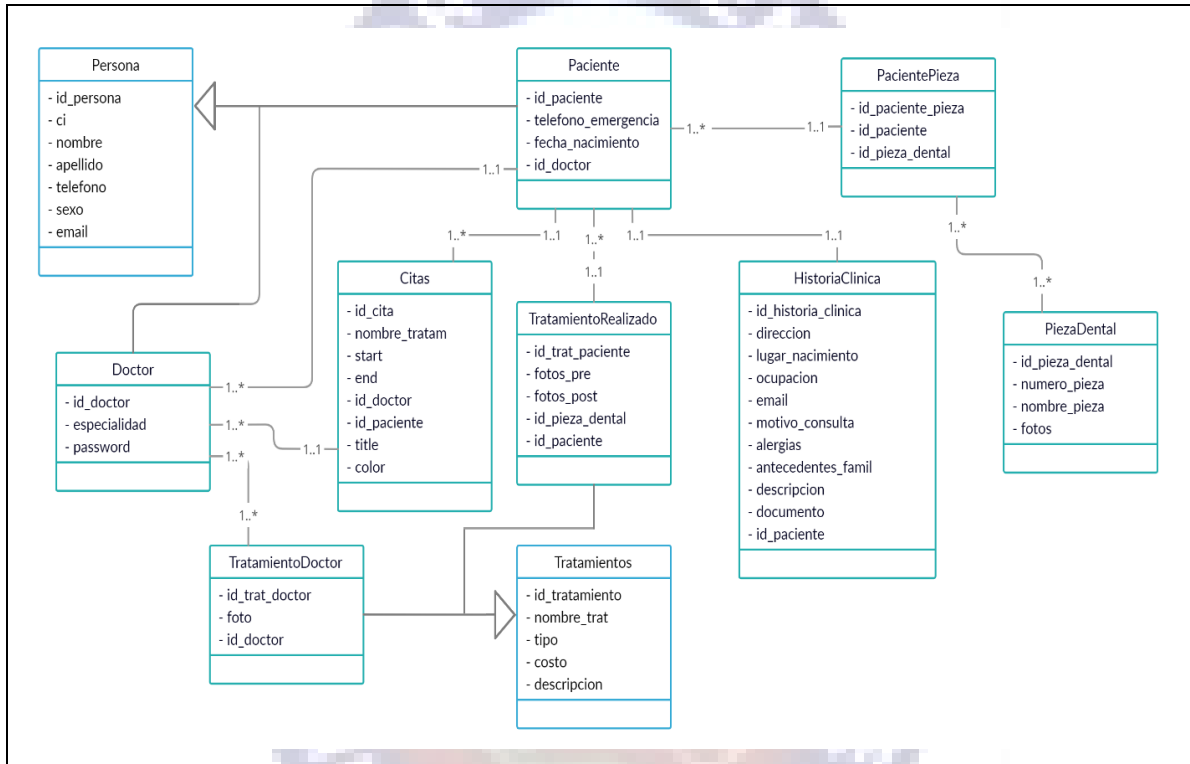


Figura 3.28. Diagrama de clases

3.5 MODELO ENTIDAD RELACIÓN

Partiendo del diagrama de clases, todas las clases pasan a ser entidades, los atributos de las clases pasan a ser atributos de la entidad, se crean nuevos atributos identificadores para cada entidad, en este modelo no hay operaciones.

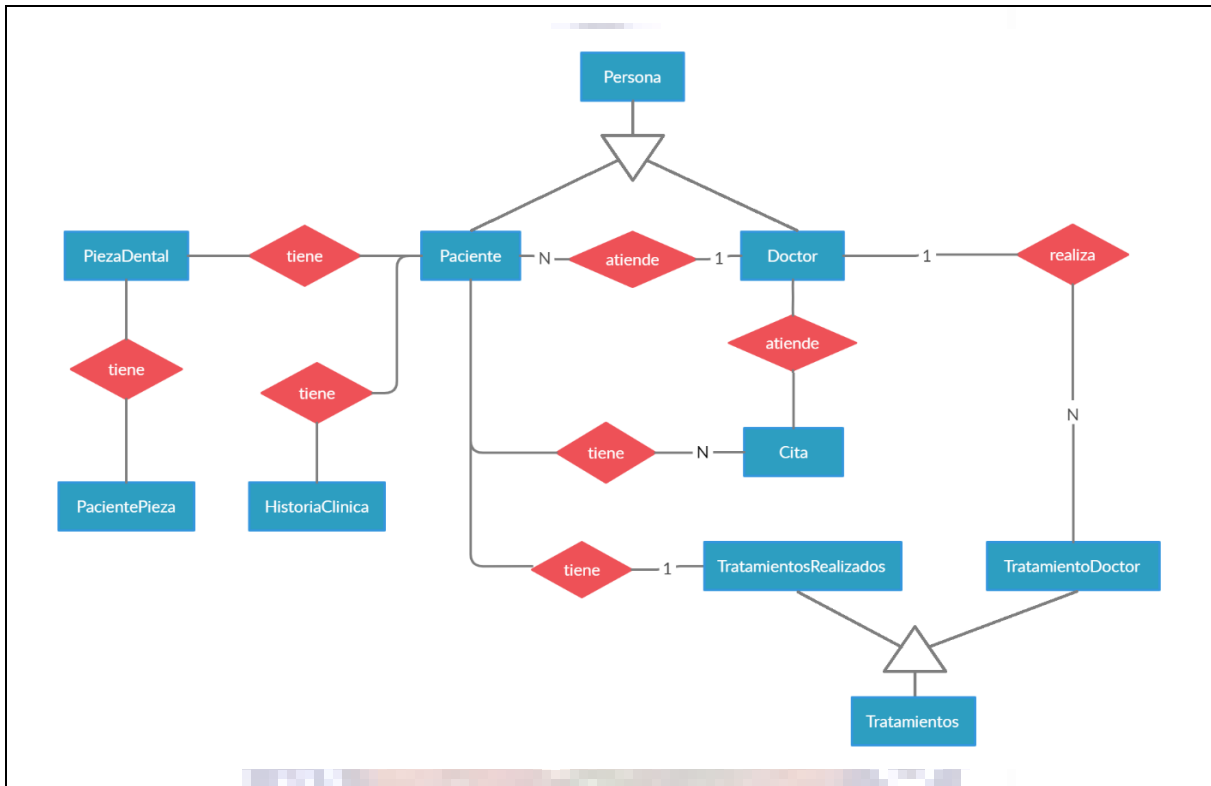


Figura 3.29. Modelo entidad-relación

3.6 FASE DEL POST – GAME

En esta fase generalmente se realizan pruebas unitarias para cada iteración y las pruebas de stress. Sin embargo, para el proyecto, como se vio anteriormente, se realizó una sección de pruebas unitarias que son detalladas en el desarrollo y redacción de cada sprint.

3.6.1 PRUEBAS DE STRESS

Esta prueba se realiza para determinar la solidez del sistema en momentos de carga extrema y ayuda a los administradores a determinar si el sistema rendirá lo suficiente en caso de sobrecarga.

Sin embargo, el consultorio dental solo se contará con máximo 5 médicos odontólogos para el manejo del sistema lo cual un tiempo de respuesta corto y el margen de error que existe es 0, lo que implica que rinde de manera óptima.

CAPITULO IV

4. CALIDAD Y SEGURIDAD

El presente capítulo se realizará y se comprobará la calidad al igual que la seguridad del software haciendo uso de uno de los estándares para evaluar el proyecto. Para medir la calidad de software se utilizó la norma 9126, cuyos aspectos a desarrollar son: funcionalidad, confiabilidad, usabilidad, mantenibilidad y portabilidad.

4.1 FACTORES DE SEGURIDAD

4.1.1 SEGURIDAD DE BASE DE DATOS

La base de datos que se utilizó fue MySQL. En cuanto a la forma de resguardo se realizó el siguiente punto:

- El sistema tiene registrado el nombre de usuario y contraseña de acceso, cada usuario tiene información distinta de sus pacientes.

La información del consultorio es importante, por lo tanto, su resguardo es fundamental, la conexión a la base de datos y el cierre de la conexión es de forma automática.

4.1.2 COPIAS DE SEGURIDAD

Para la realización de copias de seguridad se aplicó la utilidad cliente MysqlDump que realiza copias de seguridad lógicas, produciendo un conjunto de sentencias SQL que pueden ser ejecutadas para reproducir las definiciones de objetos de la base de datos original y los datos de las tablas. El comando MysqlDump también puede generar la salida en formato CSV, otro texto delimitado o XML

4.1.3 SEGURIDAD DE AUTENTIFICACIÓN DE USUARIO

4.1.3.1 BCRIPT

Para nuestro sistema se aplicó un componente de autenticación de usuario, en este componente se tomó en cuenta la forma en la que la contraseña es almacenada en la base de datos. En ese sentido la contraseña fue encriptada con una biblioteca llamada “Bcrypt” la cual se encarga de hacer un “hasheo” de la contraseña antes de almacenarla en la base de datos, a diferencia de una encriptación que utiliza una especie de llave o firma, esta biblioteca nos permite guardar una encriptación que no puede ser descifrada de ninguna manera. Para nuestro sistema aplicamos 10 rondas de encriptación, lo cual es suficiente y apropiado para la cantidad de memoria que utilizará nuestro sistema.

4.1.4 SEGURIDAD DEL SISTEMA

4.1.4.1 JSON WEB TOKEN

Después de la autenticación del usuario el sistema genera una encriptación de jsonwebtoken que es almacenada en el local storage del navegador con el propósito de mantener la sesión activa mientras el médico odontólogo utilice el sistema.

Cuando el médico desea salir del sistema elige la opción de “cerrar la sesión”, esto hace que el navegador borre todos los registros almacenados de la sesión y el sistema nos redirige a la pantalla de inicio de sesión.

4.1.4.2 ANGULAR GUARDS

Para el sistema se aplicó un servicio propio de angular llamado Angular Guards del cual se implementaron específicamente dos interfaces llamadas: CanActivate y CanLoad.

CanActivate: Decide si se puede activar una ruta, esta protección puede no ser la mejor manera para los módulos de características que se cargan de forma diferida, ya que esta protección siempre cargará el módulo en la memoria, incluso si la protección devolvió falso, lo que significa que el usuario no está autorizado para acceder la ruta.

CanLoad: Decide si un módulo se puede cargar de forma perezosa, controla si una ruta se puede cargar. Esto se vuelve útil para los módulos de características que se cargan de forma diferida. Ni siquiera cargarán si el guardia devuelve falso.

4.2 CALIDAD DE SOFTWARE

A continuación, se detallarán los aspectos de la norma ISO-9126 que se tomaron en cuenta para nuestro sistema.

4.2.1 TÉCNICA ISO 9126

4.2.1.1 FUNCIONALIDAD

La métrica que se utiliza para medir la habilidad de filtrar las actividades de la garantía de calidad y control es EED, que significa eficiencia de defectos.

Para el cálculo de punto función se utilizará la siguiente fórmula de acuerdo a la International Function Point Users Group (IFPUG)/ISO 9126:

$$PF = \text{Cuenta Total} * (X + \text{Min}(Y) * \sum Fi) \quad (1)$$

Donde:

PF: Medida de funcionalidad.

Cuenta total: Resultado de conteo de parámetros.

X: Confiabilidad del proyecto, varía entre 1% a 100%.

Min(Y): Error mínimo aceptable al de la complejidad, el margen de error es igual a 0.01.

Fi: Valores de ajuste de complejidad, donde $i=1$ a $i=14$.

En la siguiente tabla se calcula el punto función, los cuales miden el software desde una perspectiva del usuario, sin tomar en cuenta los detalles de la codificación, el factor de ponderación se lo determina de acuerdo a las tablas de determinación de niveles de influencia de Albrecht, la cantidad es el número de los parámetros del sistema, obteniendo un total con la multiplicación de ambas.

PARÁMETRO	CANTIDAD	FACTOR DE PONDERACIÓN	TOTAL
ENTRADA DE USUARIOS	8	*4	32
SALIDAS DE USUARIO	18	*5	90
PETICIONES DE USUARIO	20	*4	80
ARCHIVO INTERNO LÓGICO	22	*4	88
INTERFAZ EXTERNA	0	*5	0
TOTAL			290

Tabla 4.1. Conteo paramétrico – Funcionalidad

Cada organización que utiliza métodos de puntos desarrolla criterios para determinar si una entrada es denominada simple o compleja. Los valores de F_i , se obtiene de los resultados de la siguiente tabla, bajo las ponderaciones descritos en la escala de 0 a 5.

Las preguntas fueron realizadas de manera conjunta entre el cliente y el desarrollador, se obtuvo las respuestas bajo las necesidades tanto del cliente y el desarrollador.

IMPORTANCIA	0	1	2	3	4	5
	Sin Importancia	Incremental	Moderado	Medio	Significativo	Esencial
1.- ¿Requiere el sistema copias de seguridad y de recuperación fiable?						X
2.- ¿Existe funciones de procesos distribuidos?						X

3.- ¿Se requiere comunicación de datos especializadas para transferir información a la aplicación u obtenerlas de ellas?				X		
4.- ¿Es crítico el rendimiento?				X		
5.- ¿El sistema web será ejecutado en el S.O. Actual?					X	
6.- ¿Se requiere una entrada interactiva para el sistema?					X	
7.- ¿Se requiere que el sistema tenga entradas a datos con múltiples ventanas?						X
8.- ¿Se actualiza los archivos de forma interactiva?						X
9.- ¿Son complejas las entradas, salidas, los archivos o las peticiones?				X		
10.- ¿Es complejo el procesamiento interno del sistema?					X	
11.- ¿Se ha diseñado el código para ser reutilizado?					X	
12.- ¿Están incluidas en el diseño la conversión y la instalación?					X	
13.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?				X		
14.- ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizados por el usuario?						X
CUENTA TOTAL				$\Sigma(fi) = 57$		

Tabla 4.2. Ajuste de complejidad

Para el ajuste se utiliza la ecuación:

$$PF = \text{cuenta total} * (\text{grado de confiabilidad} + \text{tasas de error} * \Sigma(fi))$$

Donde los valores de **grado de confiabilidad** y **tasas de error** ya están definidos por la International Function Point Users Group (IFPUG) / ISO 9126:

$$\text{Grado de confiabilidad} = 0.65$$

$$\text{Tasas de error} = 0.01$$

Entonces tenemos:

$$PF = 290 * (0.65 + 0.01 * 57)$$

$$PF = 353,8$$

Para el ajuste se utiliza la ecuación para hallar el punto función ideal al 100% de los factores que sería 70:

$$PF = 290 * (0.65 + 0.01 * 70)$$

$$PF \text{ ideal} = 391,5$$

Calculando del % funcionalidad real: $PF \text{ real} = PF \text{ obtenida} / PF \text{ ideal}$

$$FUNCIONALIDAD = \frac{353,8}{391,5} * 100 = 90,3$$

Por lo tanto:

$$FUNCIONALIDAD = 90 \%$$

4.2.1.2 USABILIDAD

Las métricas que tomaremos en cuenta para la usabilidad serán: Claridad, comprensión, capacidad de aprendizaje, operatividad atractiva, entre otros.

Se realizó un cuestionario a los usuarios del sistema con las preguntas necesarias para determinar las métricas mencionadas.

En la columna de evaluación de la tabla se cuentan las respuestas positivas a las preguntas formuladas a partir de las cuales se calcula la facilidad de uso del sistema, dado que se tiene 20 preguntas y se hicieron las pruebas con los 5 usuarios del sistema.

Se mostrarán los resultados obtenidos en la siguiente tabla (ver tabla 4.3).

Nro.	PREGUNTA	EVALUACIÓN
1	¿Le parece adecuada la selección de contenidos presentes en el menú principal? ¿Falta otras áreas de información? ¿Le parece que sobra algo?	5
2	Al hacer clic en un contenido del menú. ¿Halló en la información ofrecida lo que esperaba encontrar?	4
3	En general. ¿Los nombres de las opciones de los menús son suficientemente descriptivos de lo que se ofrece en las páginas hacia las cuales llevan? En particular. ¿Hubo alguno que lo confundió?	5
4	En caso de haber información relacionada con la que estaba viendo, ¿Se lo ofreció de manera simple? ¿O tuvo que volver a navegar por encontrarla?	4
5	En general. ¿Cree que los textos introductorios a información, cómo títulos y opciones del menú, son claros y concisos? ¿Le costó leerlos?	5
6	Los textos y párrafos desplegados por el sistema ¿Son de fácil lectura?	5
7	En general. ¿Cree que la navegación interna del sitio le permite explorarlo adecuadamente?	4
8	¿Existen elementos dentro de las páginas, que le	5

	permitan saber exactamente donde se encuentran dentro de este sitio? ¿Tuvo dificultades en encontrarlos?	
9	¿Pudo encontrar la manera de volver atrás y avanzar sin usar los botones del programa navegador?	5
10	¿Cree que las opciones secundarias de navegación, le ayudaron a usar el sitio de mejor manera?	4
11	¿Cree que existe sincronía entre la forma en que se despliegan los contenidos y en la que se desplaza a través de ellos y el uso físico de su teclado y mouse o pantalla táctil?	5
12	En términos generales. ¿Dónde cree que se sintió más perdido dentro del sitio?	5
13	En general. ¿Cree que las distintas áreas y secciones tanto de la página principal, cómo de sus páginas interiores son distinguibles visualmente una de otras?	5
14	¿Le parece proporcional la relación entre el menú principal, barra superior y el resto del contenido?	5
15	¿Se logran distinguirla zonas clickeables de las no clickeables?	4
16	Los íconos presentes en el sitio. ¿Le parece lo suficientemente legibles reconocibles y explicativos?	5
17	¿Qué le pareció la forma en que se muestran las imágenes en el sitio web?	5

18	¿Qué le parece el tiempo de carga de los reportes?	3
19	¿Considera que gráficamente el sitio está recargado?	3
20	¿Cree que los íconos son auto explicativos?	5

Tabla 4.3. Ajuste de complejidad

Para hallar la facilidad del uso del sistema se usará la siguiente fórmula de acuerdo a la ISO/IEC 9126 SLCP:

$$FU = [(\sum x_i / n * t) * 100] / N$$

Donde:

$\sum x_i$: Sumatoria de respuestas positivas.

n: Número de usuarios encuestados.

t: Número total de preguntas.

Reemplazando nuestros datos tenemos:

$$FU = (80/5*20) * 100 = 80$$

Por lo tanto:

$$\text{USABILIDAD} = 80 \%$$

4.2.1.3 CONFIABILIDAD

En esta sección evaluaremos algunos aspectos como: vencimiento, tolerancia a las fallas, capacidad de recuperación, entre otros.

Tiempo de servicio	Número de peticiones	Fallos encontrados	Probabilidad de fallo	Tiempo medio entre fallos
1 horas	10	0	0	
5 horas	20	0	0	
10 horas	50	2	0,04	40
20 horas	90	2	0,06	40
TOTAL				80

Tabla 4.4. Confiabilidad del Sistema

Las fallas que se encontró después de haber realizado el cálculo, están dadas por el promedio de fallas producidas en un cierto tiempo de servicio (PTFS).

$$PTFS = \frac{0 + 0 + 0,02 + 0,04}{4} = \frac{0,06}{4} = 0,015$$

Se puede interpretar como el número aproximado de fallas que se presentan cada 90 peticiones.

Posteriormente, el número de peticiones para que ocurra una falla, está dado por la siguiente relación.

$$TPF = \frac{0 + 0 + 40 + 40}{4} = \frac{80}{4} = 20$$

El resultado nos indicó que el sistema tiene la probabilidad de presentar fallas cada 20 horas durante su uso.

Por lo tanto, tenemos la siguiente relación:

$$1 - PTFS = 1 - 0,015 = 0,98$$

El resultado que obtuvimos nos indica que la probabilidad de que el sistema está libre de fallas, es de un 0,98 con una probabilidad de que ocurra una falla cada 90 peticiones.

Por lo tanto:

$$\text{CONFIABILIDAD} = 98 \%$$

4.2.1.4 MANTENIBILIDAD

Para calcular la mantenibilidad del sistema es necesario hallar el índice de madurez del sistema el cual es un indicador de estabilidad del producto y será determinado por la siguiente formula:

$$IMS = \frac{[MT - (FA + FC + FD)]}{MT} \quad (2)$$

Donde:

MT = es el número de módulos de la versión actual.

FC = número de módulos que han cambiado.

FA = número de módulos en la versión actual que se han añadido.

FD = número de módulos de la versión actual que se han borrado.

Para nuestro sistema tenemos los siguientes datos:

$$MT = 6, \quad FC = 1, \quad FA = 0, \quad FD = 0$$

Reemplazamos los valores en (2) y obtenemos:

$$IMS = \frac{[6 - (1 + 0 + 0)]}{6} = \frac{5}{6} = 0.83$$

Por lo tanto:

$$\text{MANTENIBILIDAD} = 82 \%$$

4.2.1.5 PORTABILIDAD

La portabilidad Se define como la característica que posee un software para ejecutarse en diferentes plataformas, el código fuente del software es capaz de reutilizarse en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra. A mayor portabilidad menor es la dependencia del software con respecto a la plataforma. La capacidad del software para ser trasladado de un entorno a otro. El entorno puede incluir entornos organizacionales, de hardware o de software.

Para medir la portabilidad se hará uso de la siguiente fórmula de acuerdo a la ISO/IEC 9126:

$$\text{GP} = 1 - \text{ET}/\text{ER}$$

Donde:

GP: Grado de portabilidad

ET: Recursos necesarios para llevar el sistema a otro entorno.

ER: Recursos necesarios para crear el sistema en el entorno residente.

Si:

GP > 0, la portabilidad es más rentable que el re-desarrollo.

GP < 0, el re-desarrollo la portabilidad es más rentable que la portabilidad.

GP = 0, la portabilidad es perfecta.

Para llevar el sistema web a otro entorno se necesita una memoria extraíble.

Para crear el sistema en el entorno residente se necesita inicialmente 1 servidor con sistema operativo este puede ser cualquiera (Windows, Linux o Mac OS en sus diferentes versiones), y servidor Apache, el entorno de tiempo de ejecución NodeJs, el framework opensource Angular, el gestor de base de datos MySQL, los cuales deben estar instalados en los servidores y un navegador web.

Con esta información requerida por la fórmula, se procede a calcular el grado de portabilidad:

$$GP = 1 - 1/5 = 1 - 0.20 = 80\%$$

Por lo tanto:

$$\text{PORTABILIDAD} = 80\%$$

4.2.2 RESULTADOS

A continuación, se mostrará la tabla con los resultados obtenidos de todas las métricas evaluadas en nuestro sistema (ver tabla 4.5).

MÉTRICA	PORCENTAJE
Funcionalidad	90 %
Usabilidad	80 %
Confiabilidad	98 %
Mantenibilidad	82%
Portabilidad	80%
EVALUACIÓN DE CALIDAD TOTAL	86 %

Tabla 4.5. Porcentaje total de Métricas del Calidad

En la tabla 4.5 se muestran los resultados de todas las métricas de calidad que se tomaron en cuenta de la norma ISO 9126. Podemos ver que el sistema presenta óptimas condiciones garantizando la seguridad y calidad de nuestro producto.

CAPÍTULO V

5. COSTO Y BENEFICIO

5.1 INTRODUCCIÓN

La técnica de Análisis de Costo y Beneficio, tiene como objetivo fundamental proporcionar una medida de la rentabilidad de un proyecto, mediante la comparación de los costos previstos con los beneficios esperados en la realización del mismo. Esta técnica se debe utilizar al comparar proyectos para la toma de decisiones.

5.2 COSTO DEL SISTEMA

El costo del sistema desarrollado se hará través de COCOMO II. El costo del sistema se planteará en tres partes: Desarrollo del software, Elaboración del Proyecto y Total del Software.

5.2.1 COCOMO II

El Modelo de Construcción de Costo COCOMO (Constructive Cost Model), es un modelo empírico se utiliza para la estimación de costos de un software. Incluye tres sub-modelos, cada uno ofrece un nivel de detalle y aproximación. En los tres modelos de estimación se utilizan las tres siguientes ecuaciones:

$$E = a(KLDC)^b * m(X), \text{ en persona mes}$$

$$D = c(E)^d, \text{ en meses}$$

$$P = E/D, \text{ en personas}$$

Dónde:

E: esfuerzo requerido por el proyecto, en mes.

D: Tiempo requerido por el proyecto, en meses.

P: Número de personas requerido por el proyecto.

a, b, c y d: constantes con valores definidos, según cada submodelo.

KLDC: Cantidad de líneas de código, en miles.

m(X): multiplicador que depende de 15 atributos.

A la vez cada modelo se divide en modos, estos modos son:

- Modo orgánico: es un pequeño grupo de programadores experimentados desarrollando software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas docenas de miles (medio).
- Modo semi – libre o semi – acoplado: Corresponde a un esquema intermedio entre el orgánico y el rígido, el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- Modo rígido o empotrado: El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único, siendo difícil basarse en la experiencia puesto que puede no haberla.

PROYECTO DE SOFTWARE	a _b	b _b	c _b	d _b
Orgánico	2.1	1.01	2.2	0.38
Semi-acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Tabla 5.1. Relación de valores en el modelo COCOMO

Fuente: Pressman, 2002

5.2.2 COSTO DEL SOFTWARE DESARROLLADO

Para la determinación del costo del software desarrollado, se hará uso del modelo constructivo de costos COCOMO II, usamos el dato obtenido de punto función en el cap.

4.2.1.1.

$$PF = 353,8$$

Para la conversión de los puntos de función a KLDC se convierte los puntos función a miles de líneas de código, teniendo en cuenta lo que se muestra en la siguiente tabla:

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2.5	128
ANSI BASIC	5	6464
JAVA	6	53
PL/I	4	80
ANSI COBOL 74	3	107
VISUAL BASIC	7.00	46
ASP	9.00	36
Angula	11.00	29
VISUAL C++	9.50	34

Tabla 5.2. Conversión de puntos función

Fuente: Pressman, 2002

Calculando las líneas de código en la siguiente ecuación tenemos:

$$LDC = PF * factor / PF = 354 * 29$$

$$LDC = 10266$$

Para convertirlo a KLDC dividimos LDC entre 1000. Calculando el número de líneas distribuidas en el sistema KLCD se tiene:

$$KLDC = total LCD / 1000$$

$$KLDC = 10266 / 1000$$

$$KLDC = 10.26$$

Por tanto, existen 17 líneas de código distribuidas para el proyecto.

A continuación, haremos el cálculo del esfuerzo necesario para la programación del sistema.

La ecuación que nos ayudará a hallar el esfuerzo, viene dada de la siguiente manera:

$$E = a*(KLDC)^b, \text{ en persona/mes}$$

Donde:

E: Esfuerzo expresado en personas por mes.

a, b: Constantes empíricas

KLDC: Número estimado de código fuente en miles distribuidas.

Como nuestro proyecto es del tipo Orgánico utilizaremos $a = 2.4$ y $b = 1.05$ (Ver tabla 5.1) reemplazando estos valores en la ecuación tenemos:

$$E = 2.4 * (10,26)^{1.05} = 26.92$$

$$E = 26.92$$

Calculando D con $c = 1.05$ y $d = 0.38$ (Ver tabla 5.1) tenemos:

$$D = c(E)^d \text{ en meses}$$

$$D = 1.05 * (10.26)^{0.38}$$

$$D = 2.54$$

El proyecto deberá tener un desarrollo aproximado de 3 meses.

Para calcular el personal requerido (en este caso 1) número de programadores se obtiene con la siguiente formula:

$$P = \frac{E}{D} \text{ en personas}$$

Reemplazando los datos ya conocidos se tiene:

$$P = \frac{26.92}{3} = 8.97$$

$$P = 9 \text{ [programadores]}$$

El salario promedio de un programador oscila entre los 2000 y 4000 Bs., en nuestro caso tomaremos un promedio con un valor de 3000 Bs, a partir de este monto podemos calcular el costo del software desarrollado:

Costo del software desarrollado = Número de programadores * salario de un programador

Costo del software desarrollado por persona = $9 * 3000$

Costo del software desarrollado por persona = 27000 Bs.

Como el desarrollo de software se lo estima en 3 meses tendremos

Costo total de desarrollo = costo del software desarrollado * Número de meses Costo total

de desarrollo = $27000 * 3$

Costo total de desarrollo = 71000 Bs.

5.2.3 COSTO DE IMPLEMENTACIÓN DEL SISTEMA

Las herramientas de software son Angular y MySQL, son de uso libre y las mismas que son utilizadas por muchas empresas por ser software libre.

5.2.4 COSTO DE ELABORACIÓN DEL SISTEMA

El costo de elaboración de proyecto hace referencia a los gastos que se realizaron a lo largo de las diferentes fases de Scrum, los mismos se presentan en la siguiente tabla, los datos son obtenidos por criterio propio:

DETALLE	IMPORTE(\$us)
Análisis y diseño del proyecto	250
Material de escritorio	30
Internet	200
Otros	30
Total	510

Tabla 5.3. Costos elaboración

5.2.5 COSTO TOTAL DE SOFTWARE

El costo total del software se muestra de forma detallada en la siguiente tabla:

Convertimos a dólares el costo de desarrollo en bolivianos.

DETALLE	IMPORTE (\$us)
Costo de desarrollo	3884.89
Costo de elaboración del software	510
Total	4394.89

Tabla 5.4. Costo total

Por tanto, el costo total del proyecto es igual a **4394.89** \$, su equivalente en bolivianos es de 30764,24 Bs.

5.3 BENEFICIO

Para evaluar el beneficio se calculará con el método del Valor Actual Neto (VAN) y la tasa interna de Retorno (TIR).

5.3.1 CÁLCULO DEL VALOR ANULA NETO (VAN)

El VAN es un procedimiento que permite calcular el valor presente de un determinado número de flujos de futuros ingresos y egresos que tendrá el proyecto. La metodología consiste en descontar al momento actual, es decir actualizar mediante una tasa todos los flujos de caja futuros del proyecto, a este valor se le resta la inversión inicial, así el valor obtenido es el valor actual neto del proyecto.

$$VAN = \sum \frac{Ganancias}{(1 + k) * n} - I_0$$

Donde:

VAN: Valor actual neto.

Ganancias: Ingreso del flujo anual.

I_0 : Valor del desembolso inicial de la inversión.

k: Tasa de descuento o tasa de interés al préstamo.

n: Número de periodos considerados.

Los valores de ganancia esperados para el presente proyecto se calculan para 3 años, para este caso en particular utilizaremos una tasa de descuento del 11%, ya que es la tasa actual de interés de préstamo en las entidades financieras. Para calcular el valor del VAN se tiene lo siguiente:

$$\text{Inversión} = \text{Bs. } 30764,24$$

$$\text{TD} = 11\%$$

Los valores de ganancia esperados fueron proporcionados por la Asociación, se detallan en la siguiente tabla:

TIEMPO	1° AÑO	2° AÑO	3° AÑO
Flujo de caja neto	25000	23000	21000

Tabla 5.5. Cantidad por año
Fuente: Consultorio Dental Ortho-Braes.

Entonces para hallar el VAN tenemos:

$$VAN = \frac{25000}{(1 + 0.11)^1} + \frac{23000}{(1 + 0.11)^2} + \frac{21000}{(1 + 0.11)^3} - 30764,24$$

$$VAN = \frac{25000}{1.11} + \frac{23000}{2.22} + \frac{21000}{3.33} - 30764,24$$

$$VAN = 22522 + 10360 + 6306 - 30764,24$$

$$VAN = 8423,76$$

Por tanto, la ganancia que se espera recibir del sistema al cabo de 3 años es de Bs. 8423,76, para ver si el proyecto es rentable utilizamos la siguiente tabla:

VALOR DE VAN	INTERPRETACIÓN
VAN > 0	El proyecto es rentable
VAN = 0	El proyecto es rentable porque ya está incorporada ganancia de la tasa de interés
VAN < 0	El proyecto no es rentable

Tabla 5.6. Interpretación del VAN

Como el resultado que obtuvimos es de VAN = 8423,76 podemos afirmar que nuestro proyecto es rentable.

5.3.2 LA TASA INTERNA DE RETORNO TIR

El TIR es una tasa de descuento TD de un proyecto de inversión para que sea rentable. Cuando el VAN toma un valor igual a 0, k pasa a llamarse TIR. En términos generales: Las inversiones más interesantes son aquellas que proporcionan mayor TIR.

- Si TIR es inferior a la tasa de descuento de la empresa, la inversión debería ser desestimada.
- Si TIR es superior la tasa de descuento de la empresa la inversión es factible.

$$TIR = -I_0 \frac{Q_1}{(1+k)^1} + \frac{Q_2}{(1+k)^2} + \dots + \frac{Q_n}{(1+k)^n}$$

Donde:

TIR: Tasa interna de retorno

I_0 : Valor del desembolso inicial de la inversión

k: tasa de interés de ahorro

n: Número de periodos considerados.

Entonces para hallar el TIR se necesita la inversión de Bs. 30764,24

Para hallar el TIR se hace uso de la fórmula del VAN, solo hace que el valor de VAN sea igual a 0, a continuación, se halla la tasa de descuento:

$$0 = \frac{25000}{1.11} + \frac{23000}{2.22} + \frac{21000}{3.33} - 30764,24$$

Con la ayuda de la hoja de cálculo podemos resolver la ecuación y hallar el valor de i

	A	B	C	D
1	30764,24			
2	25000			
3	23000			
4	21000			
5	18%			

Figura 5.1. Cálculo del TIR

Lo cual indica que el proyecto está en condiciones de retornar 18% de la inversión cada año lo que indica que el proyecto es muy rentable, además si la tasa fuera mayor de 20% el proyecto empezaría a no ser rentable. Pues el BNA empezaría a ser menor que la inversión Y si la tasa fuera menor al 11% (como en el caso del VAN donde la tasa es de 11% y en nuestro caso 18%), a menor tasa, el proyecto es más rentable, pues el BNA es cada vez mayor que la inversión. También se puede mencionar que los beneficios calculados en dinero tienen otros como ser el ahorro del tiempo en el proceso de realizar publicaciones.

TIR a 3 años = 18%

Lo cual nos indica que el proyecto sea siempre rentable la tasa de descuento TD no debe ser mayor a 20%

5.4 COSTO/BENEFICIO

Para hallar el costo/beneficio de un proyecto aplicamos la siguiente ecuación:

C: Costo total del proyecto

B: VAN

$$C/B$$

Reemplazando en la ecuación con los valores hallados anteriormente (cap. 5.2.5 y 5.3.1), tenemos:

$$C/B = 30764.24/8423,76$$

$$C/B = 3.65$$

Por tanto, por cada boliviano invertido, la asociación tiene una ganancia de **Bs. 3.65**

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

Luego de haber finalizado el proyecto de grado “Sistema Web de administración, control y gestión de historiales clínicos Caso: consultorio dental Ortho-Braes”, se ha logrado alcanzar el objetivo principal planteado, cumpliendo con las necesidades del consultorio dental.

6.1 CONCLUSIONES

En consideración de los objetivos secundarios se puede concluir que:

- Con el desarrollo del sistema se logró disminuir el tiempo de programación de citas para el consultorio médico.
- Se logró facilitar al usuario el acceso a una información detallada de una pieza dental que es tratada en múltiples sesiones.
- Se logró mejorar la administración de los tratamientos que se realizan en el consultorio del paciente y de la pieza dental mediante fotos y radiografías que se almacenan de manera digital.
- Se logró facilitar el control de pagos de los tratamientos del consultorio, que se realizan en múltiples sesiones y tienen distintos tipos de materia prima.
- Los costos de tratamientos que son regidos por el colegio de odontólogos son más visibles y de fácil acceso por parte del médico.

6.2 RECOMENDACIONES

De acuerdo a los logros obtenidos del presente proyecto de grado, se sugieren algunas recomendaciones.

- Se recomienda registrar a todos los pacientes que realicen una consulta o un tratamiento en el consultorio dental, para poder interactuar con él de manera específica, con los procesos que ofrece el sistema.
- Cuando se quiera agendar a un paciente, se recomienda revisar visualmente los datos

del paciente agendado para que el sistema pueda mostrar la información correcta.

- Cada vez que el paciente solicite un tratamiento, se recomienda no olvidar iniciar el tratamiento en el sistema para que este tratamiento se guarde como información del paciente.
- La revisión periódica del sistema cada cierto periodo de tiempo es recomendable para la eficiencia y un funcionamiento adecuado.



7. BIBLIOGRAFÍA

Huanca S. Sistema web de control de pagos, citas e historiales clínicos Caso: Clínica Dental Lavadent (Proyecto de Grado). Universidad Mayor de San Andrés, Informática, La Paz, Bolivia. 2015.

Lozano R. Sistema de Administración y Control de Historiales clínicos para los Consultorios Clínicos de la U.M.S.A. (Proyecto de Grado). Universidad Mayor de San Andrés, Informática, La Paz, Bolivia. 2014.

Chirina M. & Morales M. Sistema de Seguimiento a Paciente Control-Asignación Proyecciones Hospital La Paz (Proyecto de Grado). Universidad Mayor de San Andrés, Informática, La Paz, Bolivia. 2000.

Hamidian & Ospino, ¿Por qué los sistemas de información son esenciales? 2015.

Hidalgo, Las Nuevas Tecnologías de la Información y Comunicación. 2017.

Bolaños, Urrea, Gómez. La ingeniería web, 2015.

Nieves, Ucán, Menéndez. UWE en Sistema de Recomendación de Objetos de Aprendizaje Aplicando Ingeniería Web: Un Método en Caso de Estudio. Revista Latinoamericana de Ingeniería de Software. 2014.

Pinzón, Ingeniería Web: Una Metodología para el Desarrollo de Aplicaciones Web Escalables y Sostenibles. 2017.

Pressman, Ingeniería de Software, Quinta edición. Editorial Mc Graw Hill. México, 2004.

Schwaber, K., & Sutherland, J. La Guía Definitiva de Scrum: Las Reglas del Juego. Obtenido de www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf 2013.

Trigas, M. (s.f.). Metodología Scrum. Obtenido de: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/2/mtrigasTFC0612presentacion.pdf> 2012.

F. J. García-Peñalvo, "Ingeniería del Software,". Área de Ciencia de la Computación e Inteligencia Artificial pp. 277-388, Salamanca, España. Universidad de Salamanca, 2018.

R. S. Pressman y B. R. Maxim, Software Engineering: A practitioner's approach, 8th ed. New York, NY, USA: McGraw-Hill Education. 2015.

J. Nogueira, C. Jones y Luqi, "Surfing the Edge of Chaos: Applications to Software Engineering," , Monterey, CA, USA, 2000.

Chen, P. The Entity-Relationship model - Toward a unified view of data., ACM Transactions on Database System, Vol.1, No.1, March, pp.9-36. 1976.

Fuster, Gonzalo; Llorens Juan; Fuentes José; Morato Jorge; Martínez Paloma. Las jerarquías conceptuales en UML: comparando la norma ISO 2788 con el metamodelo de UML. Técnica Administrativa, Vol. 10, No. 1. 2011

Gornik, D. (2003). Relational Modeling with UML. A technical discussion of UML. Disponible en el sitio web:

http://www.ibm.com/developerworks/rational/library/content/03July/2500/2785/2785_uuml.pdf 2011

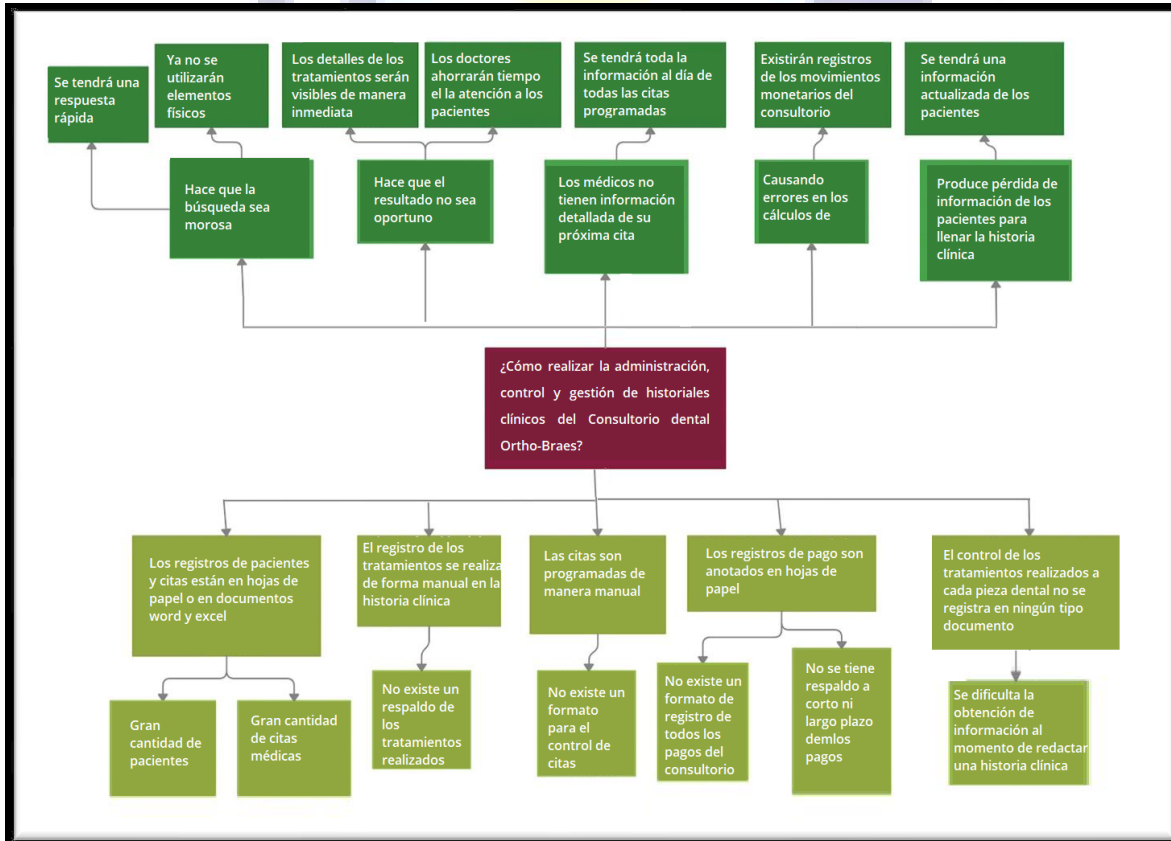
Hofreiter, B., Huemer, C., Liegl, P., Mosser, R., Schuster, R., Zapletal, M. (2007). Modeling e-Government processes with UMM. *Informatica. An International Journal of Computing and Informatics*, 31(4): 407-417.

Muñoz M., Pedro J., Modelado Web. Disponible en el sitio web: <http://www.it.uc3m.es/pedmume/> 2008

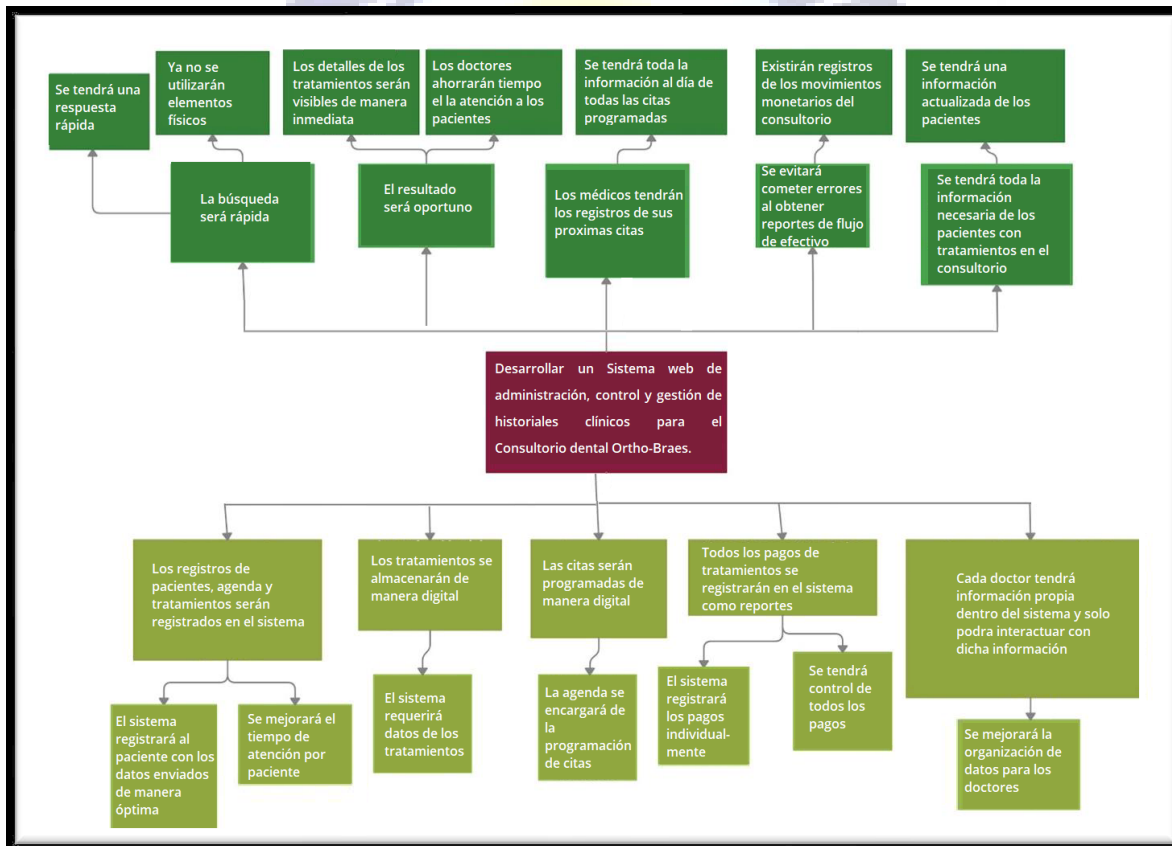


8. ANEXOS

8.1 ANEXO A – ÁRBOL DE PROBLEMAS



8.2 ANEXO B – ÁRBOL DE OBJETIVOS



8.3 ANEXO C – CUESTIONARIOS



Pruebas de navegación

Consultorio Dental Ortho-Braes

Usuario	Brayan Almazán Garate
Encuestador	
Versión del sistema:	1.0
Fecha:	30-Nov-2021

- ¿Le parece adecuada la selección de contenidos presentes en el menú principal?
¿Falta otras áreas de información? ¿Le parece que sobra algo?
R. Si, me parece adecuada la selección de contenidos.....
- Al hacer clic en un contenido del menú. ¿Halló en la información ofrecida lo que esperaba encontrar?
R. Si.....
- En general. ¿Los nombres de las opciones de los menús son suficientemente descriptivos de lo que se ofrece en las páginas hacia las cuales llevan? En particular. ¿Hubo alguno que lo confundió?
R. No, hubo nada confuso.....
- En caso de haber información relacionada con la que estaba viendo, ¿Se lo ofreció de manera simple? ¿O tuvo que volver a navegar por encontrarla?
R. Si.....
- En general. ¿Cree que los textos introductorios a información, cómo títulos y opciones del menú, son claros y concisos? ¿Le costó leerlos?
R. Son claros.....
- Los textos y párrafos desplegados por el sistema ¿Son de fácil lectura?
R. Si.....

7. En general. ¿Cree que la navegación interna del sitio le permite explorarlo adecuadamente?
 R... Si
8. ¿Existen elementos dentro de las páginas, que le permitan saber exactamente donde se encuentran dentro de este sitio? ¿Tuvo dificultades en encontrarlos?
 R... No hubo dificultades en la navegación
9. ¿Pudo encontrar la manera de volver atrás y avanzar sin usar los botones del programa navegador?
 R... Si
10. ¿Cree que las opciones secundarias de navegación, le ayudaron a usar el sitio de mejor manera?
 R... Si
11. ¿Cree que existe sincronía entre la forma en que se despliegan los contenidos y en la que se desplaza a través de ellos y el uso físico de su teclado y mouse o pantalla táctil?
 R... Si, en varios contenidos
12. En términos generales. ¿Dónde cree que se sintió más perdido dentro del sitio?
 R... En el componente de información del paciente
13. En general. ¿Cree que las distintas áreas y secciones tanto de la página principal cómo de sus páginas interiores son distinguibles visualmente una de otras?
 R... Si
14. ¿Le parece proporcional la relación entre el menú principal, barra superior y el resto del contenido?
 R... Si

15. ¿Se logran distinguir las zonas clickeables de las no clickeables?

R. Si, en varios casos

16. Los íconos presentes en el sitio. ¿Le parece lo suficientemente legibles reconocibles y explicativos?

R. Si, muchos de ellos

17. ¿Qué le pareció la forma en que se muestran las imágenes en el sitio web?

R. Muy clara

18. ¿Qué le parece el tiempo de carga de los reportes?

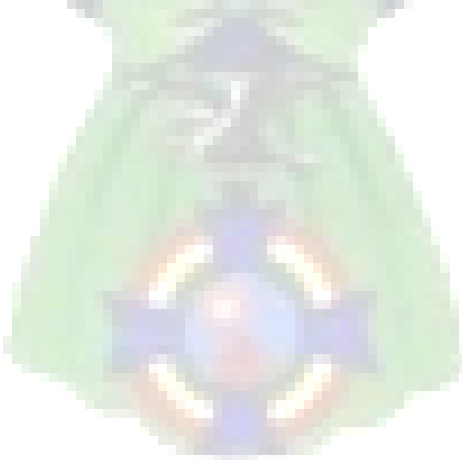
R. Eficiente

19. ¿Considera que gráficamente el sitio está recargado?

R. Si

20. ¿Cree que los íconos son auto explicativos?

R. Si



8.4 ANEXO D – CRONOGRAMA DE AVANCE

ACTIVIDADES	NR O. DÍAS	DEL 1 DE JULIO AL 30 DE NOVIEMBRE																					
		JULIO				AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE				DIC	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
Redacción del Capítulo II - Marco Teórico	7	█																					
Desarrollo del Capítulo III - Marco Aplicativo	7	█																					
PRE – GAME	14		█	█																			
Taller de Recopilación de Preguntas	14		█	█																			
GAME	14			█	█																		
1ra Iteración	16				█	█																	
2da Iteración	20					█	█	█															
3ra Iteración	17						█	█	█														
4ta Iteración	15									█	█												
5ta Iteración	14										█	█											
6ta Iteración	21											█	█	█									
POST-GAME	28													█	█	█	█						
Implementación	28														█	█	█	█					
Retroalimentación	28															█	█	█	█				
Redacción del Capítulo III - Marco Aplicativo	7																				█		
Redacción del Capítulo IV – Calidad y Seguridad	7																					█	
Redacción del Capítulo V – Costo Beneficio Capítulo VI -Conclusiones y Recomendaciones	7																					█	



DOCUMENTACIÓN

La Paz, marzo de 2022

Señor

Lic. Rubén Alcón López

Director

Carrera de Informática

Facultad de Ciencias Puras y Naturales

Presente

REF. AVAL DE CONFORMIDAD DE PROYECTO DE GRADO

De mi consideración

Por intermedio de la presente, y en mi calidad de Tutor Metodológico, tengo a bien dirigirme a su autoridad, para dar a conocer que luego de efectuar el seguimiento a la estructura y contenido del Proyecto de Grado, titulado: "SISTEMA WEB DE ADMINISTRACIÓN, CONTROL Y GESTIÓN DE HISTORIALES CLÍNICOS CASO: CONSULTORIO DENTAL ORTHO-BRAES", elaborado por el postulante Diego Vladimir Mamani Nina con C.I. 9100638 LP, me corresponde **dar mi CONFORMIDAD Y AVAL**, para que el mismo proceda a la **DEFENSA PÚBLICA**, de acuerdo a normas y reglamentos universitarios vigentes.

Sin otro particular, me despido de usted con las consideraciones más distinguidas.



PhD. Javier Hugo Gustavo Reyes Pacheco

Docente Tutor

CONSULTORIO DENTAL ORTHO-BRAES

ESPECIALIDAD EN ODONTOLOGÍA

La Paz, junio de 2021

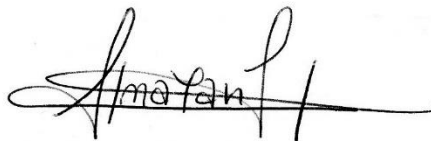
Señor
PhD. José María Tapia Baltazar
Director
Carrera de Informática
Facultad de Ciencias Puras y Naturales
Presente

Ref. Aval de Conformidad del Perfil de Proyecto de Grado

De mi mayor consideración

Tengo a bien dirigirme a su persona para darle a conocer, que luego de efectuar el seguimiento a la estructura y contenido del Perfil de Proyecto de Grado, titulado "Sistema Web de Administración, Control y Gestión de Historiales Clínicos Caso: Consultorio Dental Ortho-Braes", elaborado por el universitario Diego Vladimir Mamani Nina, con CI:9100638 LP, y habiendo el postulante realizado las respectivas entrevistas y reuniones, y no existiendo impedimento alguno en la propuesta, me corresponde dar mi Conformidad Aval, recomendando que el mencionado universitario, inicie sus correspondientes tramites, para su respectiva aprobación de la Materia y pueda empezar a realizar el Proyecto de Grado.

Atentamente.



Dr. Brayan Almazán Gárate
CIRUJANO DENTISTA
Mat. Prof. A-6988443