

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA WEB PARA LA ADMINISTRACIÓN DE CONSULTAS
MÉDICAS DENTALES”

CASO: CLINICA DENTAL PATZI EMANUEL

Para optar al título de Licenciatura en Ingeniería de Sistemas
Mención: INFORMATICA Y COMUNICACIONES

Postulante: Israel Ivan Mendoza Apaza

Tutor Metodológico: Ing. Maricel Yarari Mamani

Tutor Especialista: Lic. Gladys Francisca Chuquimia Mamani

Tutor Revisor: Lic. Freddy Salgueiro Trujillo

EL ALTO – BOLIVIA
2020

Dedicatoria

El presente trabajo está dedicado a Dios por ser mi guía, por darme la maravillosa oportunidad y la dicha de la vida al regalarme la presencia de mi familia y todo lo necesario para poder desarrollarme en mis estudios y como ser humano.

A mis padres por estar siempre conmigo apoyándome incondicionalmente para culminar mi carrera profesional.

AGRADECIMIENTOS

Quiero agradecer primeramente a Dios por ser siempre la guía en mi vida, por darme sabiduría y fortaleza

A mi familia a mis padres, hermanos le doy gracias por el apoyo amor y comprensión que siempre me han brindado.

Agradecer a mi tutor revisor Lic. Freddy Salgueiro Trujillo, por brindarme su conocimiento, experiencia, apoyo incondicional, el tiempo dedicado y por su paciencia, que me ayudaron a culminar el presente proyecto.

Agradecer a mi tutor metodológico Ing. Maricel Yarari Mamani, por su tiempo, comprensión, paciencia y orientación con sus conocimientos en la realización de este proyecto.

Agradecer a mi tutor especialista Lic. Gladys Francisca Chuquimia Mamani, por su apoyo desinteresado, y su conocimiento compartido que me ayudo a construir este proyecto.

RESUMEN

El presente proyecto se desarrolla para dar conformidad a los requisitos exigidos por la Universidad Pública de El Alto para dar fin a la carrera de Ingeniería de Sistemas mediante el proceso de titulación. El proyecto lleva por nombre "Sistema web para la administración de consultas médicas dentales" para la Clínica Dental Patzi Emanuel.

EL desarrollo del proyecto trata de brindar un apoyo al proceso de Atención a las Consultas Médicas Dentales realizadas por el personal de la institución, mismo que componen las tareas como, recibir e ingresar al paciente en el sistema, registrar datos personales, verificación de tratamiento, registro del pago y emisión de la receta médica, emisión de reportes de las consultas dentales atendidas por fechas.

Para el desarrollo del proyecto se utilizó el framework Codeigniter que se basa en el modelo MVC (modelo, vista, controlador) y herramientas de desarrollo de aplicaciones Web como ser PHP con la utilización de la metodología ágil XP (Programación Extrema). En cuanto a los aspectos visuales, se utilizó Bootstrap.

Asimismo, para determinar la calidad de sistema Web desarrollado, se hace uso de los factores de calidad ISO 9126.

Finalmente se realiza el análisis de costos utilizando COCO

ÍNDICE DE CONTENIDOS

	Pag.
1. MARCO PRELIMINAR.....	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES.....	2
1.3. PLANTEAMIENTO DEL PROBLEMA	4
1.3.1. Problema Principal.....	4
1.3.2. Problemas Secundarios.....	4
1.4. OBJETIVOS.....	5
1.4.1. Objetivo General	5
1.4.2. Objetivo Específicos	5
1.5. JUSTIFICACIÓN.....	6
1.5.1. Técnica	6
1.5.2. Económica	6
1.5.3. Social	6
1.6. METODOLOGIA	7
Metodología XP (Extreme Programming)	7
1.7. HERRAMIENTAS.....	9
1.8. LIMITES Y ALCANCES	10
1.8.1. Limites.....	10
1.8.2. Alcances	11
1.9. APORTES.....	12
2. MARCO TEORICO	13
2.1. INTRODUCCION	13
2.2. SISTEMA	13
2.3. INTRANET	14
2.3.1. Elementos de la Intranet	15
2.3.2. ¿Cómo funciona una Intranet?.....	16
2.3.3. Ventajas y Desventajas de la intranet.....	17
2.4. SISTEMA WEB	18
2.4.1. Ventajas y Desventajas de un Sistema Web	18

2.5. INGENIERIA DE SOFTWARE	19
2.5.1. MODELOS DE DESARROLLO DE SOFTWARE	20
2.6. METODOLOGIA DE DESARROLLO XP	21
2.6.1. Valores de XP	23
2.6.2. Los diferentes roles dentro de XP	24
2.6.3. Fases de la metodología XP	26
2.7. HERRAMIENTAS DE DESARROLLO	31
2.7.1. Gestor de base de datos MySQL	31
2.7.2. Servidor Apache	32
2.7.3. Lenguaje de programación	33
2.7.4. Framework	35
2.7.5. Codeigniter.....	36
2.8. FACTOR DE CALIDAD ISO 9126.....	40
2.9. SEGURIDAD DE LA INFORMACION	44
2.10. ANALISIS DE COSTO DEL SISTEMA.....	44
2.10.1. Características e inconvenientes	45
2.10.2. Modelos de estimación	46
2.10.3. Modelo básico.....	47
2.10.4. Modelo intermedio	48
2.10.5. Modelo Detallado	51
2.11. HISTORIAL CLINICO.....	51
2.12. ODONTOLOGIA	52
3. MARCO APLICATIVO.....	55
3.1. INTRODUCCIÓN	55
3.2. ANALISIS DE LA SITUACION ACTUAL	56
3.3. PLANIFICACION.....	57
3.3.1. Identificación de Tipos de Usuario (Roles).....	57
3.3.2. Historias de Usuario.....	57
3.3.3. Identificación de Tareas	64
3.3.4. Iteraciones	71
3.3.5. Plan de entregas	72

3.4. DISEÑO	72
3.4.1. Tarjetas CRC	75
3.5. DESARROLLO.....	78
3.5.1. Implantación de las Interfaces de Usuario	78
3.6. PRUEBAS.....	89
3.6.1. Pruebas de Aceptación	89
3.7. PRUEBAS DE SOFTWARE.....	94
3.7.1. Pruebas de Caja Blanca	94
3.7.2. Prueba de Caja Negra	97
3.7.3. Pruebas de Estrés	100
3.8. ANALISIS DE CALIDAD	101
3.8.1. NORMA ISO 9126	101
3.9. ANALISIS DE COSTOS.....	108
3.10. SEGURIDAD.....	112
3.10.1. Seguridad a Nivel de Base de Datos	112
3.10.2. Seguridad a Nivel Aplicación	113
4. CONCLUSIONES Y RECOMENDACIONES	114
4.1. CONCLUSIONES	114
4.2. RECOMENDACIONES	114
BIBLIOGRAFIA.....	116
ANEXOS.....	118

ÍNDICE DE FIGURAS

	Pag.
Figura 1. Internet – Extranet – Intranet	15
Figura 2. Evolución de los largos ciclos de desarrollo en cascada (a) a ciclos iterativos más cortos (b) y a la mezcla que hace XP.....	22
Figura 3 Extreme Programming (XP)	23
Figura 4 el proceso de la programación extrema	27
Figura 5. Ciclo de vida del MVC.....	38
Figura 7. Historial Clínico	56
Figura 8. Modelo de la base de datos propuesto	74
Figura 9. Pantalla: Control de acceso al sistema	79
Figura 10. Administración de la clínica dental.....	79
Figura 11. pantalla registro de usuario.....	80
Figura 12. Pantalla lista de odontólogos	80
Figura 13. Pantalla: Modificación dato del Odontólogo	81
Figura 14. Pantalla: listado de pacientes	81
Figura 15. Pantalla: Registro de pacientes	82
Figura 16. Pantalla: Modificar de pacientes	82
Figura 17. Pantalla: Consulta de pacientes.....	83
Figura 18. Pantalla: Agregar tratamiento	84
Figura 19. Pantalla: Modificar tratamiento.....	84
Figura 20. Pantalla: consulta del tratamiento	85
Figura 21. Pantalla: Historial del tratamiento.....	85
Figura 22. Pantalla: Historial de Atención	86
Figura 23. Pantalla: Administración de pagos.....	86
Figura 24. Pantalla: Ingreso de pagos	87
Figura 25. Pantalla: Administración de Deudas	87
Figura 26. Pantalla: Administración receta Medica	88
Figura 27. Pantalla: Registro de nueva Receta.....	88
Figura 28. Pantalla: Registro de nueva Cita.....	89
Figura 29. Prueba de caja blanca: Inicio de sesión.....	95

Figura 30. Prueba de caja blanca: atención al cliente	96
Figura 31. Prueba de caja negra: Inicio de sesión	98
Figura 32. Prueba de caja negra: Registro de paciente	99
Figura 33. Prueba de estrés: formulario paciente	100

ÍNDICE DE TABLAS

	Pag.
Tabla 1- Comparación de Internet, Intranet y Extranet	15
Tabla 2 - Diferencias entre metodologías ágiles y no ágiles	21
Tabla 3 - Características de la norma ISO-9126	40
Tabla 4 - Constantes para el cálculo de costes Modelo Básico	48
Tabla 5 - Constantes para el cálculo de costes Modelo Intermedio	48
Tabla 6 - Atributos para el cálculo de costes.....	50
Tabla 7 - Fases y Procesos de la Metodología XP	55
Tabla 8 - Clasificación e Identificación de Roles	57
Tabla 9 - Historia de usuario: Control de acceso al sistema	58
Tabla 10 - Historia de usuario: Ingresar datos del odontólogo	58
Tabla 11 - Historia de usuario: Modificar datos del odontólogo	59
Tabla 12 - Historia de usuario: Ingresar datos del paciente	59
Tabla 13 - Historia de usuario: Modificar datos del paciente	60
Tabla 14 - Historia de usuario: Consultar paciente	61
Tabla 15 - Historia de usuario: administrar pagos.....	61
Tabla 16 - Historia de usuario: Odontólogo actualiza datos del tratamiento	62
Tabla 17 - Historia de usuario: Consulta del tratamiento	62
Tabla 18 - Historia de usuario: Administrar historial del paciente.....	63
Tabla 19 - Historia de usuario: Administrar pagos	63
Tabla 20 - Historia de usuario: Administrar citas medicas	64
Tabla 21 - Tareas de historias de usuario.....	65
Tabla 22 - Tarea: Control de acceso al sistema.....	66
Tabla 23 - Tarea: Ingresar datos del odontólogo	66
Tabla 24 - Tarea: Modificar datos del odontólogo	67
Tabla 25 - Tarea: Ingresar datos del paciente.....	67
Tabla 26 - Tarea: Modificar datos del paciente	68
Tabla 27- Tarea: Administrar historial del paciente	68
Tabla 28 - Tarea: Administrar historial del paciente	68
Tabla 29- Tarea: Odontólogo actualiza datos del tratamiento.....	69

Tabla 30 - Tarea: Consulta del tratamiento	69
Tabla 31 - Tarea: Administrar historial del paciente	70
Tabla 32 - Tarea: Administrar pagos.....	70
Tabla 33 - Tarea: Administrar citas medicas	70
Tabla 34 - Planificación de iteraciones.....	71
Tabla 35 - Plan de entregas	72
Tabla 36 - Tarjeta CRC: Autenticación.....	75
Tabla 37 - Tarjeta CRC: Usuario.....	75
Tabla 38 - Tarjeta CRC: Rol.....	76
Tabla 39 - Tarjeta CRC: Paciente	76
Tabla 40 - Tarjeta CRC: Tratamiento	76
Tabla 41 - Tarjeta CRC: Detalle	76
Tabla 42 - Tarjeta CRC: Odontograma	77
Tabla 43 - Tarjeta CRC: Detalle	77
Tabla 44 - Tarjeta CRC: Historial	77
Tabla 45 - Tarjeta CRC: Cita.....	77
Tabla 46 - Tarjeta CRC: Atención	78
Tabla 47 - Prueba de aceptación: Control de acceso al sistema	90
Tabla 48 - Pruebas de Aceptación: Ingresar datos del odontólogo.....	90
Tabla 49 - Pruebas de Aceptación: Modificar datos del odontólogo	91
Tabla 50 - Pruebas de Aceptación: Ingresar datos del paciente	91
Tabla 51 - Pruebas de Aceptación: Modificar datos del paciente.....	91
Tabla 52 - Pruebas de Aceptación: Consultar paciente	92
Tabla 53 - Pruebas de Aceptación: Crear tratamiento odontológico del paciente..	92
Tabla 54 - Pruebas de Aceptación: Odontólogo actualiza datos del tratamiento ...	92
Tabla 55 - Pruebas de Aceptación: Consulta del tratamiento	93
Tabla 56 - Prueba de Aceptación Consulta del tratamiento	93
Tabla 57 - Prueba de Aceptación: Administrar pagos	93
Tabla 58 - Prueba de Aceptación: Administrar citas medicas	94
Tabla 59 - Evaluación de flujo sesión.....	96
Tabla 60 - Evaluación de flujo atención paciente	97

Tabla 61 - Prueba de caja negra: inicio sesión	98
Tabla 62 - Prueba de caja negra: Registro de paciente	99
Tabla 63 - Resultados obtenidos de la prueba de estrés	101
Tabla 64 - factores de ponderación.....	101
Tabla 65 - Tabla de valores de ajuste de complejidad	102
Tabla 68 - Atributos y sus valores usados en el cálculo de FAE.....	109
Tabla 69 - Seguridad a nivel de Aplicación	113

CAPITULO I

MARCO PRELIMINAR

1. MARCO PRELIMINAR

1.1. INTRODUCCIÓN

Hoy en día los aportes tecnológicos son parte de grandes soluciones en la vida cotidiana, es por esto que distintos establecimientos se apoyan en sistemas, buscando mejorar por distintos medios la relación con sus clientes antiguos o nuevos para satisfacer sus necesidades, ya que los mismos tienen un papel importante para el funcionamiento del negocio o entidad.

De este modo, el presente proyecto nace con el objetivo de desarrollar un sistema informático capaz de satisfacer las necesidades de un consultorio odontológico, que permita el control de los pacientes y brinde a los odontólogos una herramienta adecuada para poder agilizar la atención de cada paciente. Logrando ahorrar espacio físico en el almacenamiento del historial clínico, optimizando los tiempos para la búsqueda de información, alcanzando un mayor control en el flujo de citas y finalmente ofrecer un servicio de mejor calidad con mayor seguridad y estabilidad de los datos

Una historia clínica es un instrumento médico legal que tiene una gran utilidad para los profesionales de la salud oral, ya que la misma representa la información que el médico mediante su trabajo obtuvo del paciente y esta a su vez se convierte en la mejor fuente de información que tiene un médico para evaluar a su paciente, razón por la cual es fundamental tener un sistema informático que brinden esta información de manera oportuna para una atención de calidad.

Para el presente proyecto se utilizará herramientas tecnológicas para el desarrollo del software como ser: Lenguaje PHP y JavaScript para la programación de los módulos, gestor de base de datos MySQL y la metodología XP para el modelado.

1.2. ANTECEDENTES

La clínica dental Patzi Emanuel ubicado en la zona Ballivián El Alto Av. Alfonso Ugarte N° 200, fundado por el Dr. Abraham Patzi (Implantólogo), brinda asistencia médica odontológica mediante una atención personalizada con tecnología de punta, con calidad y calidez. En cuanto a conceptos de salud bucal, gracias a los tratamientos con alta efectividad, hacen que el paciente obtenga una buena salud oral con servicios de: Odontología¹ General, Ortodoncia, Implantes Dentales, Prótesis fija/removible, Blanqueamiento dental, Periodoncia.

El proyecto titulado: "**Análisis diseño e implementación de un portal web para la gestión y administración hospitalaria de la clínica dental Barrera**", presentado por (Palacios, 2013), el cual tiene el objetivo de elevar los niveles de planificación, administración, calidad y promoción de los servicios, propone un sistema computarizado y una base de datos que permite a la clínica optimizar el manejo y control de información concerniente a la atención de pacientes y la administración hospitalaria. Para este proyecto se utilizó php para el desarrollo de procedimientos y MySQL para el diseño de base de datos. El proyecto tuvo una etapa de prueba de un mes en los cuales no se reportaron errores ni anomalías.

"**Sistema Web de Seguimiento a Historias Clínicas para la Empresa SPA Medico CIME basado en CRM**" por (Gonzales, 2014), el trabajo tiene por objetivo desarrollar un Sistema Web² de Seguimiento a Historias Clínicas para la empresa SPA Medico CIME, que permita, a través del modelado de negocio CRM fidelizar a clientes antiguos y ganar nuevos clientes a través del manejo de información en una constante actualización, y así mejorar el manejo de historiales clínicos dentro del SPA Medico CIME, y de esta manera fidelizar a los clientes. Para el desarrollo de

¹ La odontología es la ciencia de la salud que se centra en el diagnóstico, tratamiento y prevención de las enfermedades de la cavidad bucal. Además de los dientes, este sistema incluye las encías, el tejido periodontal, el maxilar superior, el maxilar inferior y la articulación temporomandibular.

² Web es un vocablo inglés que significa "red", "telaraña" o "malla". El concepto se utiliza en el ámbito tecnológico para nombrar a una red informática y, en general, a Internet.

este sistema se utilizó Php, Java Script, Ajax, Json y PostgreSql. Finalmente elaborar la documentación técnica y de usuario del sistema, utilizando la metodología de desarrollo Scrum.

“Sistema de Seguimiento y Control de Historiales Clínicos Clínica Bustillos”, (Hilaquita, 2005), Universidad Mayor de San Andrés, Carrera de Informática, el problema principal radica en la pérdida de tiempo en la búsqueda de historial clínico y el mal control de los análisis del paciente que es realizado manualmente en el cual se logró implementar los módulos de consultas, registros de pacientes clínico. Para tal efecto se utilizó la metodología Scrum y UML (Unified Modeling Language) para el modelado, para el desarrollo se implementó una base de datos SQL server 2003 y para el sistema con lenguajes Visual Basic .NET y ASP.NET.

“Implementación del sistema para el control de historia clínica de pacientes en centro odontológico Dental Group”.

Realizado por (Elizabeth Sanunga, 2017), un sistema que propone el desarrollo de un sistema computarizado y una base de datos que permite a la clínica dental optimizar el manejo y control de información concerniente a los pacientes. Para su desarrollo el proyecto utilizó Java, Ireport, Python, Xml, JavaScript y MySql para el modelado de la base de datos. Para el análisis y diseño se trabajó bajo la metodología RUP (Rational Unified Process) y UML (Unified Modeling Language).

“Software como servicio para la administración de historias clínicas”.

Realizado por (Yecid Yahuita, 2014), El trabajo pretende dar conocimiento y un empujón para la investigación y desarrollo de aplicaciones en Cloud computing, que se ve limitado en nuestro medio. El Software se desarrolló con PHP, JavaScript, Laravel y Sql Server. Para el análisis y diseño se utilizó la metodología Scrum.

1.3. PLANTEAMIENTO DEL PROBLEMA

1.3.1. Problema Principal

La clínica dental Patzi Emanuel actualmente realiza sus actividades de forma manual, el historial clínico³ es almacenado en carpetas, lo que ocasiona que la búsqueda del historial del paciente sea morosa, lo que genera pérdida de tiempo y un retardo significativo en la atención de los pacientes.

1.3.2. Problemas Secundarios

- El registro manual del paciente genera pérdida de tiempo y en ocasiones errores de inserción.
- No se tiene un adecuado control y organización de los archivos en la institución lo cual ocasiona demora en la consulta de información.
- Existencia de errores al momento de recolectar información.
- La búsqueda del paciente ocasiona retrasos en la atención, pues los historiales se encuentran archivados.
- El almacenamiento de historias clínica requiere un espacio adicional, debido a la utilización de gaveteros donde se guardan las historias dentro de sobres manila o folders.

³ La historia clínica es un documento imprescindible para guardar toda la información relativa a la salud de un paciente y los servicios y atención prestados al mismo.

1.4. OBJETIVOS

1.4.1. Objetivo General

Desarrollar un sistema dental web para mejorar la administración de consultas médicas dentales para la clínica Dental Patzi Emanuel, que permitan el acceso cómodo, rápido y seguro a la información por parte del personal de la clínica y así optimizar⁴ los tiempos de atención tanto a pacientes antiguos como nuevos.

1.4.2. Objetivo Específicos

- Analizar la situación actual de la clínica dental.
- Determinar los requerimientos y las necesidades relacionados con el sistema que se desarrollara.
- Diseñar una interfaz, que le permita al usuario manipular de manera eficiente⁵ la información que se genera en la clínica.
- Elaborar el manual de usuario del sistema propuesto, que sirva como referencia para el manejo y administración.
- Realizar el análisis de calidad y costos.

⁴ Conseguir que algo llegue a la situación óptima o dé los mejores resultados posibles.

⁵ La noción de eficiencia tiene su origen en el término latino *efficientia* y refiere a la habilidad de contar con algo o alguien para obtener un resultado. El concepto también suele ser equiparado con el de fortaleza o el de acción.

1.5. JUSTIFICACIÓN

1.5.1. Técnica

El presente trabajo se justifica técnicamente porque el proyecto utilizará tecnología informática avanzada como Base de datos MySQL⁶ de licencia publica general, PHP y JavaScript como herramientas de programación, metodología de desarrollo de software XP para un rápido avance.

1.5.2. Económica

Se justifica económicamente porque en el desarrollo del sistema planteado no tendrá pago de licencias de uso al ser desarrollado con herramientas de software libre.

Así la sistematización de las tareas reducirá costos relacionados en implementación de espacio y materiales de oficina.

1.5.3. Social

El sistema beneficia al consultorio dental, personal médico, pacientes. A través de automatización de seguimiento de historias clínicas, de tal manera se pueda tener una información actualizada, reduciendo en gran parte el trabajo manual que se realiza.

⁶ Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle

1.6. METODOLOGIA

Metodología XP (Extreme Programming)

La metodología XP o Programación Extrema es una metodología ágil y flexible utilizada para la gestión de proyectos.

Extreme Programming se centra en potenciar las relaciones interpersonales del equipo de desarrollo como clave del éxito mediante el trabajo en equipo, el aprendizaje continuo y el buen clima de trabajo.

Esta metodología pone el énfasis en la retroalimentación continua entre cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes. (calvo, 2018, www.diegocalvo.es)

Entre las principales ventajas esta:

- Programación organizada.
- Menor tasa de errores.
- Satisfacción del programador.

Fases de la programación extrema (XP)

Primera fase: Planificación del proyecto.

- Historias de usuarios
- Iteraciones
- Velocidad del proyecto.
- Reuniones diarias.

Segunda fase: Diseño.

- Diseños simples.
- Glosario de términos.
- Riesgos.
- Funcionalidad extra.

Tercera Fase: Codificación-desarrollo.

- Programación en parejas

Cuarta Fase: Pruebas.

- Evaluación del cliente.

Dentro del **ciclo⁷ de vida de XP** Se presenta las siguientes fases:

La primera fase es la de **exploración**, durante esta se planea a grandes rasgos las historias de usuarios, se elige y familiariza con la tecnología para el desarrollo.

La siguiente fase es la fase de **planeación**, en ella se prioriza las historias de usuario, y se acuerdan los alcances.

Otra fase es la de **producción**, requiere prueba y comprobación extra del funcionamiento del sistema antes de que este se pueda liberar al cliente, en esta fase, los nuevos cambios pueden todavía ser encontrados y debe tomarse la decisión de si se incluye o no en el reléase actual, durante esta fase las iteraciones pueden ser aceleradas de una a tres semanas.

Otra fase es la de **mantenimiento**, requiere un mayor esfuerzo para satisfacer también las tareas del cliente, la fase de mantenimiento puede requerir la incorporación de nueva gente y cambiar la estructura del equipo.

⁷ Serie de fases o estados por las que pasa un acontecimiento o fenómeno y que se suceden en el mismo orden hasta llegar a una fase o estado a partir de los cuales vuelven a repetirse en el mismo orden

Finalmente, la fase **de muerte**, cuando el cliente no tiene más historias para incluirlas en el sistema, se genera la documentación final del sistema y no se realiza más cambios en la arquitectura del sistema. (Tobón y Carmona, 2007, p.32)

1.7. HERRAMIENTAS

Framework para el desarrollo web:

- **BOOTSTRAP**

Bootstrap, es un framework⁸ originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo. (Solis, 2014, www.arweb.com)

- **CODEIGNITER**

Es un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. «Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas. (Pineda, 2016, www.coriaweb.hosting)

⁸ Un entorno de trabajo, o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

PHP

Acrónimo recursivo en inglés de PHP: Hypertext Preprocessor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el preprocesado de texto plano en UTF-8. Posteriormente se aplicó al desarrollo web de contenido dinámico, dando un paso evolutivo en el concepto de aplicación en línea, por su carácter de servicio. (Danilo, 2013, es.slideshare.net)

JAVASCRIPT

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. (www.arimetrics.com)

MySQL

MySQL es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL). (searchDatacenter).

1.8. LIMITES Y ALCANCES

1.8.1. Limites

El sistema será desarrollado solamente para cumplir con el objetivo de realizar el registro de pacientes, tratamiento y pagos del paciente.

1.8.2. Alcances

Para desarrollar el proyecto es necesario realizar un estudio profundo de los procesos técnicos, resolviendo las dificultades y deficiencias más importantes.

Los alcances del sistema para la clínica dental Patzi Emanuel se definen a través de tres módulos que se encargan de la gestión de información específica, necesaria para ampliar, establecer un vínculo entre el paciente y el consultorio dental, descrita a continuación:

- **Módulo de autenticación:** El usuario que ingresará al sistema deberá identificarse mediante un nombre de usuario y una contraseña.
- **Modulo de registro de usuarios:** Dentro de este módulo se ingresa los datos del usuario.
- **Módulo de registros de pacientes:** Dentro de este módulo se ingresará los datos personales del paciente.
- **Módulo de modificación:** En este módulo se modificará los datos que sean dinámicos.
- **Modulo pagos:** Mostrara los detalles del tratamiento que se le realiza a los pacientes.
- **Módulo de reportes:** Dentro de la cual se mostrará la cantidad de pacientes atendidas por mes.

1.9. APORTES

El Sistema propuesto tendrá un aporte principalmente para el medico dentista, puesto que ayudará a un manejo ordenado del registro de pacientes, así mismo tendrá un aporte tecnológico ya que servirá de antecedente para futuros proyectos abocados a esta temática.

CAPITULO II

MARCO TEORICO

2. MARCO TEORICO

2.1. INTRODUCCION

En este capítulo se pretende detallar los antecedentes o marco referencial⁹ y las consideraciones teóricas del tema de investigación.

La teoría constituye la base donde se sustentará cualquier análisis o propuesta de desarrollo. Al ser un sistema que se desarrollara con una metodología ágil denominada XP (Programación Extrema), se debe entender la estructura que tiene la misma, el cual será empleado para la solución de problemas.

2.2. SISTEMA

El concepto de sistema es muy general. Podemos definir un sistema como un conjunto de elementos estructurados e interconectados de una manera definida. En otras palabras, es un conjunto de elementos asociados a un conjunto de relaciones entre elementos. Entre otros conceptos se encuentran los siguientes:

- “El sistema es un conjunto de elementos organizados que se encuentran integrados, que buscan alguna meta o metas comunes, operando para ello sobre datos o información sobre energía o materia u organismos en una referencia temporal para producir como salida información o energía o materia y organismos”.
- “Un sistema de control estudia la conducta del sistema con el fin de regularla

⁹ Es un texto que identifica y expone los antecedentes, las teorías, las regulaciones y/o los lineamientos de un proyecto de investigación, de un programa de acción o de un proceso. Referencia a obras capitales de otros investigadores sobre el tema.

- de un modo conveniente para su supervivencia. Una de sus características es que sus elementos deben ser lo suficientemente sensitivos y rápidos como para satisfacer los requerimientos para cada función del control”.
- “Un sistema es un conjunto organizado de cosas o partes interactivas e interdependientes, que se relacionan formando un todo unitario y complejo”. (Murdick & Munson, 1988).

2.3. INTRANET

es una red informática interna dentro de una organización que utiliza el protocolo TCP/IP¹⁰, tecnologías avanzadas de publicación electrónica WEB (http), correo electrónico(email), archivos y recursos compartidos (file server, printer server, backup server, ftp server), acceso remoto (vpn, ssh), y otros. brindando así todo el potencial del esquema cliente-servidor en un ámbito acotado, segurizado y racionalizado de acuerdo a los objetivos de eficacia y eficiencia estimados en dicha organización. (Mogetta, Audisio, & Casco, 2014)

Al tratarse de una red interna independiente de Internet donde solo acceden los empleados, la intranet puede funcionar aunque no se conecte a Internet. Son redes independientes y diferentes.

¹⁰ es un conjunto de protocolos que permiten la comunicación entre los ordenadores pertenecientes a una red. La sigla TCP/IP significa Protocolo de control de transmisión/Protocolo de Internet y se pronuncia "T-C-P-I-P". Proviene de los nombres de dos protocolos importantes incluidos en el conjunto TCP/IP, es decir, del protocolo TCP y del protocolo IP.

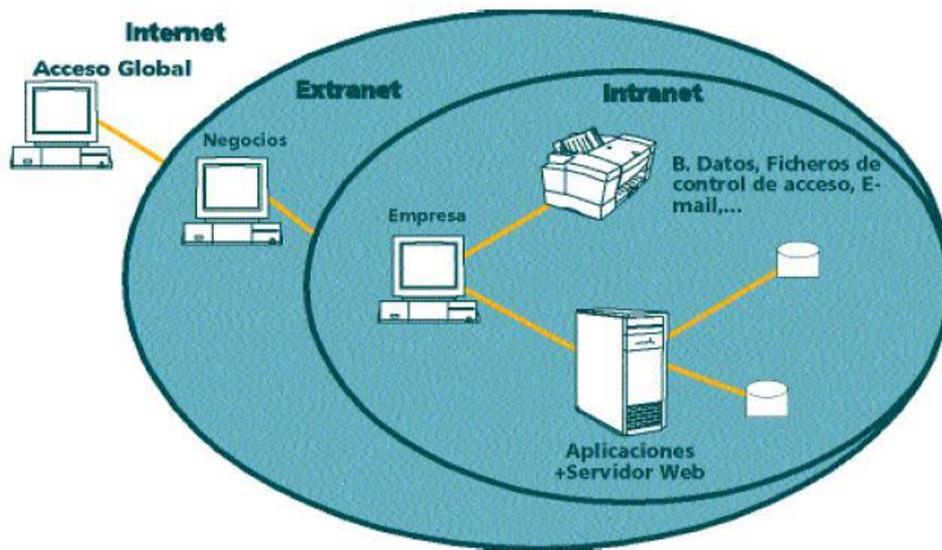


Figura 1. Internet – Extranet – Intranet
 Fuente: (Mogetta, Audisio, & Casco, 2014)

En la siguiente tabla se muestran de manera resumida las diferencias entre las aplicaciones en una empresa:

Tabla 1- Comparación de Internet, Intranet y Extranet

	Internet	Intranet	Extranet
Acceso	Publico	Privado	Semi – Publico
Usuarios	Cualquiera	Miembros de una Compañía	Grupo de Empresas estrechamente relacionadas
Información	Fragmentada	propietaria	Compartida dentro de un círculo de empresas

Fuente: (Mogetta, Audisio, & Casco, 2014)

2.3.1. Elementos de la Intranet

- **SEGURIDAD:** define el ámbito y normativa para asegurarlo de virus, violación de acceso, fuga de información, y otros.

- **INTEGRIDAD:** protege los cambios o eliminación de datos no queridos.
- **AUTENTICACIÓN:** control de ingreso a la red de acuerdo al estándar de verificación de identidades.
- **AUDITORÍA:** control de la actividad en tiempo real y posterior, con posibilidad de retroalimentación del sistema (autoaprendizaje).
- **DISPONIBILIDAD Y OPORTUNIDAD:** asegura la presencia de los servicios y datos requeridos en tiempo y forma predefinidos.

2.3.2. ¿Cómo funciona una Intranet?

En la Intranet de una compañía, el acceso a la información es muy similar al acceso a Internet. Ello supone que, si conocemos el proceso para Internet, será casi automático empezar a utilizar una Intranet (con el ahorro en formación que eso supone para la empresa).

Se puede describir el proceso de la siguiente manera:

- Todas las máquinas cliente (PCs locales conectados a una máquina Servidor de Web), utilizan un software especial, un browser¹¹ (por ejemplo, Netscape Navigator o Internet Explorer), además de software especial para el correo electrónico, FTP y demás servicios.
- El usuario hace click en un hipervínculo¹², que envía una petición de información al servidor. El hipervínculo contiene el nombre del servidor donde

¹¹ Es un navegador de Internet: un software que permite la visualización de los contenidos que presenta una página web. Se trata de un navegador que apuesta por la estabilidad, la velocidad y la seguridad.

¹² Es un enlace, normalmente entre dos páginas web de un mismo sitio, pero un enlace también puede apuntar a una página de otro sitio web, a un fichero, a una imagen, etc. Para navegar al destino al que apunta el enlace, hemos de hacer clic sobre él

está localizada la información, y el nombre del fichero que ha de ser cargado en el browser.

- El servidor lee la petición, vuelca el fichero, y lo manda al cliente.
- El cliente recibe el fichero y lo deposita en la ventana del browser (siempre que dicho fichero sea compatible con el web, por ejemplo, con formato HTML y otros). Puede ocurrir también que el fichero se deposite en discos o en aplicaciones especiales.

2.3.3. Ventajas y Desventajas de la intranet

Ventajas:

- Bajos Costos, tanto como de integración y mantenimiento.
- Es una forma muy eficiente y económica de distribuir datos.
- Fácil manejo y configuración a la infraestructura tecnológica de la organización, se encuentra disponible en todas las plataformas informáticas.
- Es adaptable a las necesidades, tanto como de empresa, departamento, área de negocio, y otros.
- Sencilla integración de multimedia.
- Acceso a internet, tanto al exterior, como al interior, por parte de usuarios registrados con control de acceso.

Desventajas

- Muchas Intranet bien construidas tecnológicamente terminan en fracaso. Con frecuencia vemos información desactualizada, mal gestionada por qué no se ha elegido el vehículo apropiado o porque los boletines están inactivos.
- Los aspectos de seguridad son muy importantes. Las intranets son redes expuestas a notables riesgos de seguridad internos.

2.4. SISTEMA WEB

Se denomina sistema web a aquellas aplicaciones de software que puede utilizarse accediendo a un servidor web a través de Internet o de una intranet mediante un navegador.

Las aplicaciones web son muy usadas hoy en día, debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo. (www.aeurus.cl, 2016)

2.4.1. Ventajas y Desventajas de un Sistema Web

Ventajas:

- **Ahorran costes de hardware y software**, sólo es necesario usar un ordenador con un navegador web y conectarse a Internet. Las aplicaciones basadas en web usan menos recursos que los programas instalados. Incluso puede usar un equipo de bajos recursos.
- **Fáciles de usar**, las aplicaciones web son muy sencillas de utilizar, sólo necesitará conocimientos básicos de informática para trabajar con ellas. Si sabe escribir un correo electrónico, ya sabe usarlas.

- **Provocan menos errores y problemas**, las aplicaciones web son menos propensas a colgarse y crear problemas técnicos debido a conflictos con hardware, con otras aplicaciones existentes, protocolos o con software personal interno.

Desventajas:

- **El desarrollo tarda tiempo**, crear un sistema web a la medida implica un proceso de análisis, planificación, diseño, desarrollo e implementación que puede tardar varios meses dependiendo de la complejidad de la aplicación.
- **Mantenimiento**, Sólo en algunos casos, la empresa deberá asumir la responsabilidad del mantenimiento de la aplicación, cuya periodicidad dependerá de la complejidad del software y de la calidad de la codificación.

2.5. INGENIERIA DE SOFTWARE

La evolución de la disciplina de la ingeniería de software ha traído consigo propuestas diferentes para mejorar los resultados del proceso de construcción. Las metodologías tradicionales haciendo énfasis en la planeación, mientras las metodologías ágiles haciendo énfasis en la adaptabilidad del proceso, delinean las principales propuestas presentes en la literatura. De manera paralela, el tema de modelos para el mejoramiento de los procesos de desarrollo ocupa un lugar importante en la búsqueda de la metodología adecuada para producir software de calidad en cualquier contexto de desarrollo. De una u otra forma, las características de los proyectos de software hacen necesario seguir prácticas específicas para optimizar los resultados de los desarrollos. (Arboleda Jiménez, s.f.)

2.5.1. MODELOS DE DESARROLLO DE SOFTWARE

2.5.1.1. Metodologías tradicionales

Las metodologías tradicionales se caracterizan por exponer procesos basados en planeación exhaustiva. Esta planeación se realiza esperando que el resultado de cada proceso sea determinístico y predecible. La experiencia ha mostrado que, como consecuencia de las características del software, los resultados de los procesos no son siempre predecibles y sobre todo es difícil predecir desde el comienzo del proyecto cada resultado. Sin embargo, es posible por medio de la recolección y estudio de métricas de desarrollo lograr realizar estimaciones acertadas en contextos de desarrollo repetibles.

2.5.1.2. Metodologías Ágiles

Grupos de desarrollo han experimentado soluciones que basan su fundamento en la adaptabilidad de los procesos de desarrollo, en lugar de seguir esperando lograr resultados predecibles de un proceso que no evoluciona. Esta comunidad de desarrolladores e investigadores han nombrado su trabajo bajo lo que conocemos como metodologías ágiles. Las metodologías ágiles como puede entenderse mal, no están en contra de administrar procesos de desarrollo. Por el contrario, promueve la formalización de procesos adaptables.

La compilación de los principios y valores que resaltan las metodologías ágiles fue formalizada en el manifiesto para el desarrollo de software ágil. Este documento desarrollado por los representantes de cada una de las metodologías que en el momento se presentaban como ágiles, logra resumir en un conjunto de ideas las prácticas que una metodología de este estilo debe llevar a cabo. Como característica fundamental, la habilidad de responder al cambio es la principal característica de las metodologías ágiles.

XP, una de las más difundidas, es una metodología de desarrollo de software ágil que define pocas reglas y pocas prácticas. XP promueve la adaptabilidad de los procesos de desarrollo basándose en los principios y prácticas que presenta. (Arboleda Jiménez, s.f.).

Tabla 2 - Diferencias entre metodologías ágiles y no ágiles

Metodologías Ágiles	Metodologías tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo de desarrollo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Mas artefactos
Pocos roles	Mas roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Fuente: (Letelier & Penadés, 2006)

2.6. METODOLOGIA DE DESARROLLO XP

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el

equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Letelier & Penadés, 2006)

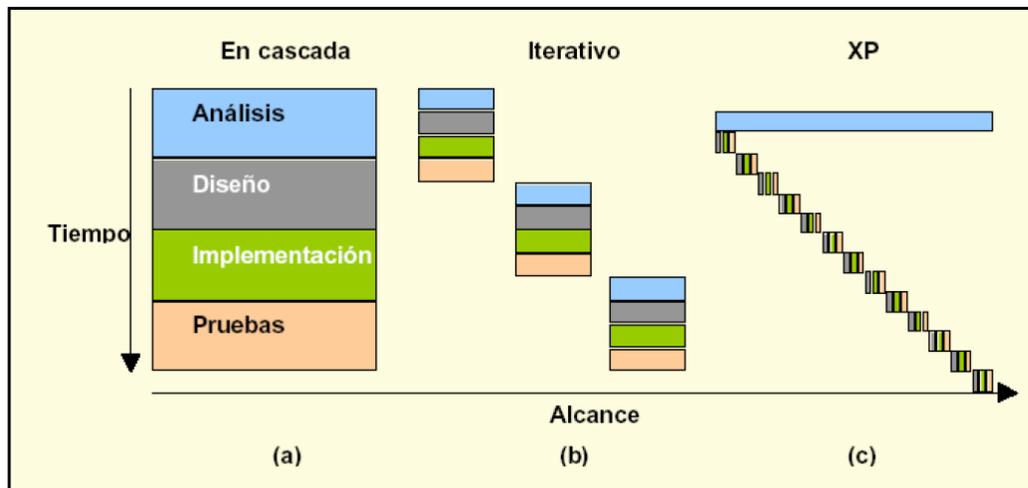


Figura 2. Evolución de los largos ciclos de desarrollo en cascada (a) a ciclos iterativos más cortos (b) y a la mezcla que hace XP.

Fuente: (Borja López, 2017)

XP es una de las llamadas metodologías ágiles de desarrollo de software más exitosas de los tiempos recientes. La metodología propuesta en XP está diseñada para entregar el software que los clientes necesitan en el momento en que lo necesitan. XP alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo.

La metodología también enfatiza el trabajo en equipo. Tanto gerentes como clientes y desarrolladores son partes del mismo equipo dedicado a entregar software de calidad. (Joskowicz, 2008)

Para alcanzar el objetivo de software como solución ágil, la metodología XP se estructura en tres capas que agrupan las doce prácticas básicas de XP:

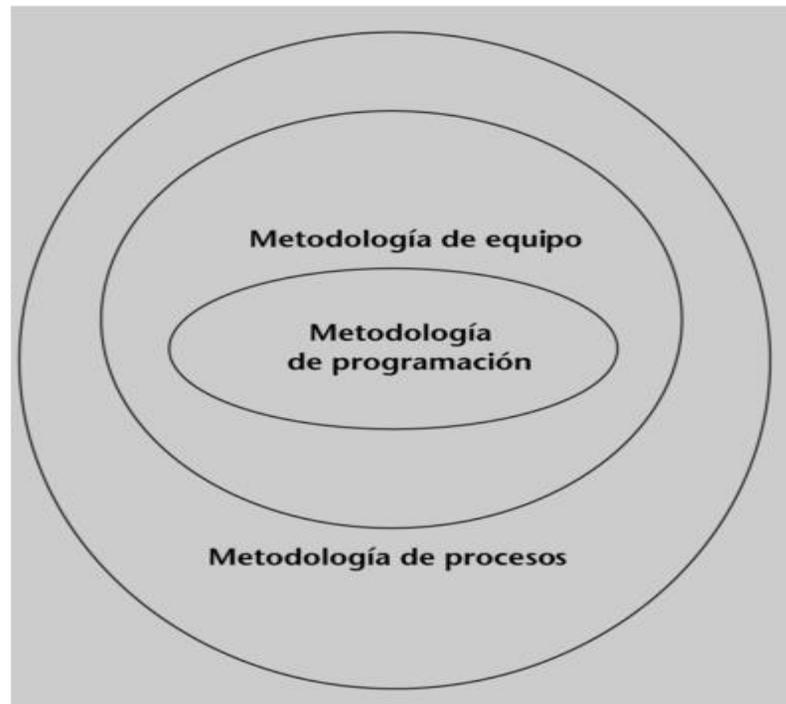


Figura 3 Extreme Programming (XP)

Fuente: (Fernández, 2011)

- **Metodología de programación:** diseño sencillo, test, refactorización y codificación con estándares.
- **Metodología de equipo:** propiedad colectiva del código, programación en parejas, integración continua, cuarenta horas semanales y metáfora del negocio.
- **Metodología de procesos:** cliente in situ, entregas frecuentes y planificación del juego. (Fernández, 2011).

2.6.1. Valores de XP

Los creadores de esta metodología quisieron medir su utilidad a través de cuatro valores, que representan aquellos aspectos cuyo cumplimiento nos va a garantizar el éxito en el proyecto:

- **Comunicación.** Debe ser fluida entre todos los participantes en el proyecto; además el entorno tiene que favorecer la comunicación espontánea¹³, ubicando a todos los miembros en un mismo lugar. La comunicación directa nos da mucho más valor que la escrita, podemos observar los gestos del cliente, o la expresión de cansancio de nuestro compañero.
- **Simplicidad.** Cuanto más sencilla sea la solución, más fácilmente podremos adaptarla a los cambios. Las complejidades aumentan el coste del cambio y disminuyen la calidad del software. En XP nos vamos a olvidar de frases como "haremos un sistema genérico que...", o "esto lo pongo por si acaso algún día lo necesitamos".
- **Realimentación.** El usuario debe utilizar desde la primera entrega el software desarrollado, dándonos sus impresiones y sus necesidades no satisfechas, de manera que esas historias vuelvan a formar parte de los requisitos del sistema.
- **Coraje.** Coraje para vencer la frase más típica de los desarrolladores: "si funciona no lo toques". Con XP debemos tocar continuamente cosas que ya funcionan, para mejorarlas. Hemos de cambiar esta frase por la de: "si funciona, puedes mejorarlo". Y eso, os lo aseguramos, requiere de mucho valor y coraje. (Fernández, 2011)

2.6.2. Los diferentes roles dentro de XP

Según Fernández (2011) Cada rol tiene unas funciones claras dentro de la metodología XP. Cada persona del equipo puede ejecutar uno o varios roles, o incluso cambiar de rol durante las diferentes fases del proyecto.

¹³ Es muy usada en nuestro idioma para expresar aquello que se realiza de manera voluntaria, es decir, que no fue instado por la fuerza, por agentes exteriores, o por una orden, para concretarse.

Son muchas las extensiones que se han hecho de los roles de la propuesta original de Beck, pero los roles que permanecen siempre en cualquier implementación de la metodología XP son los siguientes:

Programador:

- Escribe las pruebas unitarias.
- Produce el código del programa.

Cliente:

- Escribe las historias de usuario.
- Diseña las pruebas de aceptación.
- Prioriza las historias de usuario.
- Aporta la dimensión de negocio al equipo de desarrollo.
- Representa al colectivo de usuarios finales.
- Esta siempre disponible para las consultas.

Encargado de pruebas (tester):

- Ayuda al cliente a diseñar pruebas de aceptación.
- Ejecuta las pruebas de aceptación.
- Ejecuta las pruebas de integración.
- Difunde los resultados entre el equipo de desarrollo y el cliente.
- Es el responsable de las herramientas automatizadas de las pruebas.

Encargado de seguimiento(tracker):

- Se encarga de realimentar todo el proceso de XP, midiendo las desviaciones.
- Con respecto a las estimaciones y comunicando los resultados para mejorar las siguientes estimaciones.
- Realiza el seguimiento de cada iteración del proceso de XP tanto en la etapa de iteraciones como en la producción.
- Revalúa la posibilidad de incorporar o eliminar historias de usuario.

Entrenador (coach):

- Se encarga del proceso global.
- Garantiza la filosofía de XP.
- Conoce al fondo la metodología.
- Provee guías y ayudas a los miembros del equipo a la hora aplicar las practicas básicas de XP.

Consultor:

- No forma parte del equipo.
- Tiene un conocimiento específico de un área en concreto.
- Ayuda a resolver un problema puntual, ya sea de spike tecnológico o de valor y negocio.

Gestor (boss):

- Es el máximo responsable del proyecto.
- Hace de enlace con los clientes.
- Se encarga de coordinar y de garantizar las condiciones necesarias para el desarrollo del trabajo.

2.6.3. Fases de la metodología XP

La programación extrema engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales:

La figura 5 ilustra el proceso XP y resalta algunas ideas y tareas clave que se asocia con cada actividad estructural.

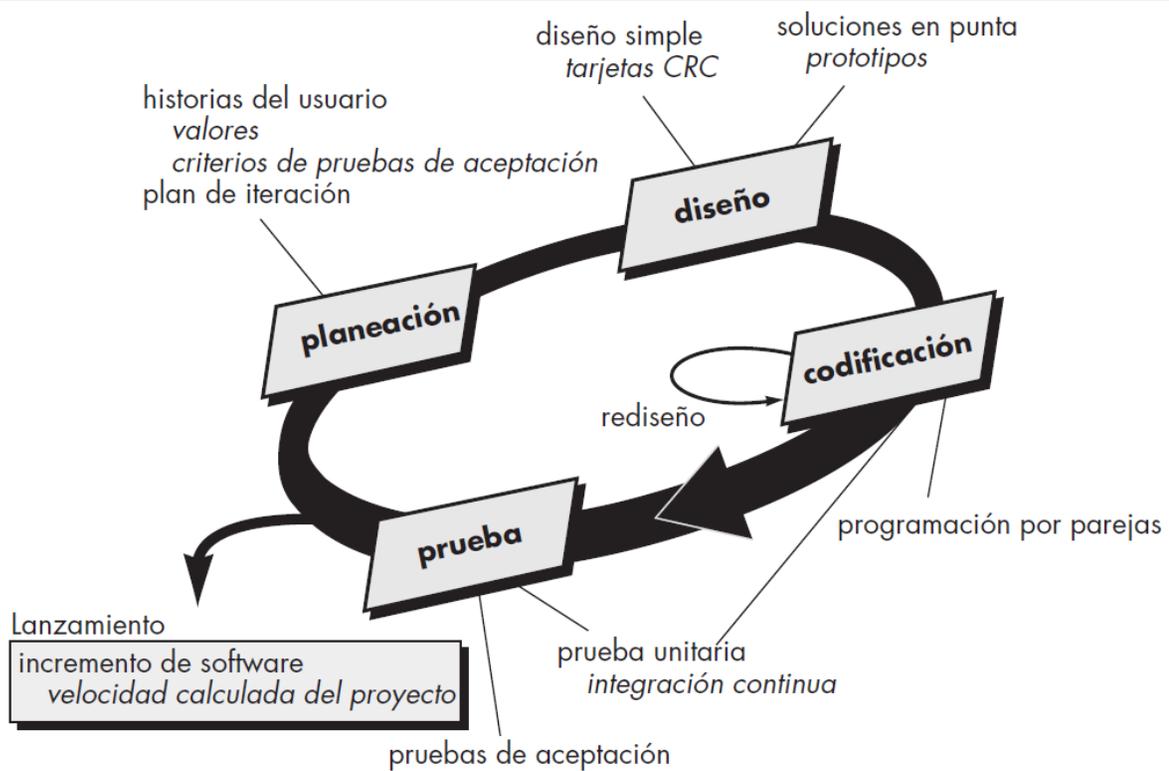


Figura 4 el proceso de la programación extrema

Fuente: (Pressman, 2010, p.62)

La metodología XP tiene un conjunto importante de reglas y prácticas. En forma genérica, se pueden agrupar en:

Planificación:

La Metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores. El proyecto comienza recopilando las historias de usuarios, las que constituyen a los tradicionales casos de uso. Una vez obtenidas estas historias de usuarios, los programadores evalúan rápidamente el tiempo de desarrollo de cada una.

Los Conceptos básicos de la planificación son:

- **Las Historias de Usuarios**, las cuales son descritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software.
- **El Plan de Entregas (Release Plan)**, establece que las historias de usuarios serán agrupadas para conformar una entrega y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto.
- **Plan de Iteraciones (Iteration Plan)**, las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido.
- **Reuniones Diarias de Seguimiento (Stand – Up Meeting)**, el objetivo es mantener la comunicación entre el equipo y compartir problemas y soluciones.

Diseño:

La Metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes: Simplicidad, Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione.

- **Soluciones “Spike”**, Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba (llamados “Spike”), para explorar diferentes soluciones.

- **Recodificación (“Refactoring”)**, Consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de crearlo más simple, conciso y entendible. Las metodologías de XP sugieren re codificar cada vez que sea necesario.
- **Tarjetas CRC**, la utilización de tarjetas CRC (Class-Responsibility-Collaboration) es una técnica de diseño orientado a objetos propuesta por Kent Beck (introducido de la metodología de programación extrema) y Ward Cunningham. El objetivo de la misma es hacer, mediante tarjetas, un inventario de las clases que vamos a necesitar para implementar el sistema y la forma en que van a interactuar, de esta forma se pretende facilitar el análisis y discusión de las mismas por parte de varios actores del equipo de proyecto con el objeto de que el diseño sea lo más simple posible verificando las especificaciones del sistema.

Desarrollo:

- **Disponibilidad del Cliente**, Uno de los requerimientos de XP es tener al cliente disponible durante todo el proyecto. No solamente como apoyo a los desarrolladores, sino formando parte del grupo. El involucramiento del cliente es fundamental para que pueda desarrollarse un proyecto con la metodología XP.
- **Programación en Pares**, XP propone que se desarrolle en pares de programadores, ambos trabajando juntos en un mismo ordenador. Si bien parece que ésta práctica duplica el tiempo asignado al proyecto (y por ende, los costos en recursos humanos), al trabajar en pares se minimizan los errores y se logran mejores diseños, compensando la inversión en horas. El producto obtenido es por lo general de mejor calidad que cuando el desarrollo se realiza por programadores individuales.

- **Integraciones Permanentes**, Todos los desarrolladores necesitan trabajar siempre con la “última versión”. Realizar cambios o mejoras sobre versiones antiguas causan graves problemas, y retrasan al proyecto. Es por eso que XP promueve publicar lo antes posible las nuevas versiones, aunque no sean las últimas, siempre que estén libres de errores. Idealmente, todos los días deben existir nuevas versiones publicadas. Para evitar errores, solo una pareja de desarrolladores puede integrar su código a la vez.
- **Propiedad Colectiva del Código**, En un proyecto XP, todo el equipo puede contribuir con nuevas ideas que apliquen a cualquier parte del proyecto. Asimismo, una pareja de programadores puede cambiar el código que sea necesario para corregir problemas, agregar funciones o re codificar.
- **Ritmo Sostenido**, La Metodología XP indica que debe llevarse un ritmo sostenido de trabajo. Anteriormente, ésta práctica se denominaba “Semana de 40 horas”. El concepto que se desea establecer con esta práctica es planificar el trabajo de forma a mantener un ritmo constante y razonable, sin sobrecargar al equipo.

Pruebas:

- **Pruebas Unitarias**, Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código (“Test-Driven Programming”). Que todo código liberado pase correctamente las pruebas unitarias, es lo que habilita que funcione la propiedad colectiva del código.

- Pruebas de Aceptación, Son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El Cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta que pase correctamente todas las pruebas de aceptación. (Borja López, 2017).

2.7. HERRAMIENTAS DE DESARROLLO

2.7.1. Gestor de base de datos MySQL

El software MySQL proporciona un servidor de base de datos SQL (Structured Query Language) veloz, multi-hilo, multiusuario y robusto. El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

El servidor está proyectado tanto para sistemas críticos en producción soportando intensas cargas de trabajo como para empotrarse en sistemas de desarrollo masivo de software. El software de bases de datos MySQL consiste de un sistema cliente/servidor que se compone de un servidor SQL multihilo, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación (APIs). Se puede obtener también como una biblioteca multihilo que se puede enlazar dentro de otras aplicaciones para obtener un producto más pequeño, más rápido, y más fácil de manejar.

MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos. El software MySQL tiene licencia dual, pudiéndose usar de

forma gratuita bajo licencia GNU o bien adquiriendo licencias comerciales de MySQL AB en el caso de no desear estar sujeto a los términos de la licencia GPL.

MySQL es un sistema de administración de bases de datos relacionales, una base de datos relacional almacena los datos en tablas separadas en lugar de poner todos los datos en un solo lugar. Esto agrega velocidad y flexibilidad. Las tablas son enlazadas al definir relaciones que hacen posible combinar datos de varias tablas cuando se necesitan consultar datos. La parte SQL de "MySQL" significa "Lenguaje Estructurado de Consulta", y es el lenguaje más usado y estandarizado para acceder a bases de datos relacionales (Eduardo, 2007).

2.7.2. Servidor Apache

Apache es un software de servidor web gratuito y de código abierto con el cual se ejecutan el 46% de los sitios web de todo el mundo. El nombre oficial es Apache HTTP Server, y es mantenido y desarrollado por la Apache Software Foundation.

Le permite a los propietarios de sitios web servir contenido en la web, de ahí el nombre de "servidor web". Es uno de los servidores web más antiguos y confiables, con la primera versión lanzada hace más de 20 años, en 1995.

Cuando alguien quiere visitar un sitio web, ingresa un nombre de dominio en la barra de direcciones de su navegador. Luego, el servidor web envía los archivos solicitados actuando como un repartidor virtual.(hostinger.es, 2019)

- **Funcionamiento**

Aunque llamamos a Apache un servidor web, no es un servidor físico, sino un software que se ejecuta en un servidor. Su trabajo es establecer una conexión entre un servidor y los navegadores de los visitantes del sitio web (Firefox, Google Chrome, Safari, y otros) mientras envían archivos entre ellos

(estructura cliente-servidor). Apache es un software multiplataforma, por lo cual funciona tanto en servidores Unix como en Windows.

Cuando un visitante quiere cargar una página de tu sitio web, por ejemplo, la página de inicio o tu página “Acerca de nosotros”, su navegador le envía una solicitud a tu servidor y Apache le devuelve una respuesta con todos los archivos solicitados (texto, imágenes, y otros) El servidor y el cliente se comunican a través del protocolo HTTP y Apache es responsable de garantizar una comunicación fluida y segura entre las dos máquinas.

Apache es altamente personalizable, ya que tiene una estructura basada en módulos. Los módulos le permiten a los administradores del servidor activar y desactivar funcionalidades adicionales. Apache tiene módulos de seguridad, almacenamiento en caché, reescritura de URL, autenticación de contraseña y más. También puedes ajustar tus propias configuraciones del servidor a través de un archivo llamado htaccess. (Hostinger.es, 2019)

2.7.3. Lenguaje de programación

2.7.3.1. PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. PHP es un lenguaje de script incrustado dentro del HTML, la mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente

recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores (Alvarez, 2001).

Se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, y otros.

Al utilizar una tecnología del tipo pre-procesado en el servidor es necesario visualizar las páginas generadas con PHP utilizando el protocolo http. Al contrario de lo que ocurre con las páginas de la tecnología cliente, en las que se puede visualizar mediante la opción "Archivo>Abrir" en cualquier navegador, las páginas generadas con PHP necesitan ser servidas por un servidor web para que sean procesadas y luego enviadas al navegador del usuario.

Permite la conexión a todo tipo de servidores de base de datos como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. PHP es una alternativa a otros sistemas como el ASP.NET / C# / VB.NET de Microsoft o a ColdFusion de Macromedia, a JSP/Java de Sun Microsystems, y a CGI/Perl. La ventaja con los de Microsoft o Macromedia es que es totalmente gratuito, no hay que pagar licencias. Una aplicación web basada en PHP necesita dos tipos de software. El primero es un servidor web que va a atender las peticiones de los usuarios y devolverá las páginas solicitadas. El servidor Apache, tanto su versión Windows como Linux es el más utilizado (php.net, 2009).

2.7.3.2. JAVASCRIPT

JavaScript es un lenguaje interpretado orientado a las páginas web, con una sintaxis semejante a la del lenguaje Java. El lenguaje fue inventado por Brendan Eich en la

empresa Netscape Communications, que es la que fabricó los primeros navegadores de Internet comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0. Se utiliza en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación cliente.

JavaScript por su sencillez sigue siendo una herramienta muy útil en la elaboración de páginas web que tengan algo más que texto. Microsoft desarrolló el JScript para su Explorer que, en términos generales, es compatible con Netscape.

JavaScript es un lenguaje de scripting multiplataforma y orientado a objetos. Es un lenguaje pequeño y liviano. Dentro de un ambiente de host, JavaScript puede conectarse a los objetos de su ambiente y proporcionar control programático sobre ellos.

JavaScript contiene una librería estándar de objetos, tales como Array, Date, y Math, y un conjunto central de elementos del lenguaje, tales como operadores, estructuras de control, y sentencias (RamiroNeher, 2015).

2.7.4. Framework

2.7.4.1. Bootstrap

Bootstrap es un framework CSS desarrollado inicialmente (en el año 2011) por Twitter que permite dar forma a un sitio web mediante librerías CSS que incluyen tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web. Aunque el desarrollo del framework Bootstrap fue iniciado por Twitter, fue liberado bajo licencia MIT en el año 2011 y su desarrollo continuo en un repositorio de GitHub.

Bootstrap es una excelente herramienta para crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su

tamaño. Además, Bootstrap ofrece las herramientas necesarias para crear cualquier tipo de sitio web utilizando los estilos y elementos de sus librerías.

Desde la aparición de Bootstrap 4 el framework se ha vuelto bastante más compatible con desarrollo web responsive, entre otras características se han reforzado el soporte es bastante bueno (casi completo) con HTML5 y CSS3, permitiendo ser usado de forma muy flexible para desarrollo web con unos excelentes resultados, se ha añadido un sistema GRID que permite diseñar usando un GRID de 12 columnas donde se debe plasmar el contenido, con esto podemos desarrollar responsive de forma mucho más fácil e intuitiva también Bootstrap 3 establece Media Queries para 4 tamaños de dispositivos diferentes variando dependiendo del tamaño de su pantalla, estas Media Queries permiten desarrollar para dispositivos móviles y tablets de forma mucho más fácil y por último Bootstrap 4 permite insertar imágenes responsive, es decir, con solo insertar la imagen con la clase “img-responsive” las imágenes se adaptaran al tamaño (Jareño, 2017)

2.7.5. Codeigniter

Es un entorno de desarrollo web escrito en PHP que presume de acelerar y optimizar el desarrollo de aplicaciones web gracias a un compacto diseño de software. La compañía de software norteamericana EllisLab fue la encargada de su creación y de la publicación de su primera versión en febrero de 2006. Un año después de anunciar, el 9 de julio de 2013, que la compañía ya no disponía de los recursos necesarios para continuar desarrollando el software, utiliza el MVC. Esto permite a los programadores o desarrolladores Web mejorar su forma de trabajar, además de dar una mayor velocidad a la hora de crear páginas Webs.

2.7.5.1. Patrón modelo vista controlador

El MVC es un patrón de diseño arquitectónico de software, que sirve para clasificar la información, la lógica del sistema y la interfaz que se le presenta al usuario. En este tipo de arquitectura existe un sistema central o controlador que gestiona las

entradas y la salida del sistema, uno o varios modelos que se encargan de buscar los datos e información necesaria y una interfaz que muestra los resultados al usuario final. Es muy usado en el desarrollo web porque al tener que interactuar varios lenguajes para crear un sitio es muy fácil generar confusión entre cada componente si estos no son separados de la forma adecuada. Este patrón permite modificar cada uno de sus componentes si necesidad de afectar a los demás. (codingornot.com)

- **El modelo** es donde se guarda toda la lógica del negocio de una aplicación, la lógica del negocio puede ser cualquier cosa específica acerca de cómo una aplicación almacena los datos, o utiliza servicios de terceros con el fin de cumplir con sus necesidades. Si la aplicación debe acceder a la información en una base de datos, el código deberá estar guardado en el modelo.
- **La vista** es donde se encontrarán todos los elementos de la interfaz de usuario de una aplicación, esta puede contener código HTML, hojas de estilo CSS y archivos Javascript. Cualquier cosa que el usuario pueda ver, es guardada en la vista, y algunas veces lo que ve el usuario actualmente es la combinación de varias vistas en la misma petición.
- **El controlador** es el componente encargado de conectar el modelo con la vista. Los controladores aíslan la lógica del negocio de un modelo de los elementos de la interfaz de usuario de una vista y maneja la forma en la que la aplicación responde a la interacción del usuario en la vista. Los controladores son el primer punto de entrada en estos componentes, ya que la primera solicitud se pasa a un controlador, que luego instancia a los modelos y vistas requeridas para cumplir con una petición a la aplicación. (Gómez R. , Modelo Vista Controlador, 2017).

2.7.5.2. Ciclo de vida del MVC

- El usuario realiza una petición.
- El controlador captura la petición del usuario.
- El controlador llama al modelo.
- El modelo interactúa con la base de datos, y retorna la información al controlador.
- El controlador recibe la información y la envía a la vista.
- La vista procesa la información recibida y la entrega de una manera visualmente entendible al usuario.

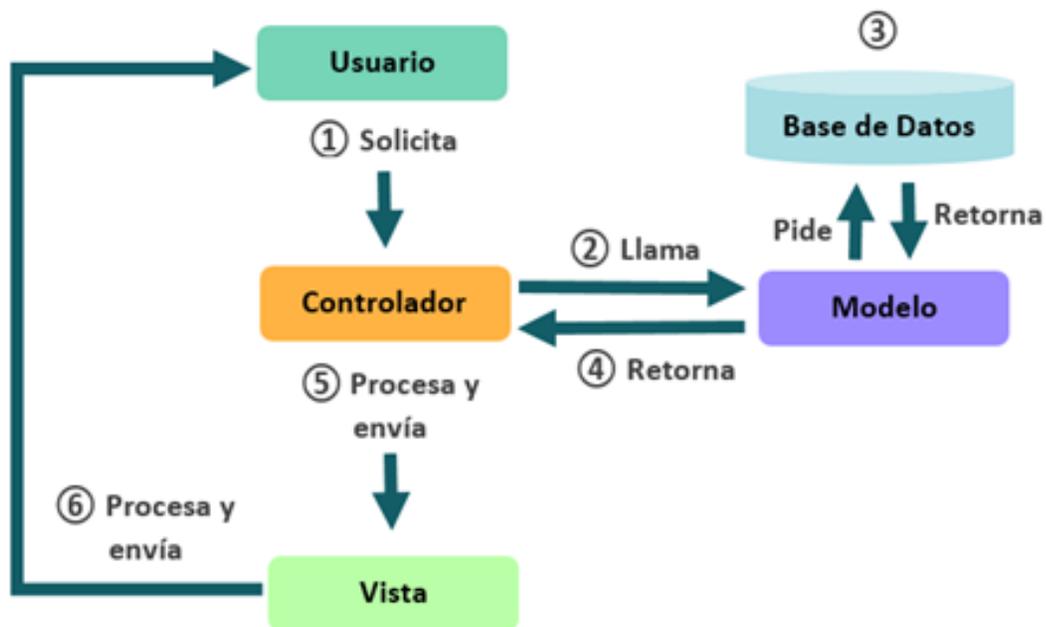


Figura 5. Ciclo de vida del MVC

Fuente: (Gómez R., Modelo Vista Controlador, 2017)

2.7.5.3. Ventajas y desventajas del patrón MVC

Se tienen ventajas como:

- La implementación se realiza de forma modular.
- Sus vistas muestran información actualizada siempre. El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
- Cualquier modificación que afecte al dominio, como aumentar métodos o datos contenidos, implica una modificación sólo en el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.
- Las modificaciones a las vistas no afectan al modelo de dominio, simplemente se modifica la representación de la información, no su tratamiento.
- MVC está demostrando ser un patrón de diseño bien elaborado pues las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones.

Como desventajas tenemos

- Para desarrollar una aplicación bajo el patrón de diseño MVC es necesario una mayor dedicación en los tiempos iniciales del desarrollo. Normalmente el patrón exige al programador desarrollar un mayor número de clases que, en otros entornos de desarrollo, no son necesarias. Sin embargo, esta desventaja es muy relativa ya que posteriormente, en la etapa de mantenimiento de la aplicación, una aplicación MVC es mucho más mantenible, extensible y modificable que una aplicación que no lo implementa.

- MVC requiere la existencia de una arquitectura inicial sobre la que se deben construir clases e interfaces para modificar y comunicar los módulos de una aplicación. Esta arquitectura inicial debe incluir, por lo menos, un mecanismo de eventos para poder proporcionar las notificaciones que genera el modelo de aplicación; una clase Modelo, otra clase Vista y una clase Controlador genéricas que realicen todas las tareas de comunicación, notificación y actualización que serán luego transparentes para el desarrollo de la aplicación.
- MVC es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma. (Marco de Desarrollo de la Junta de Andalucía, 2017)

2.8. FACTOR DE CALIDAD ISO 9126

El estándar ISO 9126 se desarrolló con la intención de identificar los atributos claves del software de cómputo.

La tabla siguiente muestra las preguntas centrales que atiende cada una de estas características. **Confiabilidad**, Grado en el que se espera que un programa cumpla con su función y con la precisión requerida, se pueden usar las siguientes ecuaciones:

Tabla 3 - Características de la norma ISO-9126

Características	Preguntas
Funcionabilidad	¿Las funciones y propiedades satisfacen las necesidades explícitas e implícitas?
Confiabilidad	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?
Usabilidad	¿El software es fácil de usar y de aprender?
Eficiencia	¿Es rápido y minimalista en cuanto al uso de recursos?

Mantenibilidad	¿Es fácil de modificar y verificar?
Portabilidad	¿Es fácil de transferir de un ambiente a otro?

Fuente: (Abud Figueroa, s.f.)

El estándar ISO-9126 establece que cualquier componente de la calidad del software puede ser descrito en términos de seis atributos cada una de las cuales se detalla a través de un conjunto de sub-atributos que permiten analizar y profundizar en la evaluación de la calidad de productos de software. A continuación, se explican estos atributos detalladamente:

Funcionalidad: En este grupo se conjunta una serie de atributos que permiten calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado. Para este propósito se establecen los siguientes atributos:

- **Adaptabilidad.** Se enfoca a evaluar si el software cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.
- **Exactitud.** Este atributo permite evaluar si el software presenta resultados o efectos acordes a las necesidades para las cuales fue creado.
- **Interoperabilidad.** Permite evaluar la habilidad del software de interactuar con otros sistemas previamente especificados.
- **Conformidad.** Evalúa si el software se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.
- **Seguridad.** Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o premeditado, a los programas y datos.

Confiabilidad: Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido. Las sub características que el estándar sugiere son:

- **Nivel de Madurez.** Permite medir la frecuencia de falla por errores en el software.
- **Tolerancia a fallas.** Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.
- **Recuperación.** Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesarios para lograrlo.

Usabilidad: Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.

- **Comprensibilidad.** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.
- **Facilidad de Aprender.** Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.
- **Operabilidad.** Agrupa los conceptos que evalúan la operación y el control del sistema.

Eficiencia: Esta característica permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados. Los aspectos a evaluar son:

- **Comportamiento con respecto al Tiempo.** Atributos del software relativos a los tiempos de respuesta y de procesamiento de los datos.
- **Comportamiento con respecto a Recursos.** Atributos del software relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones.

Mantenibilidad: Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad. En este caso, se tienen los siguientes factores:

- **Capacidad de análisis.** Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.
- Capacidad de modificación. Mide el esfuerzo necesario para modificar aspectos del software, remover fallas o adaptar el software para que funcione en un ambiente diferente.
- Estabilidad. Permite evaluar los riesgos de efectos inesperados debidos a las modificaciones realizadas al software.
- Facilidad de Prueba. Se refiere al esfuerzo necesario para validar el software una vez que fue modificado.

Portabilidad: En este caso, se refiere a la habilidad del software de ser transferido de un ambiente a otro, y considera los siguientes aspectos:

- **Adaptabilidad.** Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
- **Facilidad de Instalación.** Es el esfuerzo necesario para instalar el software en un ambiente determinado.
- **Conformidad.** Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad.
- **Capacidad de reemplazo.** Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares. (Abud Figueroa, sin fecha)

2.9. SEGURIDAD DE LA INFORMACION

ISO 27002

La seguridad de la información es un tema que ha ganado cuerpo en los últimos años, obteniendo espacio en los medios y convirtiéndose en «commodity», en empresas de los más variados portes y segmentos. En contrapartida es importante subrayar que la popularización del término SI (Seguridad de la Información) fue motivada por la elevación en el número de incidentes de seguridad, ocurridos a nivel mundial. Los trastornos generados por esos incidentes son variados, generando daños a la imagen del negocio o fuga de informaciones críticas, lo que puede resultar en pérdidas financieras sustanciales.

El aumento del número de ocurrencias influye en la percepción de valor sobre inversiones en SI y hacen que las empresas busquen la estructuración de procesos para garantizar que sus negocios estén protegidos contra los más variados tipos de amenazas virtuales.

En medio de este escenario surgió la norma internacional ISO/IEC 27002, que se centra en las buenas prácticas para gestión de la seguridad de la información. En los días de hoy esa es fundamental para la consolidación de un Sistema de Gestión de Seguridad de la Información (SGSI), garantizando la continuidad y el mantenimiento de los procesos de seguridad, alineados a los objetivos estratégicos de la organización. (www.ostec.blog, 2013)

2.10. ANALISIS DE COSTO DEL SISTEMA

MODELO CONSTRUCTIVO DE COSTOS (COCOMO)

En su libro clásico acerca de “economía de la ingeniería de software”, Barry Boehm introdujo una jerarquía de modelos de estimación de software que llevan el nombre

COCOMO, por *Constructive Cost Model*: modelo constructivo de costos. El modelo COCOMO original se convirtió en uno de los modelos de estimación de costo más ampliamente utilizados y estudiados en la industria. Evolucionó hacia un modelo de estimación más exhaustivo, llamado COCOMO II. Como su predecesor, COCOMO II en realidad es una jerarquía de modelos de estimación que aborda las áreas siguientes:

- Modelo de composición de aplicación, se usa durante las primeras etapas de la ingeniería de software, cuando son primordiales la elaboración de prototipos de las interfaces de usuario, la consideración de la interacción del software y el sistema, la valoración del rendimiento y la evaluación de la madurez de la tecnología.
- Modelo de etapa temprana de diseño, se usa una vez estabilizados los requisitos y establecida la arquitectura básica del software.
- Modelo de etapa post-arquitectónica, se usa durante la construcción del software.

Como todos los modelos de estimación para software, los modelos COCOMO II requieren información sobre dimensionamiento. Como parte de la jerarquía del modelo, están disponibles tres diferentes opciones de dimensionamiento: puntos objeto, puntos de función y líneas de código fuente (Pressman R. , 2010).

2.10.1. Características e inconvenientes

Pertenece a la categoría de modelos de subestimaciones basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en líneas de código principalmente. Es un modelo de estimación bien documentado y no registrado. Soporta el modelo en espiral de desarrollo.

A pesar de ser ampliamente utilizado en la actualidad, presenta los siguientes inconvenientes:

- Los resultados no son proporcionales a las tareas de gestión ya que no tiene en cuenta los recursos necesarios para realizarlas.
- Se puede desviar de la realidad si se indica mal el porcentaje de líneas de comentarios en el código fuente.
- Es un tanto subjetivo, puesto que está basado en estimaciones y parámetros que pueden ser "vistos" de distinta manera por distintos analistas que usen el método.
- Se miden los costes del producto, de acuerdo a su tamaño y otras características, pero no la productividad.
- La medición por líneas de código no es válida para orientación a objetos.
- Utilizar este modelo puede resultar un poco complicado, en comparación con otros métodos (que también sólo estiman).

2.10.2. Modelos de estimación

Las ecuaciones que se utilizan en los tres modelos son:

- $E = a(Kl)^b * m(X)$, en persona/mes
- $Tdev = c(E)^d$, en meses
- $P = E/Tdev$, en personas

donde:

- E es el esfuerzo requerido por el proyecto, en persona/mes.
- T_{dev} es el tiempo requerido por el proyecto, en meses.
- P es el número de personas requerido por el proyecto.
- a, b, c y d son constantes con valores definidos en una tabla, según cada submodelo.
- Kl es la cantidad de líneas de código, en miles.
- $m(X)$ es un multiplicador que depende de 15 atributos.

A la vez, cada submodelo también se divide en modos que representan el tipo de proyecto, y puede ser:

- Modo orgánico, un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).
- Modo semilibre o semi encajado, 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 y es difícil basarse en la experiencia, puesto que puede no haberla.

2.10.3. Modelo básico

Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes.

Tabla 4 - Constantes para el cálculo de costes Modelo Básico

MODO	a	b	c	d
Orgánico	2.40	1.05	2.50	0.38
Semi - Orgánico	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.32

Fuente y elaboración: Ingeniería de Software (R. Pressman, 2010)

- Personas necesarias por mes para llevar adelante el proyecto $MM = a(Kl)^b$
- Tiempo de desarrollo del proyecto $Tdev = c(MM)^d$
- Personas necesarias para realizar el proyecto $CosteH = MM/Tdev$
- Costo total del proyecto $CosteM = CosteH * salario promedio$

2.10.4. Modelo intermedio

Este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación.

Para este ajuste, al resultado de la fórmula general se multiplica por el coeficiente surgido de aplicar los atributos que se decidan utilizar. Los valores de las constantes a reemplazar en la fórmula son:

Tabla 5 - Constantes para el cálculo de costes Modelo Intermedio

MODO	a	b	c	d
Orgánico	3.20	1.05	2,5	0,38
Semi - Orgánico	3.00	1.12	2,5	0,35
Empotrado	2.80	1.20	2,5	0,32

Fuente y elaboración: Ingeniería de Software (R. Pressman, 2010)

Se puede observar que los exponentes son los mismos que los del modelo básico, confirmando el papel que representa el tamaño; mientras que los coeficientes de los modos orgánico y rígido han cambiado, para mantener el equilibrio alrededor del semilibre con respecto al efecto multiplicador de los atributos de coste.

Existe una escala de atributos que cuantifican el entorno de un proyecto, estos son: muy bajo, bajo, nominal, alto, muy alto y extremadamente alto. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula, estos son:

- RELY, garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto.
- DATA, tamaño de la base de datos en relación con el tamaño del programa.
- CPLX, representa la complejidad del producto.
- TIME, limitaciones en el porcentaje del uso de la CPU.
- STOR, limitaciones en el porcentaje del uso de la memoria.
- VIRT, volatilidad de la máquina virtual.
- TURN, tiempo de respuesta requerido.
- ACAP, calificación de los analistas.
- AEXP, experiencia del personal en aplicaciones similares.
- PCAP, calificación de los programadores.
- VEXP, experiencia del personal en la máquina virtual.
- LEXP, experiencia en el lenguaje de programación a usar.

- MODP, uso de prácticas modernas de programación.
- TOOL, uso de herramientas de desarrollo de software.
- SCED, limitaciones en el cumplimiento de la planificación.

Tabla 6 - Atributos para el cálculo de costes

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	

Fuente y elaboración: Ingeniería de Software (R. Pressman, 2010)

2.10.5. Modelo Detallado

Presenta principalmente dos mejoras respecto al anterior. Los factores correspondientes a los atributos son sensibles o dependientes de la fase sobre la que se realizan las estimaciones. Aspectos tales como la experiencia en la aplicación, utilización de herramientas de software, etc., tienen mayor influencia en unas fases que en otras, y además van variando de una etapa a otra. Establece una jerarquía de tres niveles de productos, de forma que los aspectos que representan gran variación a bajo nivel, se consideran a nivel módulo, los que representan pocas variaciones, a nivel de subsistema; y los restantes son considerados a nivel sistema.

2.11. HISTORIAL CLINICO

Es el elemento clave para el ejercicio profesional del personal sanitario, tanto desde el punto de vista asistencial, pues actúa como recordatorio para el manejo clínico del paciente, como desde el investigador y docente, permitiendo el análisis retrospectivo del quehacer profesional de los profesionales sanitarios. (Biocom, 2003)

Se define como un documento donde se recoge la información que procede de la práctica clínica relativa a un enfermo y donde se resumen todos los procesos a que ha sido sometido. Antiguamente, cuando apenas había hospitales y el médico atendía individualmente todas las necesidades del paciente, sus historias clínicas eran como un cuaderno de notas donde se registraba los datos más importantes según su criterio. (Biocom, 2003)

Cuando aparecen la especialización, el trabajo en equipo y la medicina hospitalaria, la historia clínica pasó a ser responsabilidad compartida de un grupo de profesionales. Esto obligó a estructurar la información de manera coordinada.

2.12. ODONTOLOGIA

La odontología es una rama de la salud y una ciencia que se encarga del estudio del aparato buco-dental incluyendo sus características anatómicas y fisiológicas, así como el análisis de sus diversas patologías y tratamientos respectivos, los cuales pueden ser tanto médicos como quirúrgicos.

Historia

Las prácticas odontológicas estaban de la mano con la medicina, las primeras prácticas datan de la antigua Mesopotamia entre el año 3000 y 3500 a.C., donde las dolencias de la boca se consideraban como invasión de espíritus malignos en el cuerpo. En roma se practicaba la medicina y la odontología de la misma manera no había distinción entre las enfermedades del cuerpo y la boca, la colocación de piezas de oro en los dientes era muy común entre la clase adinerada

Por su parte, los mayas no practicaban la odontología correctiva o restauradora, pero eran expertos en la colocación de piedras cortadas y que eran colocadas en los incisivos superiores e inferiores estas actividades eran de carácter religioso. De igual forma, los aztecas trataban las enfermedades de la boca con hierbas y derivados botánicos, pensaban que las caries eran ocasionadas por gusanos que eran eliminados comiendo ají picante.

Asimismo, Sebastián Garcilazo de la Vega, descendiente de Incas describía que la eliminación de las caries la realizaban utilizando un palillo encendido, para las enfermedades de las encías utilizaban bálsamo del Perú (*Myroxylon pereirae*) y en casos muy graves practicaban la cauterización. (www.definicion.xyz, 2017)

Especialidades Odontológicas.

- **Endodoncia:** se encarga del diagnóstico y tratamiento de las enfermedades de la pulpa dental, así como también de las medidas para prevenirlas, su procedimiento fundamental es quirúrgico y se denomina tratamiento conducto.
- **Ortodoncia:** se ocupa de la morfología facial y bucal en sus diferentes etapas de crecimiento, así como su corrección en caso de desviación de dicha morfología en sus funciones normales mediante el uso y control de distintos tipos de fuerza. La ortodoncia puede ser correctiva, contentiva, quirúrgica y preventiva.
- **Periodoncia:** es la especialidad médico quirúrgica de la odontología que se encarga de la prevención, diagnóstico y tratamiento de las enfermedades que afectan las encías (periodontitis y gingivitis).
- **Odontopediatra:** es la especialidad médica que se encarga de la salud bucal en niños y adolescentes, la diferencia entre ésta y la odontología habitual es el tratamiento de las caries en los dientes temporales, donde estas se tratan de una manera más agresiva.
- **Implantología:** esta especialidad se encarga de la creación de raíces artificiales para la colocación de piezas dentales artificiales.
- **Prostodoncia:** o prótesis dental es la que se encarga de devolver la función anatómica al aparato estomatognático como consecuencia de la pérdida de uno o más dientes.

- **Cirugía bucomaxilofacial:** es una especialidad que se encarga de la prevención estudio, diagnóstico, tratamiento y rehabilitación de la patología de la cavidad bucal y del territorio craneofacial, el cirujano buco maxilofacial atiende problemas de traumatología facial, labio leporino y paladar hendido.
- **Odontología estética:** esta especialidad soluciona problemas relacionados con la salud bucal y la armonía estética de la boca en su totalidad.
- **Odontología Forense:** es la aplicación de los conocimientos odontológicos con fines de identificación de cadáveres.

La odontología es muy importante, ya que nuestra boca cumple funciones como la trituración de alimentos, dado que es la entrada del aparato digestivo y con una boca enferma nos exponemos a un proceso y una digestión ineficientes, funciones fonéticas pues además de los sonidos y el habla también realizamos distintos gestos con los que nos comunicamos con las demás personas. Las expresiones faciales favorecen la comunicación verbal y no verbal, la boca participa activamente en el sentido del gusto a través de las papilas gustativas que nos permiten distinguir los sabores (dulce, amargo, ácido, agrio y salado), además juega un papel importante en nuestra autoestima las características externas de la boca y los dientes forman parte de nuestra apariencia física por la tanto es muy importante lo que proyectamos a los demás y el hecho de tener una boca sana eleva nuestra autoestima. (www.definicion.xyz, 2017)

CAPITULO III

MARCO APLICATIVO

3. MARCO APLICATIVO

3.1. INTRODUCCIÓN

Para un proyecto de desarrollo de software uno de los problemas que se presentan es no saber con exactitud los requerimientos del usuario desde el inicio de su elaboración ya que los mismo pueden cambiar en cualquier etapa del proyecto, es por esta razón que se hace uso de una metodología ágil el cual nos ayudara a lograr nuestro objetivo de una manera adecuada y un tiempo determinado.

En el presente capitulo se describe la implementación de la metodología ágil XP (programación extrema) para las fases de planificación, diseño, desarrollo y pruebas, del presente proyecto.

Para su mejor comprensión sobre la forma de aplicación de la metodología XP a continuación se muestra en la Tabla 7 El desarrollo del sistema empleando la Metodología ágil XP.

Tabla 7 - Fases y Procesos de la Metodología XP

FASES DE LA MERODOLOFIA XP	Procesos XP
PLANIFICACION	<ul style="list-style-type: none">• Historias de Usuarios• Plan de Entregas• Iteraciones• Tareas
DISEÑO	<ul style="list-style-type: none">• Tarjetas CRC
DESARROLLO	<ul style="list-style-type: none">• Interfaz grafica
PRUEBAS	<ul style="list-style-type: none">• Pruebas de Aceptación

Fuente: Elaboración propia

3.3. PLANIFICACION

En esta fase inicial de la metodología XP, se establece una comunicación continua entre el equipo de desarrollo y el cliente, para obtener principalmente los requerimientos y funcionalidades del sistema. Además, permite establecer el alcance del proyecto y fechas de entrega del sistema, tomando en cuenta en la prioridad y tiempo estimado para el desarrollo de cada historia de usuario.

3.3.1. Identificación de Tipos de Usuario (Roles)

La identificación y clasificación de los tipos de usuario permitir organizar la información evitando confusión y otorgara mejor control y seguimiento a los usuarios para el funcionamiento y acceso al mismo el cual hará que sea eficiente.

La siguiente Tabla 6 muestra la clasificación e identificación de los tipos de usuario (roles) que tendrán acceso al sistema.

Tabla 8 - Clasificación e Identificación de Roles

Rol	Descripcion
Administrador del sistema	Es quien administra los roles y usuarios
Odontólogo	Registra los datos creando la historia clínica para el seguimiento del paciente

Fuente: Elaboración propia

3.3.2. Historias de Usuario

Las historias de usuario deben ser descritas en un lenguaje común, para que puedan ser entendidas por todos (Desarrolladores y Usuarios), representando los requerimientos con los que debe cumplir el sistema a desarrollar.

La siguiente historia de usuario describe el control de acceso que deberá tener el sistema.

Tabla 9 - Historia de usuario: Control de acceso al sistema

Historia de Usuario	
Numero: 1	Usuario: administrador – odontólogo
Nombre historia: Control de acceso al sistema	
Prioridad en negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Israel Ivan Mendoza Apaza	
Descripción: En el sistema solo deberán acceder las personas autorizadas, que necesariamente sean personal de la clínica los cuales deberán tener un nombre de usuario y contraseña única	
Observaciones: Solo los usuarios que estén definidos en el sistema tendrán acceso.	

Fuente: (Elaboración propia)

Tabla 10 - Historia de usuario: Ingresar datos del odontólogo

Historia de Usuario	
Numero: 2	Usuario: administrador - odontólogo
Nombre historia: Ingresar datos del odontólogo	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Israel Ivan Mendoza Apaza	
Descripción: El sistema permitirá que el administrador realice el ingreso de datos de un nuevo médico mediante un formulario que contiene los siguientes campos: C.I., nombres, apellidos, correo, fecha de nacimiento, especialidad dirección, clave y teléfonos. El sistema validará cada uno de los campos ingresados. El sistema permitirá que el administrador guarde los datos ingresados en el formulario.	

Observaciones: El administrador y odontólogo tienen acceso al sistema con “usuario y contraseña”

Fuente: (Elaboración propia)

Tabla 11 - Historia de usuario: Modificar datos del odontólogo

Historia de Usuario	
Numero: 3	Usuario: Administrador - Odontólogo
Nombre historia: Modificar datos del odontólogo	
Prioridad en negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Israel Ivan Mendoza Apaza	
<p>Descripción: El sistema permitirá que el administrador seleccione el médico de quien se requiere realizar actualizaciones, permitiendo hacer búsquedas por: nombres, apellidos, celular o cedula.</p> <p>El sistema permitirá que el usuario realice la modificación de datos de un odontólogo.</p> <p>El sistema validará cada uno de los campos modificados y guarde las actualizaciones ingresados en el formulario.</p>	
Observaciones: Solo el personal autorizado podrá realizar cambios.	

Fuente: (Elaboración propia)

Tabla 12 - Historia de usuario: Ingresar datos del paciente

Historia de Usuario	
Numero: 4	Usuario: Odontólogo
Nombre historia: Ingresar datos del paciente	
Prioridad en negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Israel Ivan Mendoza Apaza	

<p>Descripción: El usuario luego de ingresar al sistema con rol de Odontólogo, seleccionará del menú la opción “Paciente”, donde podrá ingresar los datos personales de paciente. Podrá realizar la inserción y modificación de los datos.</p>
<p>Observaciones: Los datos personales constarán al número de CI, nombre completo, genero, fecha de nacimiento, número de celular, dirección de domicilio, correo electrónico, ocupación y la localidad. La modificación solo será de los datos dinámicos en el tiempo, tales como, (dirección de domicilio, correo, y número de teléfonos).</p>

Fuente: (Elaboración propia)

Tabla 13 - Historia de usuario: Modificar datos del paciente

Historia de Usuario	
Numero: 5	Usuario: odontólogo
Nombre historia: Modificar datos del paciente	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Israel Ivan Mendoza Apaza	
<p>Descripción: El sistema permitirá que el odontólogo seleccione el paciente de quien se requiere realizar actualizaciones, permitiendo hacer búsquedas por: nombres, apellidos, teléfono o cédula.</p> <p>El sistema permitirá que el odontólogo realice la modificación de datos de un paciente, en donde, los campos que se pueden modificar son: cédula de identidad, nombres, apellidos, localidad, fecha de nacimiento, sexo, ocupación, email, dirección, y teléfono.</p> <p>El sistema validará cada uno de los campos modificados y guarde los nuevos datos llenados el formulario.</p>	
Observaciones: Solo el Personal autorizado puede realizar esta acción	

Fuente: (Elaboración propia)

Tabla 14 - Historia de usuario: Consultar paciente

Historia de Usuario	
Numero: 6	Usuario: Odontólogo
Nombre historia: Consultar paciente	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Israel Ivan Mendoza Apaza	
Descripción: El sistema permitirá que el odontólogo pueda consultar información del paciente según un criterio de búsqueda en específico ya sea por: nombres, apellidos, cédula o teléfono.	
Observaciones: El paciente debe estar registrado para realizar la búsqueda.	

Fuente: (Elaboración propia)

Tabla 15 - Historia de usuario: administrar pagos

Historia de Usuario	
Numero: 7	Usuario: Odontólogo
Nombre historia: Crear tratamiento odontológico del paciente	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Israel Ivan Mendoza Apaza	
Descripción: El sistema permitirá que el odontólogo realice el ingreso de los datos de un nuevo tratamiento mediante el llenado de datos en un formulario que contiene los siguientes campos: selección tratamiento, detalle del tratamiento, selección del odontólogo que lo atenderá, fecha, pieza dental, estado del diente, costo, estado del tratamiento, diagnostico, procedimiento. El sistema validará cada uno de los campos ingresados y permitirá que el odontólogo guarde los datos ingresados en el formulario.	
Observaciones: El paciente debe estar registrado para recibir un tratamiento.	

Fuente: (Elaboración propia)

Tabla 16 - Historia de usuario: Odontólogo actualiza datos del tratamiento

Historia de Usuario	
Numero: 8	Usuario: Odontólogo
Nombre historia: Odontólogo actualiza datos del tratamiento	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Israel Ivan Mendoza Apaza	
<p>Descripción: El sistema permitirá que al odontólogo seleccione el paciente de quien se requiere realizar actualizaciones del tratamiento, permitiendo hacer búsquedas por: nombres, apellidos, teléfono o cédula.</p> <p>El sistema permitirá que el odontólogo realice la modificación de datos de un tratamiento, en donde, los campos que se pueden modificar son: selección tratamiento, detalle del tratamiento, selección del odontólogo que lo atenderá, fecha, pieza dental, estado del diente, costo, estado del tratamiento, diagnostico, procedimiento.</p> <p>El sistema validara cada uno de los campos modificados y permitirá al odontólogo guarde los nuevos datos llenados en el formulario</p>	
Observaciones: El paciente debe estar registrado para recibir un tratamiento.	

Fuente: (Elaboración propia)

Tabla 17 - Historia de usuario: Consulta del tratamiento

Historia de Usuario	
Numero: 9	Usuario: Odontólogo
Nombre historia: Consulta del tratamiento	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Israel Ivan Mendoza Apaza	

Descripción: El sistema permitirá que el médico pueda consultar información del paciente según un criterio de búsqueda en específico ya sea por: nombres, apellidos, cédula o código.
Observaciones: Debe existir pacientes registrados para la búsqueda de tratamientos.

Fuente: (Elaboración propia)

Tabla 18 - Historia de usuario: Administrar historial del paciente

Historia de Usuario	
Numero: 10	Usuario: Odontólogo
Nombre historia: Administrar historial del paciente	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Israel Ivan Mendoza Apaza	
Descripción: el usuario logueado con rol odontólogo, podrá elegir del menú la opción “historial”, en una lista aparecerán los pacientes que fueron registrados en el sistema. El odontólogo podrá ver el estado en el cual se encuentra el tratamiento del paciente	
Observaciones: El paciente debe estar registrado para tener un historial	

Fuente: (Elaboración propia)

Tabla 19 - Historia de usuario: Administrar pagos

Historia de Usuario	
Numero: 11	Usuario: odontólogo
Nombre historia: Administrar pagos	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Israel Ivan Mendoza Apaza	

Descripción: el usuario logueado con rol odontólogo ingresa a la opción de pagos y revisa el estado del tratamiento, así como la deuda que tiene el paciente. Se puede dejar a cuenta o cancelar el total del tratamiento
Observaciones: Este módulo se utilizará por el medico odontólogo cuando el paciente ya fue registrado anteriormente.

Fuente: (Elaboración propia)

Tabla 20 - Historia de usuario: Administrar citas medicas

Historia de Usuario	
Numero: 12	Usuario: odontólogo
Nombre historia: Administrar citas medicas	
Prioridad en negocio: media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Israel Ivan Mendoza Apaza	
Descripción: El usuario logueado en el sistema con el rol odontólogo, podrá elegir la opción “citas médicas”, donde se registrará la fecha de la siguiente atención del paciente, también se podrá ver un listado de las citas para ver si asistió a la cita médica.	
Observaciones: en caso de no asistir a la cita médica se lo registrara como inasistencia.	

Fuente: (Elaboración propia)

3.3.3. Identificación de Tareas

Las tareas están creadas en base a las historias de usuario, con el objetivo de identificar las tareas a realizar para implementar una historia de usuario. Los campos más importantes de una tarea son:

- **Tipo de tarea:** Existen varios tipos de tarea entre los cuales están el de desarrollo, corrección, mejora u otro.

- **Puntos estimados:** Son las semanas de desarrollo.
- **Fecha de Inicio y Fin:** Es la estimación de fechas en la que se planea iniciar y terminar las actividades de la tarea.

Es entonces que se desarrollará cada tarea haciendo referencia a su respectiva historia de usuario. En la Tabla 15 se muestran de manera general las correspondientes tareas a una historia de usuario y en las Tablas 3.24 - 3.53 se detallan cada una de ellas.

Tabla 21 - Tareas de historias de usuario

Numero de tarea	Numero de Historias de usuario	Nombre de la tarea
1	1	Control de acceso al sistema
2	1	Ingresar datos del odontólogo
3	1	Modificar datos del odontólogo
4	1	Ingresar datos del paciente
5	1	Modificar datos del paciente
6	1	Consultar paciente
7	1	Crear tratamiento odontológico del paciente
8	1	Odontólogo actualiza datos del tratamiento
9	1	Consulta del tratamiento
10	1	Administrar historial del paciente
11	1	Administrar pagos
12	1	Administrar citas medicas

Fuente: (Elaboración propia)

Tabla 22 - Tarea: Control de acceso al sistema

Tarea	
Numero de tarea: 1	Numero Historia: 1
Nombre tarea: Control de acceso al sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 11/01/2020	Fecha fin: 21/01/2020
Descripción: Se realizará el diseño de interfaz en el cual los usuarios del sistema podrán su nombre de usuario y contraseña los cuales serán validados y verificados en la base de datos. Posteriormente se realizará el diseño de la interfaz de usuario como ser el menú de acceso.	

Fuente: (Elaboración propia)

Tabla 23 - Tarea: Ingresar datos del odontólogo

Tarea	
Numero de tarea: 2	Numero Historia: 2
Nombre tarea: Ingresar datos del odontólogo	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 01/02/2020	Fecha fin: 24/02/2020
Descripción: Se diseñará la interfaz y se creará funciones para el registro, modificación y listado de los odontólogos que sean registrados, los datos básicos y obligatorios para el registro son: C.I., nombres, apellidos, correo, fecha de nacimiento, especialidad, dirección, clave y teléfonos.	

Fuente: (Elaboración propia)

Tabla 24 - Tarea: Modificar datos del odontólogo

Tarea	
Numero de tarea: 3	Numero Historia: 3
Nombre tarea: Modificar datos del odontólogo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 26/02/2020	Fecha fin: 03/03/2020
Descripción: Se realizará el diseño de interfaz para el formulario para modificar datos del odontólogo, se crearán funciones y las validaciones.	

Fuente: (Elaboración propia)

Tabla 25 - Tarea: Ingresar datos del paciente

Tarea	
Numero de tarea: 4	Numero Historia: 4
Nombre tarea: Ingresar datos del paciente	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 05/03/2020	Fecha fin: 29/03/2020
Descripción: Se diseñará la interfaz y se creará funciones para el registro, del paciente que será registrado, los datos básicos y obligatorios para el registro son CI, nombre completo, genero, fecha de nacimiento, número de celular, dirección de domicilio, correo electrónico, ocupación y la localidad. Se crearán los controladores, modelos y la vista	

Fuente: (Elaboración propia)

Tabla 26 - Tarea: Modificar datos del paciente

Tarea	
Numero de tarea: 5	Numero Historia: 5
Nombre tarea: Modificar datos del paciente	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 01/04/2020	Fecha fin: 05/04/2020
Descripción: Se diseñará la interfaz y se creará las funciones necesarias para modificar los datos del paciente	

Fuente: (Elaboración propia)

Tabla 27- Tarea: Administrar historial del paciente

Tarea	
Numero de tarea: 6	Numero Historia: 6
Nombre tarea: Consultar paciente	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 07/04/2020	Fecha fin: 12/04/2020
Descripción: Se creará las funciones y procedimientos necesarios para la búsqueda del paciente.	

Fuente: (Elaboración propia)

Tabla 28 - Tarea: Administrar historial del paciente

Tarea	
Numero de tarea: 7	Numero Historia: 7
Nombre tarea: Crear tratamiento odontológico del paciente	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 14/04/2020	Fecha fin: 28/04/2020
Descripción: Se diseñará la interfaz, se creará el controlador, modelo, vista y las validaciones para realizar el registro del tratamiento del paciente.	

Los datos que recibirá este formulario son: selección tratamiento, detalle del tratamiento, selección del odontólogo que lo atenderá, fecha, pieza dental, estado del diente, costo, estado del tratamiento, diagnóstico, procedimiento

Fuente: (Elaboración propia)

Tabla 29- Tarea: Odontólogo actualiza datos del tratamiento

Tarea	
Numero de tarea: 8	Numero Historia: 8
Nombre tarea: Odontólogo actualiza datos del tratamiento	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 30/04/2020	Fecha fin: 03/05/2020
Descripción: Se diseñará la interfaz para actualizar el tratamiento que tiene el paciente. Se creará las clases y funciones.	

Fuente: (Elaboración propia)

Tabla 30 - Tarea: Consulta del tratamiento

Tarea	
Numero de tarea: 9	Numero Historia: 9
Nombre tarea: Consulta del tratamiento	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 05/05/2020	Fecha fin: 10/05/2020
Descripción: Se creará las funciones para realizar la búsqueda del tratamiento	

Fuente: (Elaboración propia)

Tabla 31 - Tarea: Administrar historial del paciente

Tarea	
Numero de tarea: 10	Numero Historia: 10
Nombre tarea: Administrar historial del paciente	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 12/05/2020	Fecha fin: 24/05/2020
Descripción: Se diseñará la interfaz y se creará las funciones y procedimientos necesarios para acceder a la historia el paciente.	

Fuente: (Elaboración propia)

Tabla 32 - Tarea: Administrar pagos

Tarea	
Numero de tarea: 8	Numero Historia: 8
Nombre tarea: Administrar pagos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 26/05/2020	Fecha fin: 05/05/2020
Descripción: Se diseñará la interfaz, funciones y procedimientos para listar los pacientes que tienen deudas con la opción de ver el estado de la cuenta.	

Fuente: (Elaboración propia)

Tabla 33 - Tarea: Administrar citas medicas

Tarea	
Numero de tarea: 8	Numero Historia: 8
Nombre tarea: Administrar citas medicas	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 07/05/2020	Fecha fin: 26/05/2020
Descripción: Se diseñará la interfaz para ingresar las fechas de consulta.	

Fuente: (Elaboración propia)

3.3.4. Iteraciones

Las historias de usuario son clasificadas y ordenadas según a su prioridad para realizar la planificación de las iteraciones con estimación de tiempo para cada historia de usuario que a su vez se traduce en tareas específicas de programación.

Las iteraciones nos han de servir para medir el avance del proyecto una iteración terminada sin errores es una medida clara de avance.

Tabla 34 - Planificación de iteraciones

N°	Nombre	prioridad	Riesgo	Puntos Estimados	Iteración Asignada
1	Control de acceso al sistema	Alta	Media	1	1
2	Ingresar datos del odontólogo	Alta	Media	1	
3	Modificar datos del odontólogo	Alta	Media	1	2
4	Ingresar datos del paciente	Alta	Media	1	
5	Modificar datos del paciente	Media	Media	1	3
6	Consultar paciente	Media	Media	1	
7	Crear tratamiento odontológico del paciente	Media	Media	1	
8	Odontólogo actualiza datos del tratamiento	Media	Media	1	4
9	Consulta del tratamiento	Media	Media	1	
10	Administrar historial del paciente	Media	Media	1	5
11	Administrar pagos	Media	Media	1	
12	Administrar citas medicas	Media	Media	1	

Fuente: (Elaboración propia)

3.3.5. Plan de entregas

Partiendo de las iteraciones programadas se creó junto al cliente un plan de entregas, en donde se detalla la implementación de las historias de usuario de acuerdo a la iteración asignada a cada una de ellas.

El siguiente plan de entregas muestra un cronograma aproximado de entrega, para cada iteración.

Tabla 35 - Plan de entregas

Iteraciones semanas	Enero				Febrero				Marzo				Abril				Mayo			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Primera Iteración	■	■	■	■																
Segunda Iteración					■	■	■	■												
Tercera Iteración									■	■	■	■								
Cuarta Iteración													■	■	■	■				
Quinta Iteración																	■	■	■	■

Fuente: (Elaboración propia)

3.4. DISEÑO

En esta fase se presentan diseños simples como sugiere la metodología XP, que permitirán una mayor comprensión de la funcionalidad de las aplicaciones.

La base de datos propuesto será mostrada utilizando el modelo relacional¹⁴ para aprovechar las bondades de este tipo de modelado entre las que se encuentra, la

¹⁴ El modelo entidad relación es una herramienta que permite representar de manera simplificada los componentes que participan en un proceso de negocio y el modo en el que estos se relacionan entre sí.

facilidad en el diseño de las tablas, los datos y sus relaciones, además de su fácil entendimiento por parte de los desarrolladores de sistemas.

A continuación, en la Figura 8, se presentará el modelo de la base de datos:

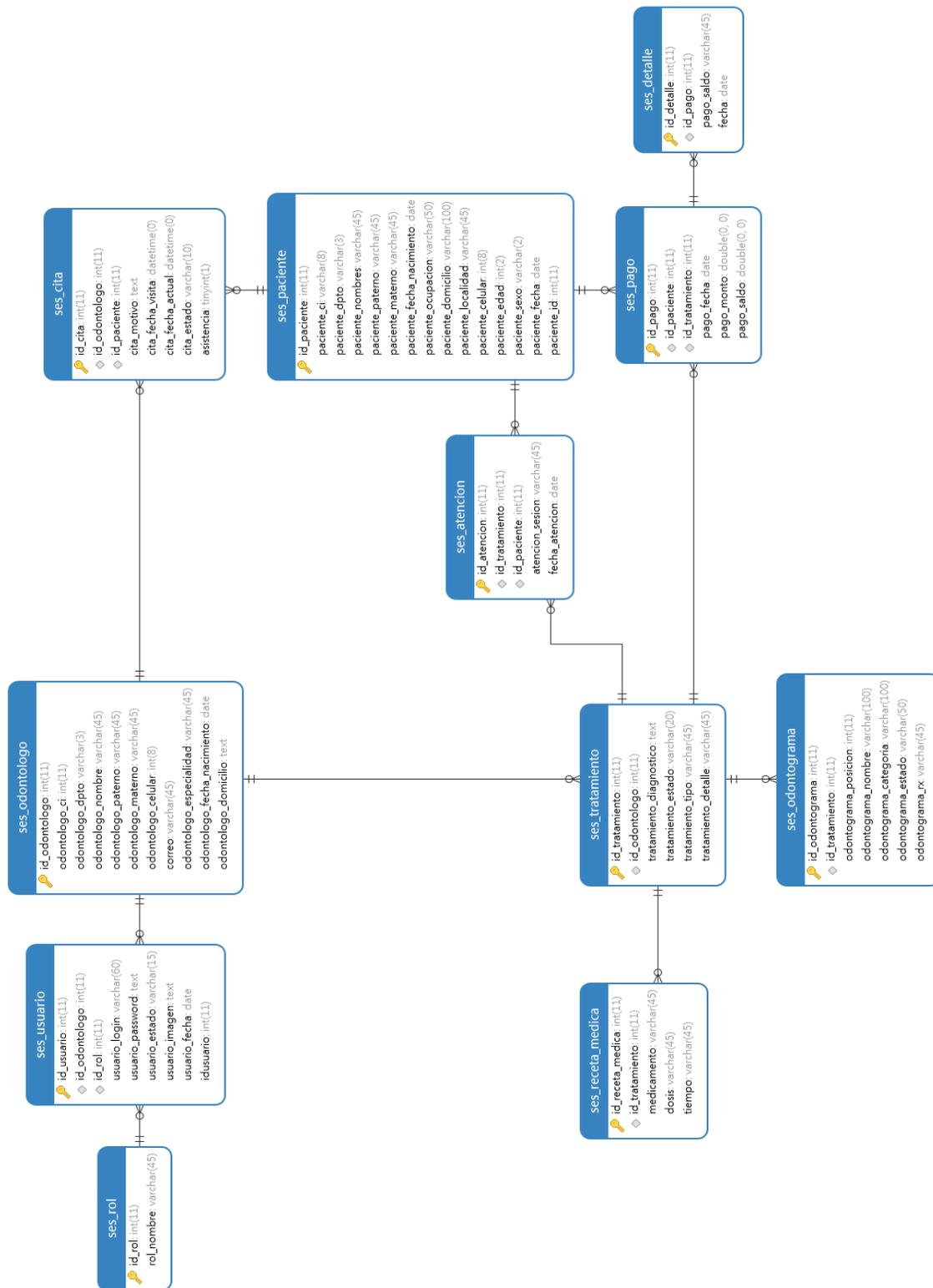


Figura 8. Modelo de la base de datos propuesto

Fuente: (elaboración propia)

3.4.1. Tarjetas CRC

En el diseño de las tarjetas CRC (clase, responsabilidad, colaborador) se realiza un inventario de las clases que vamos a necesitar para implementar el sistema y la forma en que van a interactuar. El uso de las Tarjetas CRC permite al programador centrarse y apreciar el desarrollo de la programación.

A continuación, se muestra el diseño de las tarjetas CRC para las principales historias de usuario:

Tabla 36 - Tarjeta CRC: Autenticación

Autenticación	
<ul style="list-style-type: none">• Verificar datos de usuario y contraseña• Registro de inicio de sesión	<ul style="list-style-type: none">• Usuario• Log acceso

Fuente: (Elaboración propia)

Tabla 37 - Tarjeta CRC: Usuario

Usuario	
<ul style="list-style-type: none">• Registrar al usuario• Insertar• Modificar• Eliminar• Mostrar	<ul style="list-style-type: none">• Administrador• Usuario

Fuente: (Elaboración propia)

Tabla 38 - Tarjeta CRC: Rol

Odontologo	
<ul style="list-style-type: none"> • Guardar • Modificar • Validar • Eliminar 	<ul style="list-style-type: none"> • Usuario

Fuente: (Elaboración propia)

Tabla 39 - Tarjeta CRC: Paciente

Paciente	
<ul style="list-style-type: none"> • Registrar • Modificar • Mostrar • Listar 	<ul style="list-style-type: none"> • Usuario

Fuente: (Elaboración propia)

Tabla 40 - Tarjeta CRC: Tratamiento

Tratamiento	
<ul style="list-style-type: none"> • Registrar • Detalle • Modificar • Mostrar • Búsqueda 	<ul style="list-style-type: none"> • Usuario • Paciente

Fuente: (Elaboración propia)

Tabla 41 - Tarjeta CRC: Detalle

Detalle	
<ul style="list-style-type: none"> • Guarda saldo • Registrar fecha 	<ul style="list-style-type: none"> • Usuario

Fuente: (Elaboración propia)

Tabla 42 - Tarjeta CRC: Odontograma

Odontograma	
<ul style="list-style-type: none"> • Guarda posicion • actualizar 	<ul style="list-style-type: none"> • Usuario

Fuente: (Elaboración propia)

Tabla 43 - Tarjeta CRC: Detalle

Receta medica	
<ul style="list-style-type: none"> • Busca • Registra • Guarda 	<ul style="list-style-type: none"> • Usuario

Fuente: (Elaboración propia)

Tabla 44 - Tarjeta CRC: Historial

Historial	
<ul style="list-style-type: none"> • Registro • Listado • busqueda 	<ul style="list-style-type: none"> • Usuario • paciente

Fuente: (Elaboración propia)

Tabla 45 - Tarjeta CRC: Cita

Cita	
<ul style="list-style-type: none"> • Listar • Registrar fecha • Eliminar 	<ul style="list-style-type: none"> • Usuario

Fuente: (Elaboración propia)

Tabla 46 - Tarjeta CRC: Atención

Atención	
<ul style="list-style-type: none">• Insertar• listar	<ul style="list-style-type: none">• Usuario

Fuente: (Elaboración propia)

3.5. DESARROLLO

En esta fase la codificación de cada una de las historias de usuario se realiza junto al cliente, permitiendo de esta forma una retroalimentación de lo que el cliente quiere priorizar más en las entregas del sistema.

Para esta fase de desarrollo se mostrará las interfaces de usuario más relevantes.

3.5.1. Implantación de las Interfaces de Usuario

A continuación, se muestra las interfaces de usuario desarrolladas, en cada una de las iteraciones planificadas.

a) **Primera iteración:** En esta iteración¹⁵ se pretende implementar las historias de usuario con mayor grado de prioridad para el cliente. Las historias de usuario que se implementaron en esta iteración, son los siguientes:

- **Historia de usuario 1:** Control de acceso al sistema
- **Historia de usuario 2:** Administración de la clínica dental

¹⁵ Repetir varias veces un proceso con la intención de alcanzar una meta deseada, objetivo o resultado. Cada repetición del proceso también se le denomina una "iteración", y los resultados de una iteración se utilizan como punto de partida para la siguiente iteración

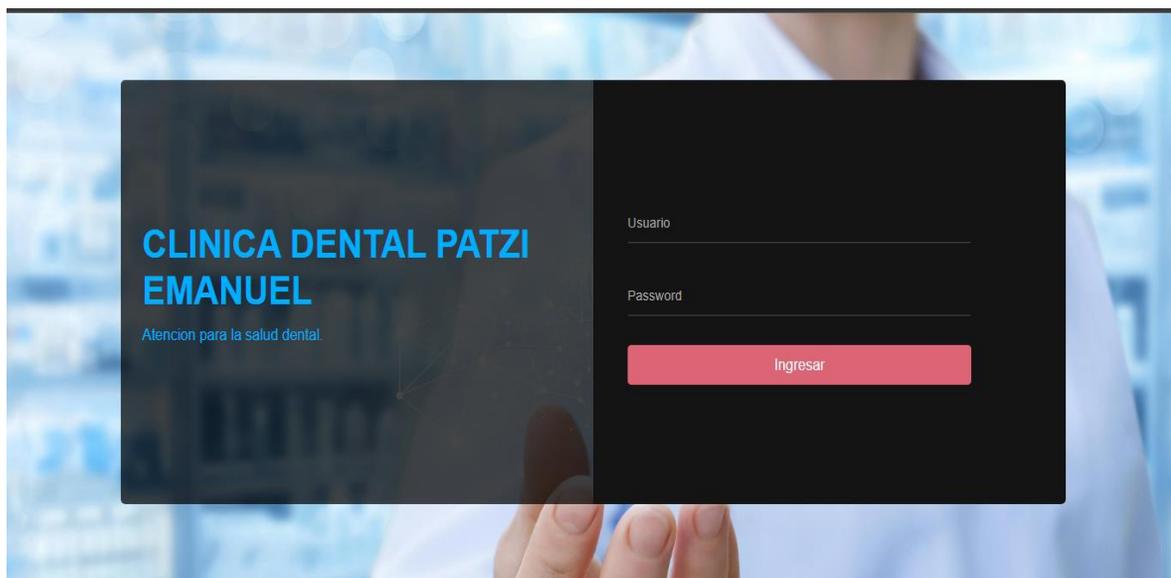


Figura 9. Pantalla: Control de acceso al sistema

Fuente: (elaboración propia)



Figura 10. Administración de la clínica dental

Fuente: (elaboración propia)

b) **Segunda iteración:** En esta iteración se pretende implementar las historias de usuario con mayor grado de prioridad para el cliente. Las historias de usuario que se implementaron en esta iteración, son los siguientes:

- **Historia de usuario 3:** Modificar datos del Odontólogo.
- **Historia de usuario 4:** Ingresar datos del paciente

MODIFICAR DATOS DEL ODONTOLOGO

USUARIO

NRO C.I.: 12345667 **LA PAZ**

MATERNO: MAMANI

CORREO: ivan@outlook.es

IMAGEN ANTIGUO: [Image placeholder]

ODONTOLOGO

NOMBRE: IVAN

TELEFONO: 3456789

ESPECIALIDAD: ORTODONCIA

TIPO USUARIO: Odontologo

USUARIO: ivan

PATERNO: LOPEZ

FECH. NAC.: 05/08/1990

IMAGEN NUEVO: Elegir archivo No se eligió ningún archivo

DOMICILIO: BALLVIA

CANCELAR **GUARDAR DATOS**

Figura 13. Pantalla: Modificación dato del Odontólogo

Fuente: (Elaboración propia)

Clinica Dental Patzi Online

SAKI SHIMIZU

PACIENTES TRATAMIENTO

ADMINISTRACION DE PACIENTES

+ NUEVO PACIENTE Imprimir pdf

10 records per page Search:

CARNET	NOMBRE PACIENTE	EDAD	SEXO	OCUPACION	DOMICILIO	CELULA	FECHA REGISTRO	PROCESO
3456976 LP	ANA LIMACHI MAMANI	29	F	ESTUDIANTE	ZONA VILLADORES CALLE INGAVI #56	1234567	2020-08-15	[Edit]
34765 LP	SAKI MAMANI KANTUTA	29	F	ESTUDIANTE	ZONA LA PORTADA AV ALCIDES ARGUEDAS # 42	43567890	2020-08-13	[Edit]
435678 LP	SAKI SHIMZU LOPEZ	29	F	ABOGADA	BALLIVIAN	56789	2020-08-25	[Edit]
4568 LP	JUAN MAMANI QUISPE	29	M	ESTUDIANTE	ZIBA BALLIVIAN	3456789	2020-08-24	[Edit]
6453456 LP	REINA CHOQUE LOPEZ	28	F	ESTUDIANTE	ZONA GERMAN BUSCH CALLE INGAVI #56	72344534	2020-08-17	[Edit]

Figura 14. Pantalla: listado de pacientes

Fuente: (Elaboración propia)

Figura 15. Pantalla: Registro de pacientes

Fuente: (Elaboración propia)

c) **Tercera iteración:** En esta iteración se pretende implementar las historias de usuario con mayor grado de prioridad para el cliente. Las historias de usuario que se implementaron en esta iteración, son los siguientes:

- **Historia de Usuario 5:** Modificar datos del paciente.
- **Historia de Usuario 6:** Consultar paciente.

Figura 16. Pantalla: Modificar de pacientes

Fuente: (Elaboración propia)

10 records per page Search:

CARNET	NOMBRE PACIENTE	EDAD	SEXO	OCUPACION	DOMICILIO	CELULAR	FECHA REGISTRO	PROCESO
3456976 LP	ANA LIMACHI MAMANI	29	F	ESTUDIANTE	ZONA VILLADORES CALLE INGAVI #56	1234567	2020-08-15	
34765 LP	SAKI MAMANI KANTUTA	29	F	ESTUDIANTE	ZONA LA PORTADA AV ALCIDES ARGUEDAS # 42	43567890	2020-08-13	
435678 LP	SAKI SHIMZU LOPEZ	29	F	ABOGADA	BALLIVIAN	56789	2020-08-25	
4568 LP	JUAN MAMANI QUISPE	29	M	ESTUDIANTE	ZIBA BALLIVIAN	3456789	2020-08-24	
6453456 LP	REINA CHOQUE LOPEZ	28	F	ESTUDIANTE	ZONA GERMAN BUSCH CALLE INGAVI #56	72344534	2020-08-17	
6845612 CB	NELSON JIMENEZ QUISPE	21	M	ESTUDIANTE	ZONA LA PORTADA	72345376	2020-08-27	
6951613 LP	ROMINA LOPEZ FORONDA	29	F	LICENCIADA	ZONA LA PORTADA AV. ALCIDES ARGUEDAS #42	1234567	2020-08-13	

Showing 1 to 7 of 7 entries

← Previous 1 Next →

Figura 17. Pantalla: Consulta de pacientes

Fuente: (Elaboración propia)

d) **Cuarta iteración:** En esta iteración se pretende implementar las historias de usuario con mayor grado de prioridad para el cliente. Las historias de usuario que se implementaron en esta iteración, son los siguientes:

- **Historia de usuario 7:** Crear tratamiento.
- **Historia de usuario 8:** Odontólogo actualiza datos del tratamiento.
- **Historia de usuario 9:** Consulta del tratamiento

AGREGAR NUEVO TRATAMIENTO

✕

SELECCIONA PACIENTE:
SAKI SHIMZU LOPEZ

DENTICIÓN PERMANENTE

DENTICIÓN TEMPORAL

SELECCIONA TRATAMIENTO:
CIRUGIA

FECHA:
dd/mm/aaaa

ESTADO DEL DIENTE:

DETALLE DEL TRATAMIENTO:

PIEZA:

COSTO:

ESTADO DEL TRATAMIENTO:
EN TRATAMIENTO

SELECCIONA MEDICO:
ISRAEL IVAN MENDOZA APAZA

NOMBRE DE LA PIEZA:

DIAGNOSTICO:
DIAGNOSTICO ...

CANCELAR
GUARDAR DATOS

Figura 18. Pantalla: Agregar tratamiento
Figura: (Elaboración propia)

MODIFICAR TRATAMIENTO

✕

SELECCIONA PACIENTE:
MARIANA GONZALES MANIA CI:647578 LP

SELECCIONA TRATAMIENTO:
PERIODONCIA

FECHA:
27/07/2020

CATEGORIA:
ADULTOS

DIAGNOSTICO:
ninguno

DETALLE DEL TRATAMIENTO:
ret

PIEZA:
35

ESTADO DEL DIENTE:
fdmnifg

PROCEDIMIENTO:
se realizo el tratamiento

SELECCIONA MEDICO:
NATI QUIspe

NOMBRE DE LA PIEZA:
12

ESTADO DEL TRATAMIENTO:
EN TRATAMIENTO

CANCELAR
GUARDAR DATOS

Figura 19. Pantalla: Modificar tratamiento
Fuente: (Elaboración propia)

CARNET	NOMBRE COMPLETO	DIENTE	DIAGNOSTICO	TRATAMIENTO	ESTADO	fecha atencion	COSTO	
6951613 LP	ROMINA LOPEZ FORONDA	32	ninguno	tallado	EN TRATAMIENTO	2020-08-12	230	
34765 LP	SAKI MAMANI KANTUTA	35	ninguno	puesta de porcelana	EN TRATAMIENTO	2020-08-12	200	
3456976 LP	ANA LIMACHI MAMANI	38	ninguno	puesta de porcelana	EN TRATAMIENTO	2020-08-14	100	
6951613 LP	ROMINA LOPEZ FORONDA	26	ninguno	revisión	FINALIZADO	2020-08-22	180	
6951613 LP	ROMINA LOPEZ FORONDA	22	necrosis pulpar	tratamiento de contacto	EN TRATAMIENTO	2020-08-24	150	
4568 LP	JUAN MAMANI QUISPE	22	ninguno	tallado	EN TRATAMIENTO	2020-08-24	150	

Figura 20. Pantalla: consulta del tratamiento

Fuente: (Elaboración propia)

e) **Quinta iteración:** En esta iteración se pretende implementar las historias de usuario con mayor grado de prioridad para el cliente. Las historias de usuario que se implementaron en esta iteración, son los siguientes:

- **Historia de usuario 10:** Administrar historial del paciente.
- **Historia de usuario 11:** Administrar pagos.
- **Historia de usuario 12:** Administrar citas medicas

HISTORIAL DE TRATAMIENTOS

[VOLVER A MENU](#)

10 records per page

Search:

CARNET	NOMBRE	TRATAMIENTO	ESTADO	ODONTOLOGO	FECHA	ACCION
3456976 LP	ANA LIMACHI MAMANI	RESTAURACIONES	EN TRATAMIENTO	ERIKA APAZA LOPEZ	2020-08-14	DETALLE

Showing 1 to 1 of 1 entries

← Previous 1 Next →

Figura 21. Pantalla: Historial del tratamiento

Fuente: (Elaboración propia)

ISRAEL MENDOZA

BUSQUEDA PACIENTE INGRESOS **ODONTOLOGOS**

CARNET DEL ODONTOLOGO: 1234567

Detalle

Resumen

ODONTOLOGO	FECHA ATENCIÓN	PACIENTE	TRATAMIENTO	PRECEDIMIENTO
ISRAEL IVAN MENDOZA	2020-08-11 00:00:00	SAKI SHIMIZU	PERIODONCIA	se realizara el aspado
ISRAEL IVAN MENDOZA	2020-08-11 00:00:00	HUGO BELTRAN	PERIODONCIA	revisión
ISRAEL IVAN MENDOZA	2020-08-19 00:00:00	LILIANA CATARI	CIRUGIA	se realizara el tallado
ISRAEL IVAN MENDOZA	2020-12-12 00:00:00	SAKI SHIMIZU	CIRUGIA	ljlklj

Figura 22. Pantalla: Historial de Atención

Fuente: (Elaboración propia)

Clinica Dental Patzi Online

SAKI SHIMIZU

ADMINISTRACION DE PAGOS

10 records per page Search:

CARNET	NOMBRE	FECHA DE ATENCIÓN	TRATAMIENTO	DIENTE	ESTADO	COSTO	SALDO	ACCION
3456976 LP	ANA LIMACHI MAMANI	2020-08-25	RESTAURACIONES	38	EN TRATAMIENTO	100	0	OPAGAR
34765 LP	SAKI MAMANI KANTUTA	2020-08-27	RESTAURACIONES	35	EN TRATAMIENTO	200	0	OPAGAR
4568 LP	JUAN MAMANI QUISPE	2020-08-28	CIRUGIA	22	EN TRATAMIENTO	150	0	OPAGAR
6951613 LP	ROMINA LOPEZ FORONDA	2020-08-26	PERIODONCIA	32	EN TRATAMIENTO	230	80	OPAGAR
6951613 LP	ROMINA LOPEZ FORONDA	2020-08-24	PERIODONCIA	26	FINALIZADO	180	180	OPAGAR
6951613 LP	ROMINA LOPEZ FORONDA	2020-08-24	OPERATORIA-ENDODONCIA	22	EN TRATAMIENTO	150	150	OPAGAR

Showing 1 to 6 of 6 entries

← Previous 1 Next →

Figura 23. Pantalla: Administración de pagos

Fuente: (Elaboración propia)

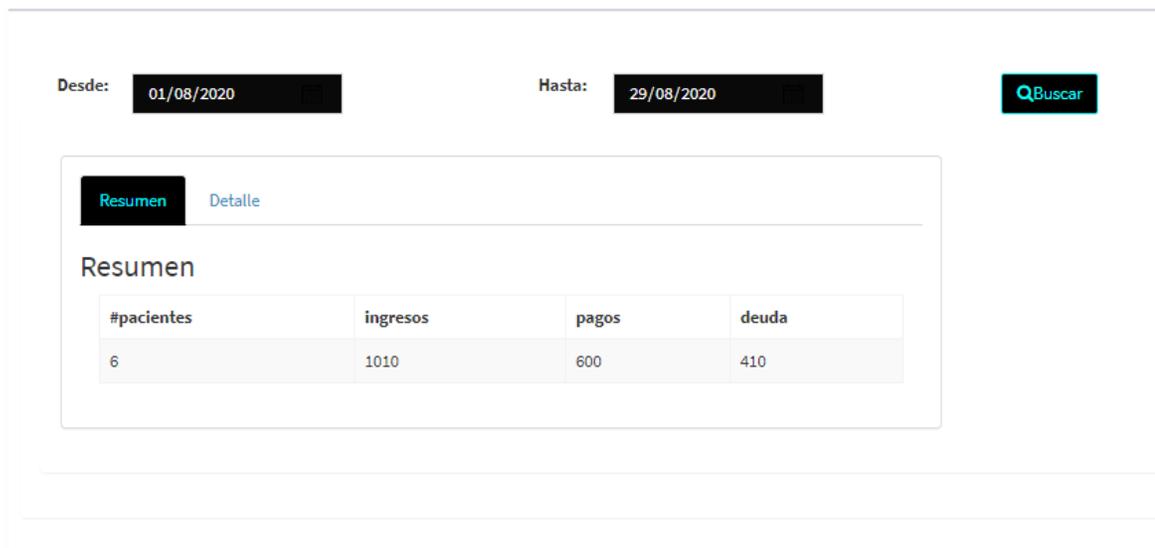


Figura 24. Pantalla: Ingreso de pagos
Fuente: (Elaboración propia)



Figura 25. Pantalla: Administración de Deudas
Fuente: (Elaboración propia)

REGISTRO DE RECETA MEDICA

+ NUEVA RECETA

10 records per page Search:

	CARNET	NOMBRE	APELLIDOS	TRATAMIENTO	MEDICAMENTO	DOSIS	TIEMPO
	647578 LP	MARIANA	GONZALES MANIA	PERIODONCIA	amoxicilina	cada 8 horas	4 dias
	8263462 TJ	SAKI	SHIMIZU OHARA	PERIODONCIA	amoxicilina	cada 8 horas	4 dias
	579238758972 BN	LILIANA	CATARI MAMANI	CIRUGIA	aspirina	cada 10 horas	2 dias
	647578 LP	MARIANA	GONZALES MANIA	PERIODONCIA	amoxicilina	10ml	4 dias

Showing 1 to 4 of 4 entries ← Previous 1 Next →

Figura 26. Pantalla: Administración receta Medica

Fuente: (Elaboración propia)

REGISTRO DE RECETA MEDICA

AGREGAR NUEVA RECETA

SELECCIONE TRATAMIENTO: SAKI SHIMIZU OHARA – Cl:8263462 TJ//PERIODI

MEDICAMENTO: amoxicilina

DOSIS: 100ml

TIEMPO: 4dias

10 records per page Search:

	CARNET	NOMBRE	APELLIDOS	TRATAMIENTO	MEDICAMENTO	DOSIS
	647578 LP	MARIANA	GONZALES MANIA	PERIODONCIA	amoxicilina	cada 8 h
	8263462 TJ	SAKI	SHIMIZU OHARA	PERIODONCIA	amoxicilina	cada 8 h

CANCELAR **GUARDAR DATOS**

Figura 27. Pantalla: Registro de nueva Receta

Fuente: (Elaboración propia)

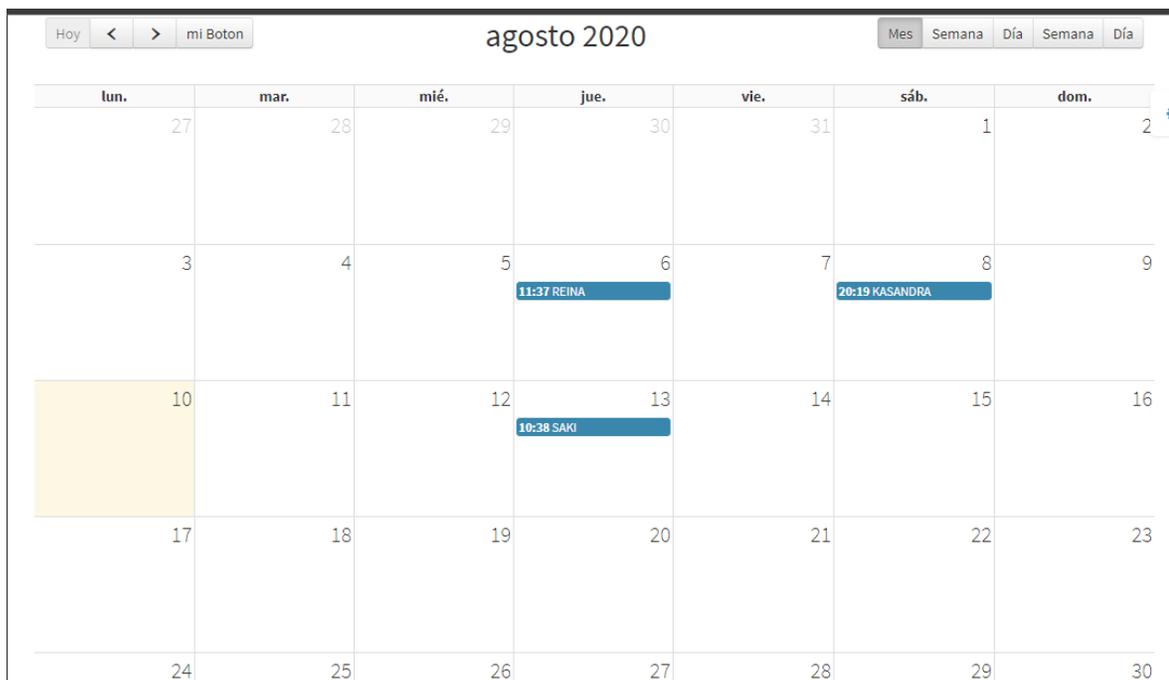


Figura 28. Pantalla: Registro de nueva Cita

Fuente: (Elaboración propia)

3.6. PRUEBAS

3.6.1. Pruebas de Aceptación

La fase de pruebas es una de las fases más importantes ya que nos permite verificar junto al cliente que se pudo atender los requerimientos especificados en las historias de usuario.

También sirven como retroalimentación¹⁶ para ver que las historias de usuario que fueron implementadas en versiones anteriores y que necesitan ser modificadas, mejoradas o simplemente descartadas.

¹⁶ La retroalimentación es un proceso mediante el cual un sistema, puede ser de cualquier tipo, recibe energía proporcionada por sí mismo para así garantizar su funcionamiento. Es decir, que el sistema en sí mismo entiende cual es el procedimiento que van a realizar todos los componentes

El tipo de pruebas realizadas en esta fase son pruebas de aceptación descritas a continuación.

Tabla 47 - Prueba de aceptación: Control de acceso al sistema

Prueba de Aceptación	
Número: 1	Historia de Usuario: 1
Nombre: Control de acceso al sistema	
Descripción: Desarrollo para controlar el acceso al sistema	
Condiciones de Ejecución: El usuario deberá contar con un nombre de usuario y contraseña.	
Pasos de Ejecución: Ingresar datos de usuario y contraseña, luego iniciar sesión.	
Resultado esperado: Acceso a todas las funcionalidades del sistema dependiendo del tipo de usuario y el rol que desempeña en el mismo.	
Evaluación de Prueba: Aceptada	

Fuente: (Elaboración propia)

Tabla 48 - Pruebas de Aceptación: Ingresar datos del odontólogo

Prueba de Aceptación	
Número: 2	Historia de Usuario: 2
Nombre: Ingresar datos del odontólogo	
Descripción: Desarrollo para realizar el registro del odontólogo	
Condiciones de Ejecución: El usuario debe iniciar sesión, ingresar en el menú “administración usuario” y escoger la opción nuevo usuario	
Pasos de Ejecución: Para registrar un nuevo odontólogo deberá seleccionar la opción NUEVO USUARIO y luego rellenar los datos solicitados en el formulario.	
Resultado esperado: El usuario obtiene un listado de todos los odontólogos registrados.	
Evaluación de Prueba: Aceptada	

Fuente: (Elaboración propia)

Tabla 49 - Pruebas de Aceptación: Modificar datos del odontólogo

Prueba de Aceptación	
Número: 3	Historia de Usuario: 3
Nombre: Modificar datos del odontólogo	
Descripción: Modificar datos del odontólogo	
Condiciones de Ejecución: El usuario deberá iniciar sesión, ingresar en el menú "Administración Usuario".	
Pasos de Ejecución: Para modificar los datos se deberá escoger la opción EDITAR y editar los campos dinámicos.	
Resultado esperado: El usuario obtiene el mensaje de datos modificados exitosamente	
Evaluación de Prueba: Aceptada	

Fuente: (Elaboración propia)

Tabla 50 - Pruebas de Aceptación: Ingresar datos del paciente

Prueba de Aceptación	
Número: 4	Historia de Usuario: 4
Nombre: Ingresar datos del paciente	
Descripción: Desarrollo para insertar datos del paciente	
Condiciones de Ejecución: El usuario deberá iniciar sesión, ingresar en el menú Administración Paciente.	
Pasos de Ejecución: Se ingresa al menú paciente, escoger la opción NUEVO PACIENTE y rellenar los datos solicitados por el formulario.	
Resultado esperado: Datos guardados exitosamente	
Evaluación de Prueba: Aceptada	

Fuente: (Elaboración propia)

Tabla 51 - Pruebas de Aceptación: Modificar datos del paciente

Prueba de Aceptación	
Número: 5	Historia de Usuario: 5
Nombre: Modificar datos del paciente	
Descripción: Desarrollo para la modificación de los datos del paciente	
Condiciones de Ejecución: El usuario deberá iniciar sesión, ingresar en el menú Administración Paciente	
Pasos de Ejecución: Se ingresa al menú paciente, escoger la opción editar y modificar los datos que fueron erróneos.	
Resultado esperado: Datos modificados exitosamente	
Evaluación de Prueba: Aceptada	

Fuente: (Elaboración propia)

Tabla 52 - Pruebas de Aceptación: Consultar paciente

Prueba de Aceptación	
Número: 6	Historia de Usuario: 6
Nombre: Consultar paciente	
Descripción: Desarrollo para consultar información del paciente	
Condiciones de Ejecución: El usuario deberá iniciar sesión.	
Pasos de Ejecución: Se ingresa al menú Administración Paciente, ingresar datos: C.I., nombre, apellidos y teléfono estos campos se ingresan en el espacio buscar	
Resultado esperado: Información encontrada	
Evaluación de Prueba: Aceptada	

Fuente: (Elaboración propia)

Tabla 53 - Pruebas de Aceptación: Crear tratamiento odontológico del paciente

Prueba de Aceptación	
Número: 7	Historia de Usuario: 7
Nombre: Crear tratamiento odontológico del paciente	
Descripción: Desarrollo para insertar datos del tratamiento	
Condiciones de Ejecución: El usuario deberá iniciar sesión.	
Pasos de Ejecución: Se ingresará al menú TRATAMIENTO, escoger la opción NUEVO TRATAMIENTO y rellenar los datos solicitados por el paciente.	
Resultado esperado: Datos guardados exitosamente	
Evaluación de Prueba: Aceptada	

Fuente: (Elaboración propia)

Tabla 54 - Pruebas de Aceptación: Odontólogo actualiza datos del tratamiento

Prueba de Aceptación	
Número: 8	Historia de Usuario: 8
Nombre: Odontólogo actualiza datos del tratamiento	
Descripción: Desarrollo para actualizar el tratamiento	
Condiciones de Ejecución: El usuario deberá iniciar sesión.	
Pasos de Ejecución: Se ingresará al menú TRATAMIENTO, escoger la opción EDITAR y modificar los datos dinámicos.	
Resultado esperado: Datos modificados exitosamente	
Evaluación de Prueba: Aceptada	

Fuente: (Elaboración propia)

Tabla 55 - Pruebas de Aceptación: Consulta del tratamiento

Prueba de Aceptación	
Número: 9	Historia de Usuario: 9
Nombre: Consulta del tratamiento	
Descripción: Desarrollo para consultar datos del tratamiento	
Condiciones de Ejecución: El usuario deberá iniciar sesión.	
Pasos de Ejecución: Se ingresará al menú TRATAMIENTO, escoger la opción BUSQUEDA y ingresar los datos: nombre, apellidos, C.I., teléfono.	
Resultado esperado: Datos del tratamiento encontrados	
Evaluación de Prueba: Aceptada	

Fuente: (Elaboración propia)

Tabla 56 - Prueba de Aceptación Consulta del tratamiento

Prueba de Aceptación	
Número: 10	Historia de Usuario: 10
Nombre: Administrar historial del paciente	
Descripción: Desarrollo para acceder al historial del paciente	
Condiciones de Ejecución: El usuario deberá iniciar sesión.	
Pasos de Ejecución: Se ingresará al menú HISTORIAL, donde podrá ver el listado de los pacientes. Elegir la opción VER para acceder al historial	
Resultado esperado: Acceder al historial del paciente	
Evaluación de Prueba: Aceptada	

Fuente: (Elaboración propia)

Tabla 57 - Prueba de Aceptación: Administrar pagos

Prueba de Aceptación	
Número: 11	Historia de Usuario: 11
Nombre: Administrar pagos	
Descripción: Desarrollo para la función pagos	
Condiciones de Ejecución: El usuario deberá iniciar sesión.	
Pasos de Ejecución: El usuario odontólogo escogerá la opción pagos, escogerá la opción pagar, ingresará el monto a pagar. La opción detalle mostrara el estado de la cuenta	
Resultado esperado: Datos guardados exitosamente	
Evaluación de Prueba: Aceptada	

Fuente: (Elaboración propia)

Tabla 58 - Prueba de Aceptación: Administrar citas medicas

Prueba de Aceptación	
Número: 12	Historia de Usuario: 12
Nombre: Administrar citas medicas	
Descripción: Desarrollo para guardar fecha de la cita	
Condiciones de Ejecución: El usuario deberá iniciar sesión, ingresar en el menú citas médicas.	
Pasos de Ejecución: El usuario odontólogo programara una cita en el calendario, hará click en la fecha a programar y después guardara el evento	
Resultado esperado: Datos guardados exitosamente	
Evaluación de Prueba: Aceptada	

Fuente: (Elaboración propia)

3.7. PRUEBAS DE SOFTWARE

3.7.1. Pruebas de Caja Blanca

Se toma en cuenta comportamiento interno y la estructura del programa, aplicando la cobertura de caminos describiremos casos de pruebas suficientes para que se ejecuten todos los caminos de un programa, entendiendo camino como una sentencia¹⁷ de sentencias encadenadas desde la entrada hasta su salida.

¹⁷ Son las unidades ejecutables más pequeña de un programa, especifican y controlan el flujo y orden de ejecución del programa.

3.7.1.1. Prueba de Caja Blanca – inicio de sesión

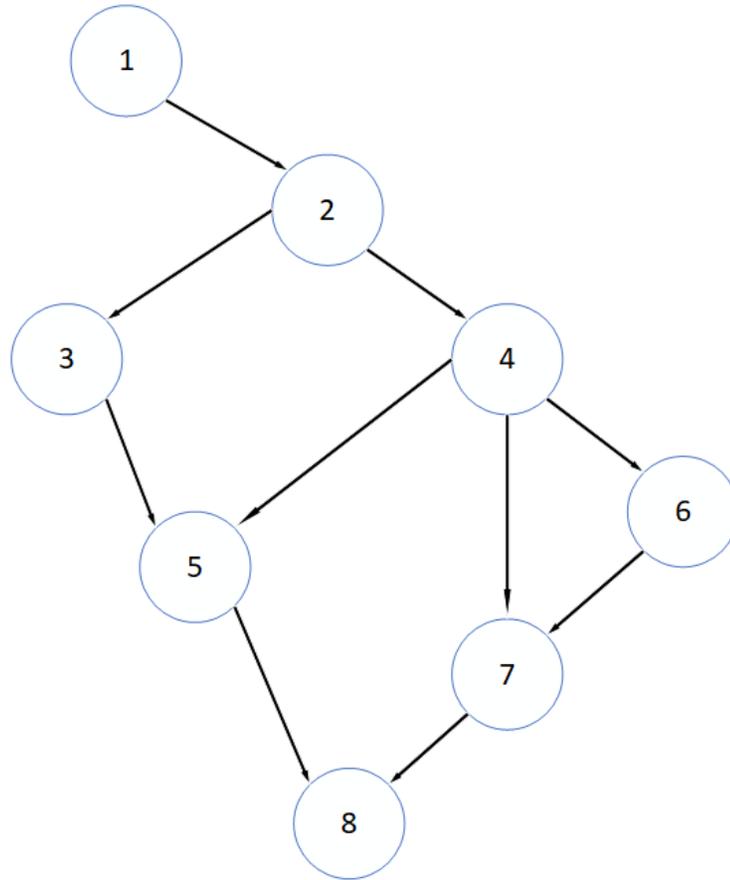


Figura 29. Prueba de caja blanca: Inicio de sesión

Fuente: (Elaboración propia)

Ahora determinamos el conjunto de caminos independientes. Los caminos deben ser probados dadas ciertas variables.

Camino 1: 1, 2, 3, 5, 8

Camino 2: 1, 2, 4, 7, 8

Camino 3: 1, 2, 4, 5, 8

Camino 4: 1, 2, 4, 6, 7, 8

Tabla 59 - Evaluación de flujo sesión

Numero	Ruta	Parámetro de entrada.
1	1,2,3,5,8	Se verifica la información.
2	1,2,4,7,8	Se verifica los campos ingresados, en caso de error no ingresa.
3	1,2,4,5,8	Verifica los datos del usuario.
4	1,2,4,5,6,7,8	Verifica los campos para ingresar al sistema.

Fuente: (Elaboración propia)

3.7.1.2. Prueba de caja blanca – atención al Paciente

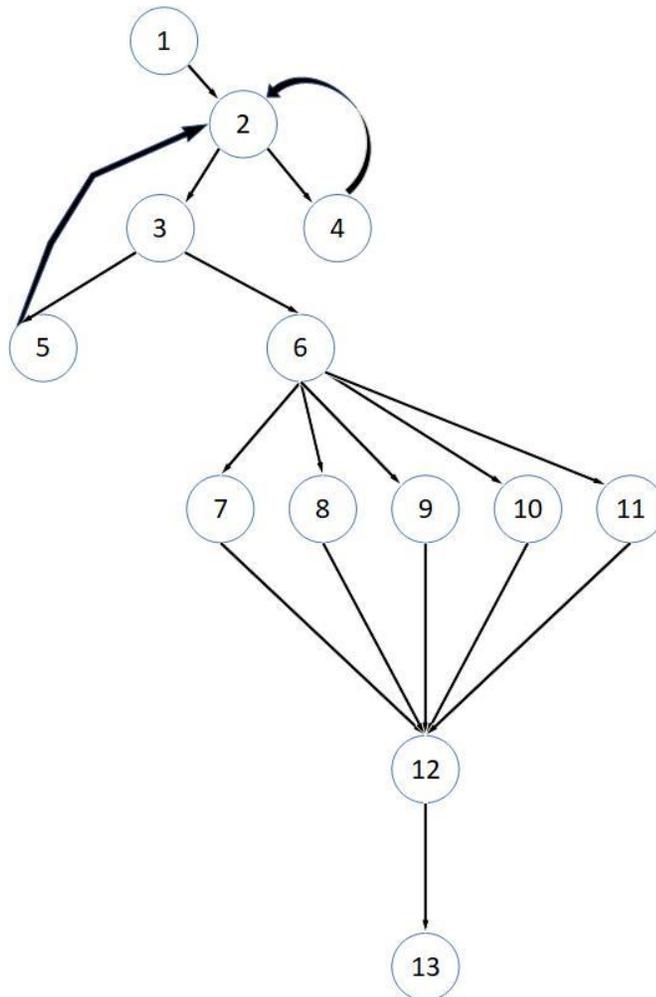


Figura 30. Prueba de caja blanca: atención al cliente

Fuente: (Elaboración propia)

Tabla 60 - Evaluación de flujo atención paciente

Numero	Ruta	Parámetro de entrada.
1	1,2,4	Se verifica el usuario y password, sin son erróneos vuelve al login.
2	1,2,3,5	Si verifica usuario y password, ingresa al sistema. Si no hay datos que ingresar sale del sistema
3	1,2,3,6,7	Realiza el ingreso del tratamiento
4	1.2.3.6.8	Realiza el historial del paciente, ingresando el tipo de tratamiento
5	1,2,3,6,9	Se listan y registran os pagos por los tratamientos.
6	1,2,3,6,10	Se muestra las citas programadas y en caso de ser necesario añadir otra cita.
7	1,2,3,6,11	Lista a los pacientes y registra la receta apropiada pata su tratamiento.
8	1,2,3,6,7,8,9,10,11,12	Registrado los datos se procede al fin del ciclo.

Fuente: (Elaboración propia)

3.7.2. Prueba de Caja Negra

En las pruebas de caja negra, nos enfocamos solamente en las entradas y salidas del sistema, sin preocuparnos en tener conocimiento de la estructura interna del programa de software. Para obtener el detalle de cuáles deben ser esas entradas y salidas, nos basamos en los requerimientos de software y especificaciones funcionales.

3.7.2.1. Prueba de caja negra inicio de sesión

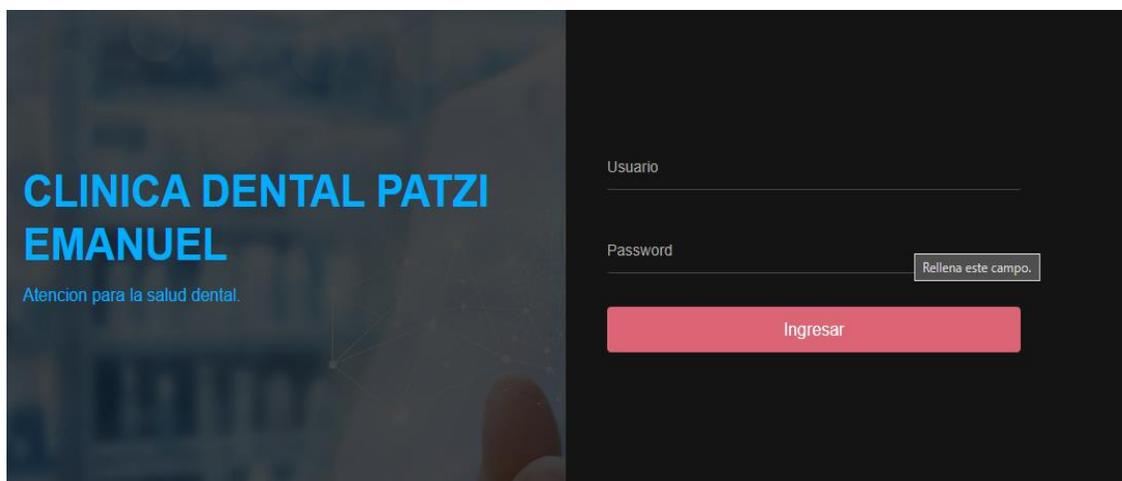


Figura 31. Prueba de caja negra: Inicio de sesión

Fuente: (Elaboración propia)

Tabla 61 - Prueba de caja negra: inicio sesión

Entrada de datos		Salida	Resultados
Usuario	Password		
isra	3608a6d1a05a ba23ea390e5f 3b48203dbb72 41f7	Ingresa un panel de bienvenida.	El sistema verifica la existencia del usuario y verifica si la clave o contraseña coinciden con los registros de la base de datos
isra	password	Regresa a la interfaz login	El sistema verifica la existencia del usuario y verifica si la clave o contraseña no coinciden con la base de datos lo regresa a interfaz

Fuente: (Elaboración propia)

La interfaz de sesión cumple con la función que tiene programada.

3.7.2.2. Prueba de caja negra ingresar paciente

The screenshot shows a web form for adding a new patient. The form is titled 'AGREGAR NUEVO PACIENTE' and is part of an 'ADMINISTRACION DE PACIENTES' system. It contains several input fields and a dropdown menu. The fields are: NRO C.I. (with a dropdown set to 'LA PAZ'), NOMBRE (Nombre), PATERNO (paterno), MATERNO (materno), TELEFONO (telefono), FECH. NAC. (dd/mm/aaaa), OCUPACION (ocupacion), LOCALIDAD (localidad), EDAD (edad), and DOMICILIO (domicilio). There are also radio buttons for 'FEMENINO' and 'MASCULINO'. At the bottom right, there are 'CANCELAR' and 'GUARDAR DATOS' buttons. The background shows a blurred view of the patient's information, including name, address, and phone number.

Figura 32. Prueba de caja negra: Registro de paciente

Fuente: (Elaboración propia)

Tabla 62 - Prueba de caja negra: Registro de paciente

ENTRADA	ETRADA VALIDA	RESULTADO
Nro. C.I.	Campo tipo numero	5943523
Departamento	Cadena de texto	La Paz
Nombre	Cadena de texto	Roberto Antonio
Apellido Paterno	Cadena de texto	Catacora
Apellido Materno	Cadena de texto	Arteaga
Teléfono	Cadena tipo numero	62034323
Fecha de nacimiento	Campo tipo date	28/02/1992
Ocupación	Cadena de texto	Estudiante
Localidad	Cadena de texto	Nuestra señora de La Paz
Edad	Campo tipo numero	28
Domicilio	Cadena tipo alfa numérico	Zona La Portada Av, Alcides Arquedas #80
Sexo	Cadena de texto	Masculino

Fuente: (Elaboración propia)

La prueba de caja negra evidencia que el formulario cumple con la función programada de registros de pacientes.

3.7.3. Pruebas de Estrés

Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga para ver los límites que pueda soportar. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema. Esto ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

Para esta prueba se utilizó el software JMeter¹⁸.

Se puso a prueba la interfaz paciente ya que es un modulo con más flujo de datos.

The screenshot shows a web application interface for adding a new patient. The form includes the following fields:

- NRO C.I.:** A text input field.
- LA PAZ:** A dropdown menu.
- NOMBRE:** A text input field with the placeholder 'Nombre'.
- PATERNO:** A text input field with the placeholder 'paterno'.
- MATERNO:** A text input field with the placeholder 'materno'.
- TELEFONO:** A text input field with the placeholder 'telefono'.
- FECH. NAC.:** A date input field with the placeholder 'dd/mm/aaaa'.
- OCUPACION:** A text input field with the placeholder 'ocupacion'.
- LOCALIDAD:** A text input field with the placeholder 'localidad'.
- EDAD:** A text input field with the placeholder 'edad'.
- DOMICILIO:** A text input field with the placeholder 'domicilio'.
- SEXO:** Radio buttons for 'FEMENINO' and 'MASCULINO'.

Below the form is a table of existing patients:

NRO C.I.	NOMBRE	EDAD	SEXO	OCUPACION	LOCALIDAD	TELEFONO	FECH. NAC.
34785 LP	ANA LIMACHI MAHANI	29	F	ESTUDIANTE	ZONA VILLADORES CALLE INGAVI #56	1234567	2020-08-15
34785 LP	SAKI MAHANI KANTUTA	29	F	ESTUDIANTE	ZONA LA PORTADA AV. ALDOREZ #42	1234567	2020-08-13
6951613 LP	ROMINA LOPEZ FORONDA	29	M	LICENCIADA	ZONA LA PORTADA AV. ALCIDES ARGUEDAS #42	1234567	2020-08-13
LP		0	M			0	2020-08-17

Figura 33. Prueba de estrés: formulario paciente

Fuente: (Elaboración propia)

Donde:

Muestra, es la cantidad de muestra para el caso de ejecución.

Media, es el tiempo promedio transcurrido de un conjunto de resultados.

Min, es el tiempo transcurrido más bajo para la muestra.

Max, es el tiempo transcurrido más largo para la muestra.

¹⁸ Meter es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el rendimiento de una variedad de servicios

Desviación Estándar, es la desviación estándar del tiempo transcurrido de la muestra.

Error %, señala el porcentaje de las solicitudes con errores.

Rendimiento, mide en solicitudes por segundo, minuto y hora.

Tabla 63 - Resultados obtenidos de la prueba de estrés

Nro.	Muestra	Media	Min	Max	Desviación Estándar	%Error	Rendimiento
1	70	1348	1	20778	7649,79	3%	10.8/sec
2	200	1878	1	46856	9345,25	8,5%	30,8/sec
3	500	2143	1	10345	8385,42	23,6%	62/sec
4	1000	2558	1	115671	7023,06	54,2%	116/sec

Fuente: (Elaboración propia)

3.8. ANALISIS DE CALIDAD

3.8.1. NORMA ISO 9126

Funcionalidad

La funcionabilidad no se puede medir directamente por esta razón corresponde derivar medidas directas como es el punto función que cuantifica el tamaño y la complejidad del sistema en términos de las funciones del usuario. Para la funcionalidad o medición del sistema, se debe determinar las siguientes características.

Tabla 64 - factores de ponderación

Valor de dominio de información	Conteo	Factor ponderado			Total
		Simple	Promedio	Complejo	
Entradas externas (EE)	30	3	4	6	120
Salidas externas (SE)	16	4	5	7	80
Consultas externas (CE)	7	3	4	6	28
Archivos lógicos internos (ALI)	12	7	10	15	120

Archivos de interfaz (AI)	0	5	7	10	0
Conteo total	348				

Fuente: Ingeniería de Software (R. Pressman, 2010, p. 532). Elaboración: Propia

Tabla 65 - Tabla de valores de ajuste de complejidad

Factor de complejidad	Valoración del factor						Total
	No influye	Incidental	Moderado	Medio	Significativo	Esencial	
¿Requiere el Sistema copias de Seguridad y de recuperación fiables?	0	1	2	3	4	5	5
¿Se requiere comunicación de datos?	0	1	2	3	4	5	4
¿Existen funciones de procesamiento distribuido?	0	1	2	3	4	5	4
¿Es crítico el rendimiento?	0	1	2	3	4	5	3
¿Será ejecutado el sistema en un entorno operativo existente y fuertemente utilizado?	0	1	2	3	4	5	3
¿Requiere el sistema entrada de datos interactiva?	0	1	2	3	4	5	4
¿Requiere la entrada de datos interactiva que las transacciones se lleven a cabo en múltiples pantallas?	0	1	2	3	4	5	3
¿Se actualizan los archivos maestros de forma interactiva?	0	1	2	3	4	5	4
¿Son complejas las entradas, las salidas, los archivos o las peticiones?	0	1	2	3	4	5	3
¿Es complejo el procesamiento interno?	0	1	2	3	4	5	3
¿Se ha diseñado el código para ser utilizado?	0	1	2	3	4	5	5
¿Están incluidas en el diseño la conversión y la instalación?	0	1	2	3	4	5	3
¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	0	1	2	3	4	5	3
¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente usada por el usuario?	0	1	2	3	4	5	5
Total							52

Fuente: Ingeniería de Software (R. Pressman, 2010, p. 533). Elaboración: Propia

Para calcular los puntos de función se utiliza la siguiente ecuación:

$$PT = \text{Conteo total} * (0,65 + 0,01 * \sum F_i)$$

Reemplazando los valores se tiene:

$$PT = 348 * (0,65 + 0,01 * 52) = 407,16$$

Observando la siguiente tabla, podemos observar que el sistema tiene una funcionalidad óptima, ya que el punto de función se encuentra en el intervalo mayor a 300.

Tabla 66 - Escala de punto función

Escala	Observación
PF > 300	Optimo
200 < PF < 300	Bueno
100 < PF < 200	Suficiente
PF < 100	Deficiente

Fuente y elaboración: Ingeniería de Software (R. Pressman, 2010).

Si calculamos al 100% el nivel de confianza considerando $\sum F_i = 70$ como el máximo valor de ajuste de complejidad entonces se tiene:

$$PT_{max} = \text{Conteo total} * (0,65 + 0,01 * \sum F_i)$$

Reemplazando valores:

$$PT_{max} = 348 * (0,65 + 0,01 * 70)$$

$$PT_{max} = 469,8$$

Con el máximo valor de ajuste de complejidad se tiene la funcionalidad real:

$$\text{Funcionalidad} = \frac{407,16}{469,8} = 0,86$$

Entonces la funcionalidad del sistema es de 86%, esto quiere decir que el sistema tiene 86% de probabilidad de funcionamiento sin riesgo de fallo y con operatividad constante y 14% de colapso de sistema.

Usabilidad

La usabilidad es la facilidad de uso, esta métrica nos muestra el costo de aprender el producto, lo cual se calcula con la siguiente formula:

$$FU = \left(\frac{\sum x_i}{n} \right) * 100$$

Tabla 67 - Ajuste de preguntas

Preguntas		Respuestas		Ponderación
		Si	No	
1	¿Puede utilizar con facilidad el sistema?	2	1	0.67
2	¿Puede controlar operaciones que el sistema solicite?	2	1	0,67
3	¿El sistema puede ser operado?	3	0	1
4	¿El sistema permitió la retroalimentación de información?	2	1	0,67
5	¿El sistema cuenta con interfaz agradable a la vista?	3	0	1
6	¿La respuesta del sistema es satisfactoria?	2	1	0,67
7	¿Le parece complicadas las funciones del sistema?	1	2	0,67
8	¿Se hace difícil aprender a manejar el sistema?	2	1	0,67
9	¿El sistema facilita el trabajo?	3	0	1
10	¿Durante el uso del sistema se produjo errores?	1	2	0,67
Total				7,69

Fuente: Ingeniería de Software (R. Pressman, 2010). Elaboración: Propia

Reemplazando en la fórmula:

$$FU = \left(\frac{7,69}{10} \right) * 100 = 76,9$$

Existe un 76,9% de comprensión o entendimiento de los usuarios con respecto a la

capacidad del sistema.

Confiabilidad

La confiabilidad se define como la probabilidad de que un sistema, activo o componente lleve a cabo su función adecuadamente durante un periodo bajo condiciones operacionales previamente definidas y constantes.

Para calcular la confiabilidad del sistema se toma en cuenta el periodo de tiempo en el cual se ejecuta y se obtiene muestras, se utilizan las siguientes ecuaciones:

$$\text{Confiabilidad} = \frac{MTBF}{MTBF + MTTR} * 100$$

$$MTBF = \left(\frac{h_t}{p}\right) * 100$$

$$MTTR = \left(\frac{h_p}{p}\right) * 100$$

Donde:

MTBF, tiempo total que funciona el sistema sin fallar.

MTTR, tiempo total de reparación del sistema.

h_t, horas trabajadas o de marcha durante el periodo de evaluación.

h_p, horas de paro durante el periodo de evaluación.

p, numero de fallas durante la evaluación.

Calculando *MTBF* y *MTTR*:

$$MTBF = \left(\frac{360}{6}\right) * 100 = 6000$$

$$MTTR = \left(\frac{48}{6}\right) * 100 = 800$$

Reemplazando en la ecuación de la confiabilidad:

$$\text{Confiabilidad} = \frac{6000}{6000 + 800} * 100 = 88.2$$

En un periodo de prueba de 15 días (360 horas), el sistema muestra un índice de confiabilidad de 88%.

Mantenibilidad

El mantenimiento se desarrolla para mejorar el sistema en respuesta a los nuevos requerimientos que la empresa tenga y los reglamentos que está regida por la misma. El estándar IEE94 sugiere un índice de madurez del software (IMS) que proporciona un indicador en la estabilidad de un producto, se lo determina con la siguiente formula:

$$IMS = \frac{M_T - (F_a + F_c + F_d)}{M_T}$$

Donde:

M_T , número de módulos de la versión actual.

F_c , números de módulos en la versión actual que se han cambiado.

F_a , número de módulos en la versión actual que se han añadido.

F_d , número de módulos de la versión anterior que se han borrado en la versión actual.

Reemplazando los valores:

$$IMS = \frac{7 - (1 + 0 + 0)}{7} = 0,857$$

Con lo que se puede decir que el nuevo sistema tiene una estabilidad de 85% que indica la facilidad de mantenimiento, el 15% restante es el margen de error correspondiente a los cambios y modificaciones.

Portabilidad

La portabilidad es la capacidad que tiene el software para ser trasladado de un entorno a otro.

Para poder medir la portabilidad del sistema usaremos la siguiente fórmula que indica el grado de portabilidad que tiene un software:

$$GP = 1 - \left(\frac{ET}{RT}\right)$$

Donde:

ET : Es la medida de los recursos necesarios para llevar el sistema a otro entorno.

ER : Es la medida de los recursos necesarios para crear el sistema en el entorno residente.

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

Si $GP = 1$, la portabilidad es perfecta

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

Para llevar el sistema a otro entorno se necesita una memoria extraíble de 1Gb o más capacidad, para crear el sistema en el entorno residente se necesita inicialmente 1 servidor con un sistema operativo este puede ser cualquiera de las distribuciones de Linux y un servidor apache, el lenguaje de programación PHP, el gestor de datos MySQL los cuales deben estar instalados.

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

$$GP = 1 - \left(\frac{1}{5}\right) = 1 - 0,20 = 80\%$$

3.9. ANALISIS DE COSTOS

Entre los distintos métodos de estimación de costes de desarrollo de software, se eligió el modelo COCOMO (*CO*nstructive *CO*st *MO*del) desarrollado por Barry M. Boehm, el cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto.

Cálculo de KLDC¹⁹

$$LDC = 10154$$

$$KLDC = \frac{10154}{1000}$$

$$KLDC = 10,15$$

Entre los distintos métodos de estimación de costes de desarrollo de software, se eligió el modelo COCOMO (*CO*nstructive *CO*st *MO*del) desarrollado por Barry M. Boehm, el cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto.

Cálculo de KLDC

Dentro de los modos de desarrollo que define el método, se utilizará el **orgánico**, debido a que el proyecto tiene una complejidad intermedia, el tamaño es menor a 50 KLDC y las restricciones no son complejas. Además, COCOMO define tres modelos de estimación (básico, intermedio y avanzado), que difieren en la precisión de los resultados. Para el presente proyecto se aplicará el modelo intermedio, que incluye un conjunto de medidas subjetivas llamadas conductores de costes los cuales calculan los resultados con bastante precisión.

¹⁹ Kilo Líneas De Código, representa la cantidad de líneas de código creadas en el desarrollo del proyecto en miles.

Para los cálculos, los coeficientes que se utilizarán son los siguientes:

$$a = 2,40$$

$$b = 1,05$$

$$c = 2,50$$

$$d = 0,38$$

Cálculo del coeficiente FAE

Tabla 68 - Atributos y sus valores usados en el cálculo de FAE

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	E. alto
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04	1,10	

Fuente y elaboración: Propia

$$FAE = 1,15 * 1,00 * 1,00 * 1,11 * 1,00 * 1,07 * 1,00 * 0,82 * 0,86 * 0,95 * 0,82 * 0,91 * 1,04$$

$$FAE = 0,7101$$

Justificación de los valores

- Fiabilidad requerida del software, si se produce un fallo el registro de una paciente, puede ocasionar pérdidas de tiempo en la clínica dental (Valoración Alta).
- Tamaño de la base de datos, la base de datos del producto será de tipo estándar (Valoración Nominal).
- Complejidad, la aplicación realizará cálculos y procesos medianamente complejos (Valoración Nominal).
- Restricciones del tiempo de ejecución, en los requerimientos se exige alto rendimiento (Valoración Alta).
- Restricciones del almacenamiento principal, no hay restricciones al respecto (Valoración Nominal).
- Tiempo de respuesta del ordenador, deberá ser interactivo con el usuario (Valoración Alta).
- Capacidad de análisis, capacidad alta relativamente, debido a la experiencia en análisis en proyectos similares (Valoración Nominal).
- Experiencia en la aplicación, se tiene cierta experiencia en aplicaciones de esta envergadura (Valoración muy alta).
- Capacidad de los programadores, teóricamente deberá tenerse una capacidad muy alta por la experiencia en anteriores proyectos similares (Valoración Alta).

- Experiencia en el lenguaje de programación, es relativamente alta, dado que se controlan las nociones básicas y las propias del proyecto (Valoración Alta).
- Prácticas de programación modernas, se usarán prácticas de programación modernas (Valoración Muy Alta).
- Utilización de herramientas software, se usarán herramientas estándar de las cuales se tiene una experiencia alta (Valoración Alta).
- Restricciones de tiempo de desarrollo, existen límites de planificación bien definidos. (Valoración Alta).

Cálculo del esfuerzo del desarrollo

$$E = a * KLDC^b * FAE$$

$$E = 2,40 * 10,42^{1.05} * 0,7101$$

$$E = 19,42 \cong 19 \text{ personas/mes}$$

Cálculo del tiempo del desarrollo

$$T = c * E^d$$

$$T = 2,50 * 19,42^{0,38}$$

$$T = 7,71 \cong 7 \text{ meses}$$

Se requiere de por lo menos 10 meses para la elaboración de todas las etapas del proyecto, la fase de programación comprende alrededor de 3 meses.

Cálculo de la productividad

$$PR = \frac{LDC}{E}$$

$$PR = \frac{10154}{19,42}$$

$$PR = 522,86 \cong 523 \text{ LDC/personas mes}$$

Cálculo del personal promedio

$$P = \frac{E}{T}$$

$$P = \frac{19,42}{7}$$

$$P = 2,7 \text{ personas} \cong 3 \text{ personas}$$

Cálculo del costo del proyecto

$$\text{Costo} = P * \text{Salario} * T$$

$$\text{Costo} = 3 * 2500 * 3$$

$$\text{Costo} = 22,500 \text{ Bs.}$$

3.10. SEGURIDAD

3.10.1. Seguridad a Nivel de Base de Datos

Se hace uso del gestor de base de datos MySQL que proporciona estabilidad, confiabilidad y alto rendimiento ya que no existen reportes de caídas en varios años de operación, además brinda extensiones para distintas funcionalidades como ser encriptación de datos.

MySQL garantiza la integridad y consistencia de la base de datos mediante procedimientos almacenados. Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

3.10.2. Seguridad a Nivel Aplicación

Tomando en cuenta las recomendaciones más relevantes especificadas en la norma ISO/IEC 27002 con respecto a la presentación de las características de confidencialidad, integridad y disponibilidad de la información se incorporado las siguientes medidas de seguridad en el sistema

Tabla 69 - Seguridad a nivel de Aplicación

Recomendaciones ISO/IEC 27002	Medidas de seguridad incorporadas en el sistema
Control de accesos	Se implementó como elemento importante la autenticación de usuarios que costa de un nombre de usuario y una contraseña, el usuario deberá estar previamente autenticado para realizar cualquiera acción en el sistema de no ser así será restringido. Se implementó niveles de roles de usuario con permisos de acceso a la información.
Controles criptográficos	Se implementó la encriptación de la contraseña de los usuarios con el uso del algoritmo de cifrado SHA1.
Registro de actividad y supervisión	Se controla los registros de información mediante la validación de datos de entrada.
Seguridad en la red	El sistema está enfocado para ser usado en una intranet el cual ofrece mayor integración, consistencia y seguridad.

Fuente y elaboración: Propia

CAPITULO IV
CONCLUSIONES Y
RECOMENDACIONES

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

Habiendo realizado el análisis, diseño y construcción del sistema, se concluyó lo siguiente:

- Se logró determinar los requerimientos y las necesidades funcionales para el desarrollo del sistema mediante las historias de usuario logrando de esta manera trabajar con el cliente de forma continua ante posibles cambios.
- Se logró diseñar el sistema Web con una interfaz amigable lo cual facilitó en gran manera al usuario final el cual pueda entenderlo rápidamente.
- El manejo de la información relacionada a los pacientes ahora se realiza con mayor velocidad, precisión y confiabilidad.
- Se elaboró la documentación necesaria para el uso y administración del sistema el cual sirve como medio de referencia a los usuarios del sistema.

4.2. RECOMENDACIONES

En base al proceso de desarrollo del sistema se observaron las siguientes recomendaciones que serán de gran utilidad las cuales son:

- Se recomienda crear normas y políticas de uso del sistema enfocándose en darle un buen uso.
- Realizar copias de seguridad de la base de datos, con el fin de evitar pérdida de información.

- Para posteriores versiones del sistema se recomienda desarrollar un módulo de facturación ya que la misma no cuenta con dicho modulo, para mejorar el servicio a los clientes y tener mayor control documento.

BIBLIOGRAFIA

- Calderón González, N. G. (Octubre de 2010). Diseño De Un Cuestionario De Evaluación De La Competencia Docente Con Base En La Opinión De Los Alumnos. Ensenada, Mexico.
- Criales Candia, R. M. (2002 - 2003). Proyecto De Elaboración De Un Instrumento Para La Evaluación Del Desempeño Docente.
- Cruz Avila, M. (Mayo de 2007). Una Propuesta Para La Evaluación Del Profesorado Universitario. Barcelona, España.
- Fernandes, C. A. (26 de Octubre de 2000). Obtenido de UTM Universidad Tecnológica De La Mixteca:
<http://www.utm.mx/~caff/doc/EI%20Proceso%20Unificado%20Rational.pdf>
- García Vázquez, J. A., Román Leal, K. M., & Avila Avila, A. (s.f.). Evaluación Del Desempeño Docente A Través De Proyectos Integradores En Nivel Secundaria. Mexico.
- Gil Aros, C. (Septiembre de 2008). Obtenido de AVANCES Investigacion En Ingenieria: <http://www.unilibre.edu.co/revistaavances/index.php/8-publicaciones/121-rup-metodolog%C3%ADa-en-los-sistemas-y-aplicaciones-basadas-en-la-web.html>
- Guzmán Pérez, J. M. (s.f.). Sistema De Evaluación Del Desempeño Docente Orientado A Resultados De Aprendizaje.
- Lexivox. (22 de Octubre de 2015). Obtenido de Lexivox:
<http://www.lexivox.org/norms/BO-L-2115.xhtml#disclaimer>
- Lina Marcano, F. (2008). Modelo De Evaluación Del Desempeño Profesional Docente. Ciudad De La Habana, Cuba.
- Lopez Luzardo, E. P., & Parrales Mendez, O. L. (Marzo de 2015). Analisis E Implementacion De Un Sisma Distribuido De Encuestas Distribuido Por Medio De Un Servidor Web Nativo De Android. Guayaquil.
- Pressman, R. (2010). *Ingeniería de Software, Un enfoque práctico*. Mexico: McGraw-Hill.

Pressman, R. S. (2010). *Ingeniería Del Software (7ª ED.)*. Mexico D. F.: The McGraw-Hill.

Vásquez Castro, I. L., & Vera Canetaco, D. P. (s.f.). Análisis, Diseño e Implementación de una Aplicación Móvil para Evaluar el Proceso de Enseñanza Aprendizaje en el Aula de Clases usando Tecnología de Redes Inalámbricas WIFI y NFC.

ANEXOS

Clínica dental Patzí Emanuel

Manual de
Usuario

SISTEMA DE ADMINISTRACIÓN
DE CONSULTAS
MEDICAS DENTALES

INDICE

PAG.

1. INTRODUCCION	1
2. DESCRIPCIÓN DEL SISTEMA.....	1
3. INGRESO AL SISTEMA	2
4.REGISTRAR USUARIOS.....	4
5. REGISTRAR PACIENTES	7
6.ADMINISTRACION DE TRATAMIENTOS	9
7. HISTORIAL DEL PACIENTE	12
8. ADMINISTRACION DE PAGOS	14
8. AGENDAR CITAS.....	16

1. INTRODUCCION

El presente sistema, fue desarrollado con el objetivo mejorar el proceso de atención al paciente, de tal manera que sea posible administrar la información de manera rápida, segura y eficiente.



El presente manual detalla las funciones que realiza el sistema, sus resultados y la forma de operar las diferentes secciones y módulos.

2. DESCRIPCIÓN DEL SISTEMA

El sistema contiene los siguientes módulos:

- ❖ Administración de usuarios
- ❖ Pacientes
- ❖ Tratamiento
- ❖ Historial
- ❖ Pagos
- ❖ Citas
- ❖ Receta medica

Adicionalmente, el sistema cuenta con las características:

- ❖ Acceso autorizado solo a usuarios que dispongan de una cuenta.
- ❖ Agenda de eventos, que hace posible la creación de tareas.

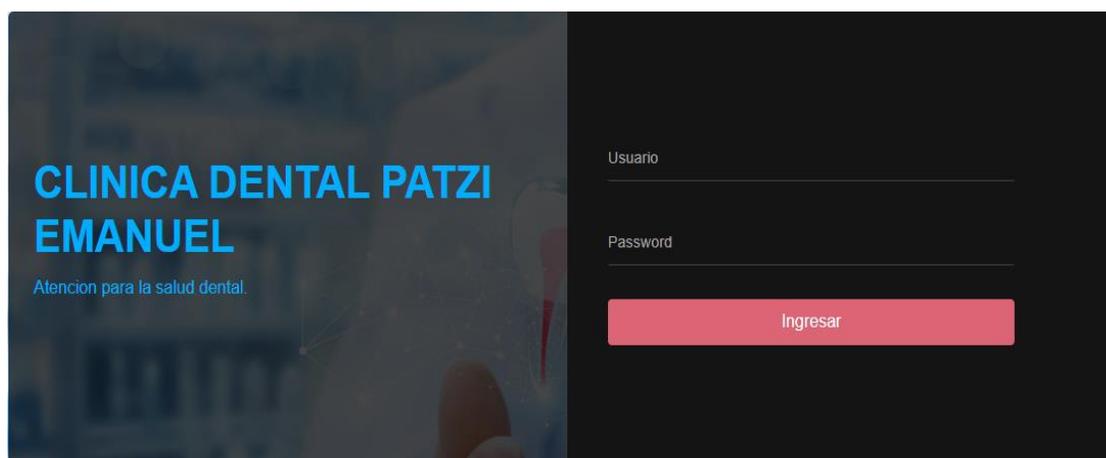
- ❖ Adición de médicos odontólogos
- ❖ Registro de pacientes
- ❖ Diseño adaptable, puede ser visualizado en la mayoría de los navegadores web.

3. INGRESO AL SISTEMA

Para ingresar al sistema asegúrese de cumplir los siguientes requisitos:

- ❖ Navegador Web (Internet Explorer 9.0+, Chrome 40+, Firefox 35.0+, Safari 3.4+ o Microsoft Edge 40.15063+). No se garantiza un correcto funcionamiento en otros navegadores.
- ❖ Disponer de un usuario y contraseña. Si no cuenta con una, contacte al administrador del sistema.

Seguidamente se vera la interfaz de ingreso:



Ingrese los datos requeridos y pulse el botón Ingresar.

SECCIONES DEL SISTEMA

The screenshot shows the 'Administración de Usuarios' page. The sidebar menu includes 'Administración Usuario', 'Administración Paciente', 'TRATAMIENTO', 'HISTORIAL', 'PAGOS', 'CITAS MEDICAS', and 'RECETA MEDICA'. The main content area has a '+ NUEVO USUARIO' button, a search bar, and a table of users.

CARNET	NOMBRE	CELULAR	TIPO DE USUARIO	ESTADO	IMAGE	ACCION
1234567 LP	ISRAEL IVAN MENDOZA APAZA	345345345	Odontologo	activo		
239487 LP	ERIKA APAZA LOPEZ	1345645545	Odontologo	inactivo		

Showing 1 to 2 of 2 entries

Navigation: Previous 1 Next

Menú Principal

Esta sección está disponible desde cualquier vista. Contiene el acceso a las secciones más importantes del sistema, algunos menús están agrupados, por ejemplo, en el menú Pacientes puede encontrar el acceso al listado de pacientes, registrar, búsqueda y editar.

Panel lateral de operaciones

Funciona dinámicamente, muestra las operaciones principales definidas para cada sección. Por ejemplo, en la sección de Operaciones del menú citas encontrará los eventos registrados.

Vista principal

Es el contenedor en donde se realizan todas las operaciones como ingreso y consulta de datos.

4. REGISTRAR USUARIOS

Esta sección comprende en el listado de los usuarios registrados, estos pueden crearse, editarse y eliminarse solo por el administrador de la clínica.

The screenshot displays the 'ADMINISTRACION DE USUARIOS' page. At the top, there is a search bar (4) and a 'NUEVO USUARIO' button (5). Below the search bar is a table with the following data:

CARNET	NOMBRE	CELULAR	TIPO DE USUARIO	ESTADO	IMAGE	ACCION
1234567 LP	ISRAEL IVAN MENDOZA APAZA	345345345	Odontologo	activo		
239487 LP	ERIKA APAZA LOPEZ	1345645545	Odontologo	inactivo		

At the bottom of the table, there are pagination controls: 'Showing 1 to 2 of 2 entries' and '← Previous 1 Next →'. Red arrows (1, 2, 3) point to the 'EDITAR', 'ACTIVAR', and 'DESACTIVAR' icons in the 'ACCION' column of the table.

1. Editar usuario, aquí puede editar la información del odontólogo.

The 'MODIFICAR USUARIO' form contains the following fields and values:

- NRO C.I.:** 239487, LA PAZ
- MATERNO:** LOPEZ
- CORREO:** erika@gmail.com
- IMAGEN ANTIGUO:**
- USUARIO:** erika
- NOMBRE:** ERIKA
- TELEFONO:** 1345645545
- ESPECIALIDAD:** ORTODONCIA
- TIPO USUARIO:** Odontologo
- DOMICILIO:** ZONA LA PORTADA CALLE INGAVI #45
- FECH. NAC.:** 12/11/1991
- IMAGEN NUEVO:** Elegir archivo (No se eligió ningún archivo)

At the bottom of the form, there are 'CANCELAR' and 'GUARDAR DATOS' buttons.

2. Activar/desactivar usuario, se usa para desactivar usuarios que no están activos.



3. Eliminar usuario, en casos especiales se requiere eliminación de un usuario.



4. buscar, se utiliza para realizar búsquedas de usuario ingresando datos como: cedula de identidad, nombre, apellidos, celular.



5. Agregar usuario, presionando este botón mostrara un formulario para registrar un nuevo usuario.

AGREGAR NUEVO USUARIO

NRO C.I.: **LA PAZ** ▼

NOMBRE:

PATERNO:

MATERNO:

TELEFONO/CELULAR:

FECH. NAC:

CORREO:

ESPECIALIDAD:

DOMICILIO:

TIPO USUARIO: ▼

IMAGEN: No se eligió ningún archivo

USUARIO:

CONTRASEÑA:

REPETIR CONTRASEÑA:

CANCELAR **GUARDAR DATOS**

6. Se utiliza la ver el listado con los datos de los usuarios.

USUARIO **ODONTOLOGO**

LISTADO DE ODONTOLOGOS

CARNET	NOMBRE	ESPECIALIDAD	CELULAR	CORREO	EDAD	DOMICILIO
1234567 LP	ISRAEL IVAN MENDOZA APAZA	NIÑOS	345345345	israel.32@hotmail.com	1991-07-09	ZONA VILLA MERCEDES
28356 LP	ROLLY PATZI ALANOCA	CIRUJIA	3456789	PATZI@GMAIL.COM	1991-07-29	ZONA BALLIVIAN AV ALFONFO UGARTE # 233
239487 LP	ERIKA APAZA LOPEZ	ORTODONCIA	1345645545	erika@gmail.com	1991-11-12	ZONA LA PORTADA CALLE INGAVI #45

5. REGISTRAR PACIENTES

Esta sección comprende en el listado de los pacientes registrados, estos pueden crearse, editarse y realizar su búsqueda.

ADMINISTRACION DE PACIENTES

+NUEVO PACIENTE

10 records per page

Search:

CARNET	NOMBRE PACIENTE	EDA	SEXO	OCUPACION	DOMICILIO	CELULAR	FECHA REGISTRO	PROCESOS
3456976 LP	ANA LIMACHI MAMANI	29	F	ESTUDIANTE	ZONA VILLADORES CALLE INGAVI #56	1234567	2020-08-15	
34765 LP	SAKI MAMANI KANTUTA	29	F	ESTUDIANTE	ZONA LA PORTADA AV ALCIDES ARGUEDAS # 42	43567890	2020-08-13	
6453456 LP	REINA CHOQUE LOPEZ	28	M	ESTUDIANTE	ZONA GERMAN BUSCH CALLE INGAVI #56	72344534	2020-08-17	
6951613 LP	ROMINA LOPEZ FORONDA	29	M	LICENCIADA	ZONA LA PORTADA AV. ALCIDES ARGUEDAS #42	1234567	2020-08-13	

Showing 1 to 4 of 4 entries

← Previous 1 Next →

1. Agregar paciente, presionando este botón mostrara un formulario para registrar un nuevo paciente.

2. buscar, se utiliza para realizar búsquedas de pacientes ingresando datos como: cedula de identidad, nombre, apellidos, celular.

10 records per page

Search: 3456976

CARNET	NOMBRE PACIENTE	EDAD	SEXO	OCUPACION	DOMICILIO	CELULAR	FECHA REGISTRO	PROCESOS
3456976 LP	ANA LIMACHI MAMANI	29	F	ESTUDIANTE	ZONA VILLADORES CALLE INGAVI #56	1234567	2020-08-15	

Showing 1 to 1 of 1 entries (filtered from 4 total entries)

← Previous 1 Next →

3. Editar paciente, aquí puede editar la información registrada del paciente.

6. ADMINISTRACION DE TRATAMIENTOS

La página principal de esta sección muestra una lista detallada de todos los pacientes registrados.

La creación, edición y búsqueda para ver el estado del tratamiento del paciente.

ADMINISTRACION DE TRATAMIENTOS

CARNET	NOMBRE COMPLETO	DIENTE	DIAGNOSTICO	PROCEDIMIENTO	ESTADO	fecha atencion	COSTO	
6951613 LP	ROMINA LOPEZ FORONDA	32	la pieza se necesita tratamiento de conductos lo y posterior tapado con porcelana	extraccion de nervios dentales	EN TRATAMIENTO	2020-08-12 00:00:00	250	
34765 LP	SÁKI MAMANI KANTUTA	39	la pieza debe realizar un tratamiento de con porlacna	limpieza	EN TRATAMIENTO	2020-08-12 00:00:00	200	
3456976 LP	ANA LIMACHI MAMANI	38	la pieza dental se encuentra con picadura	raspado	EN TRATAMIENTO	2020-08-14 00:00:00	100	

1. Agregar tratamiento, presionando este botón mostrara un formulario para registrar un nuevo para el paciente.

AGREGAR NUEVO TRATAMIENTO

ADMINISTRACION DE TRATAMIENTOS

SELECCIONA PACIENTE:
REINA CHOQUE LOPEZ / CARNET: 6453458 LP

DENTICIÓN PERMANENTE

DENTICIÓN TEMPORAL

SELECCIONE TRATAMIENTO: CIRUGIA

FECHA: dd/mm/aaaa

DIAGNOSTICO: DIAGNOSTICO ...

DETALLE DEL TRATAMIENTO:

PIEZA:

ESTADO DEL DIENTE:

PROCEDIMIENTO: CONSULTA ...

SELECCIONE MEDICO: ISRAEL IVAN MENDOZA APAZA

NOMBRE DE LA PIEZA:

COSTO:

ESTADO DEL TRATAMIENTO: EN TRATAMIENTO

RAYOS X: SI NO

Llenado los datos se tiene las opciones:

Cancelar: al presionar este botón se cancela el registro del tratamiento.

Guardar datos: al presionar este botón se guardan los datos ingresados.

Cerrar formulario: al presionarlo cierra el formulario

2. Editar tratamiento, aquí puede editar la información ingresada del tratamiento.

Cancelar: al presionar este botón se cancela las modificaciones realizadas.

Guardar datos: al presionar este botón se guardan los datos ingresados.

Cerrar formulario: al presionarlo cierra el formulario

3. Al apretar esta opción le dirige a la pantalla búsqueda de pacientes.

En esta pantalla la tenemos la opción buscar y ver.

CARNET	NOMBRE COMPLETO	CELULAR	PROCESOS
3456976 LP	ANA LIMACHI MAMANI	1234567	VER
34765 LP	SAKI MAMANI KANTUTA	43567890	VER
6453456 LP	REINA CHOQUE LOPEZ	72344534	VER
6951613 LP	ROMINA LOPEZ FORONDA	1234567	VER

Para realizar la búsqueda ingrese los siguientes datos: carnet, nombre, apellidos, y celular y la opción ver traslada a la pantalla historial de pacientes.

10 records per page

Search: romina

CARNET	NOMBRE COMPLETO	CELULAR	PROCESOS
6951613 LP	ROMINA LOPEZ FORONDA	1234567	Q VER

Showing 1 to 1 of 1 entries (filtered from 4 total entries)

← Previous 1 Next →

7. HISTORIAL DEL PACIENTE

BUSQUEDA PACIENTE INGRESOS ODONTOLOGOS

BUSQUEDA DE PACIENTES

10 records per page

Search:

CARNET	NOMBRE COMPLETO	CELULAR	PROCESOS
3456976 LP	ANA LIMACHI MAMANI	1234567	Q VER
34765 LP	SAKI MAMANI KANTUTA	43567890	Q VER
6453456 LP	REINA CHOQUE LOPEZ	72344534	Q VER
6951613 LP	ROMINA LOPEZ FORONDA	1234567	Q VER

Showing 1 to 4 of 4 entries

← Previous 1 Next →

1. Búsqueda, en esta opción se ingresa los siguientes datos: carnet nombre, apellidos y celular.

10 records per page

Search: 1234567

CARNET	NOMBRE COMPLETO	CELULAR	PROCESOS
3456976 LP	ANA LIMACHI MAMANI	1234567	Q VER
6951613 LP	ROMINA LOPEZ FORONDA	1234567	Q VER

Showing 1 to 2 of 2 entries (filtered from 4 total entries)

← Previous 1 Next →

2. ver, al apretar esta opción lo dirige a la pantalla historial de tratamientos

10 records per page

Search:

CARNET	NOMBRE	TRATAMIENTO	ESTADO	ODONTOLOGO	
3456976 LP	ANA LIMACHI MAMANI	RESTAURACIONES	EN TRATAMIENTO	ERIKA APAZA LOPEZ	Q VER DETALLE

Showing 1 to 1 of 1 entries

← Previous 1 Next →

3. ingresos, al presionar esta opción lo dirige a la pantalla ingresos donde se puede ver un resumen de los ingresos de los pacientes

Resumen

#pacientes	ingresos	pagos	deuda
3	550	150	400

fecha inicio y fecha fin se ingresa los datos para lanzar los detalles de la cantidad de pacientes atendidos en estas fechas.

Buscar. Al presionar esta opción manda el resultado de la cantidad de pacientes atendidos, la cantidad de ingresos de dinero, la cantidad de pagos dependiendo de las fechas ingresadas en los campos: fecha inicio y fecha fin.

Detalles, muestra a detalle los pacientes, el costo de sus tratamiento, la cantidad que deja y la deuda que tiene.

Resumen **Detalle**

Detalle

#pacientes	ingresos	pagos	deuda
ROMINA	250	100	150
SAKI	200	0	200
ANA	100	50	50
TOTAL	550	150	400

4. Al presionar esta opción ingresa a la pantalla odontólogo, donde al ingresar el número de carnet manda un listado del medico y la cantidad de pacientes que atendió detallando la fecha de atención, el nombre del paciente, el tratamiento y el procedimiento que le realizaron.

BUSQUEDA PACIENTE INGRESOS **ODONTOLOGOS**

CARNET DEL ODONTOLOGO: **BUSQUEDA** → 1234567 

Detalle

Resumen

ODONTOLOGO	FECHA ATENCIÓN	PACIENTE	TRATAMIENTO	PROCEDIMIENTO
ISRAEL IVAN MENDOZA	2020-08-12 00:00:00	ROMINA LOPEZ	PERIODONCIA	extraccion de nervios dentales

8. ADMINISTRACION DE PAGOS

En esta pantalla se lista a los pacientes el tratamiento

ADMINISTRACION DE PAGOS

Desde: 31/08/2020 Hasta: 31/08/2020 [Buscar] [Restablecer] [Restablecer]

10 records per page

CARNET	NOMBRE	FECHA DEL PAGO	TRATAMIENTO	DIENTE	ESTADO	COSTO	SALDO	ACCION
6951613 LP	ROMINA LOPEZ FORONDA	2020-08-24	PERIODONCIA	26	FINALIZADO	180	180	[PAGAR]
6951613 LP	ROMINA LOPEZ FORONDA	2020-08-24	OPERATORIA-ENDODONCIA	22	EN TRATAMIENTO	150	150	[PAGAR]
3456976 LP	ANA LIMACHI MAMANI	2020-08-25	RESTAURACIONES	38	EN TRATAMIENTO	100	0	[PAGAR]
6951613 LP	ROMINA LOPEZ FORONDA	2020-08-26	PERIODONCIA	32	EN TRATAMIENTO	230	80	[PAGAR]
34765 LP	SAKI MAMANI KANTUTA	2020-08-27	RESTAURACIONES	35	EN TRATAMIENTO	200	0	[PAGAR]
4568 LP	JUAN MAMANI QUISPE	2020-08-28	CIRUGIA	22	EN TRATAMIENTO	150	0	[PAGAR]

Showing 1 to 6 of 6 entries

1.se ingresa los datos de fecha inicial

2.se ingresa datos de la fecha final

3.una vez ingresado los datos se presiona buscar y mostrara el resultado de la búsqueda en base al rango de fechas ingresadas.

4.pagar, se presiona este botón y le mostrara el siguiente menú.

PAGOS DEL TRATAMIENTO

COSTO DEL TRATAMIENTO: 180 SALDO: 180 FECHA DE PAGO: dd/mm/aaaa MONTO:

[CANCELAR] [GUARDAR DATOS]

1.muestra la cantidad total del tratamiento.

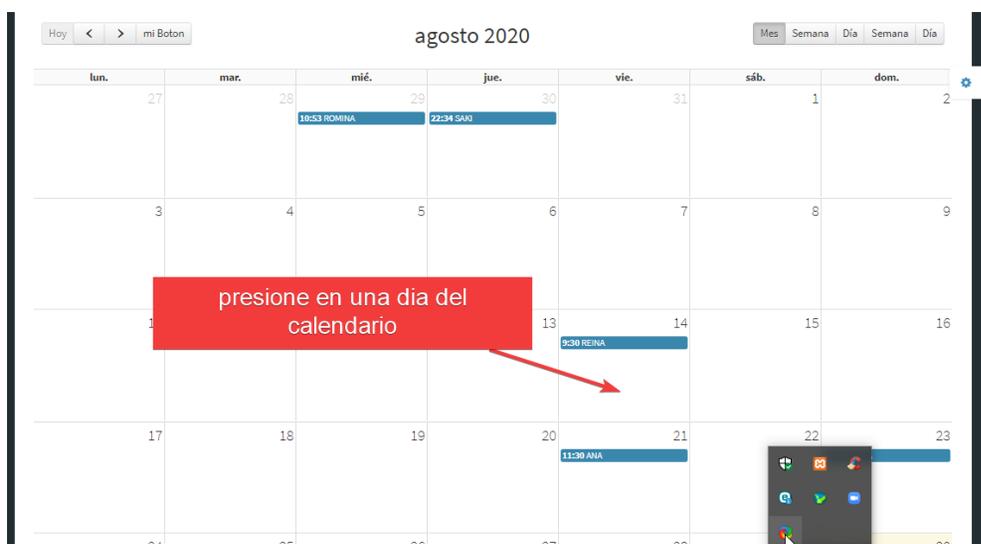
2.muestra el saldo que tiene a pagar.

3. ingresa la fecha del pago.

4.se ingresa el monto que deja.

Una vez ingresado los datos presiona el botón guardar.

8. AGENDAR CITAS



Para agendar una cita de click en una día que desea agendar, le lanzara el siguiente menú:

- 1.muestra el nombre del doctor.
- 2.muestra el nombre del paciente.

3.se ingresa el motivo de la siguiente consulta.

4.muestra la fecha de la cita.

5.ingresa la hora de la cita



Zona Ballivián El Alto Av. Alfonso Ugarte N° 200

El Alto – La Paz

Organizado por

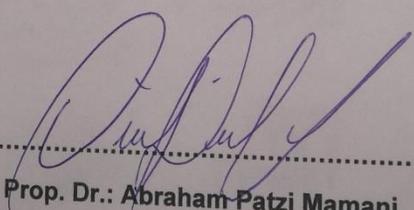
Yo, Abraham Patzi Mamani.

Como propietario y responsable de la clínica dental Patzi Emanuel.

Otorgo el presente Aval del proyecto de grado: **"SISTEMA WEB PARA LA ADMINISTRACIÓN DE CONSULTAS MÉDICAS DENTALES", CASO: CLINICA DENTAL PATZI EMANUEL.** Propuesto por Israel Ivan Mendoza Apaza con cedula de identidad 6951613 expedido en la ciudad de La Paz.

Siendo el proyecto de necesidad para la Clínica Dental, para una mejor funcionalidad.

Al pie firmo en fecha 23 de Octubre de 2019, dando mi consentimiento para la ejecución del proyecto, para fines del interesado.


.....
Prop. Dr.: Abraham Patzi Mamani

Dr. Abraham Patzi Mamani
CIRUJANO ODONTOLOGO
MAT. PROF. P-589



Zona Ballivián El Alto Av. Alfonso Ugarte N° 200

El Alto – La Paz

Yo, Abraham Patzi Mamani.

Como propietario y responsable de la clínica dental Patzi Emanuel.

Otorgo el presente Aval de conformidad del proyecto de grado: "**SISTEMA WEB PARA LA ADMINISTRACIÓN DE CONSULTAS MÉDICAS DENTALES**", CASO: **CLINICA DENTAL PATZI EMANUEL**. Propuesto y desarrollado por Israel Ivan Mendoza Apaza con cedula de identidad 6951613 expedido en la ciudad de La Paz.

Siendo que se verifico el cumplimiento de requerimiento y funcionalidad del proyecto cumpliendo las necesidades de la clínica.

Al pie firmo en fecha 24 de Agosto de 2020, dando mi conformidad de entrega del proyecto, para fines del interesado.

Prop. Dr.: Abraham Patzi Mamani

Dr. Abraham Patzi Mamani
CIRUJANO ODONTOLOGO
MAT. PROF. P-889

El Alto, Agosto del 2020

Señora:
Ing. Maricel Yarari Mamani
DOCENTE: TALLER DE LICENCIATURA II

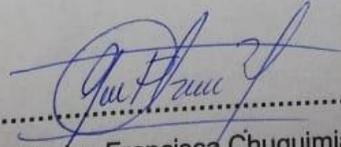
Ref. AVAL DE CONFORMIDAD

Distinguida Ingeniera:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado: "**SISTEMA WEB PARA LA ADMINISTRACIÓN DE CONSULTAS MÉDICAS DENTALES** ", **CASO: CLINICA DENTAL PATZI EMANUEL** que propone el postulante Univ.: Israel Ivan Mendoza Apaza con cédula de identidad 6951613 expedido en la ciudad de La Paz y Reg. Univ.: 11005164 para su defensa pública, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente:


.....
Lic. Gladys Francisca Chuquimia Mamani
CI 3397740 LP
Tutor Especialista

El Alto, Agosto del 2020

Señora:

Ing. Maricel Yarari Mamani

DOCENTE: TALLER DE LICENCIATURA II

Ref. AVAL DE CONFORMIDAD

Distinguida Ingeniera:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado: "**SISTEMA WEB PARA LA ADMINISTRACIÓN DE CONSULTAS MÉDICAS DENTALES**", **CASO: CLINICA DENTAL PATZI EMANUEL** que propone el postulante Univ.: Israel Ivan Mendoza Apaza con cédula de identidad 6951613 expedido en la ciudad de La Paz y Reg. Univ.: 11005164 para su defensa pública, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente:



.....
Freddy Salgueiro Trujillo
Tutor Revisor

El Alto, Agosto del 2020

Señor:
Ing. David Carlos Mamani Quispe
DIRECTOR DE CARRERA - INGENIERIA DE SISTEMAS

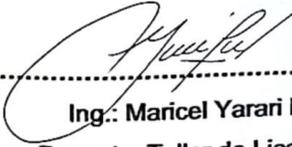
Ref. AVAL DE CONFORMIDAD

Distinguido Ingeniero:

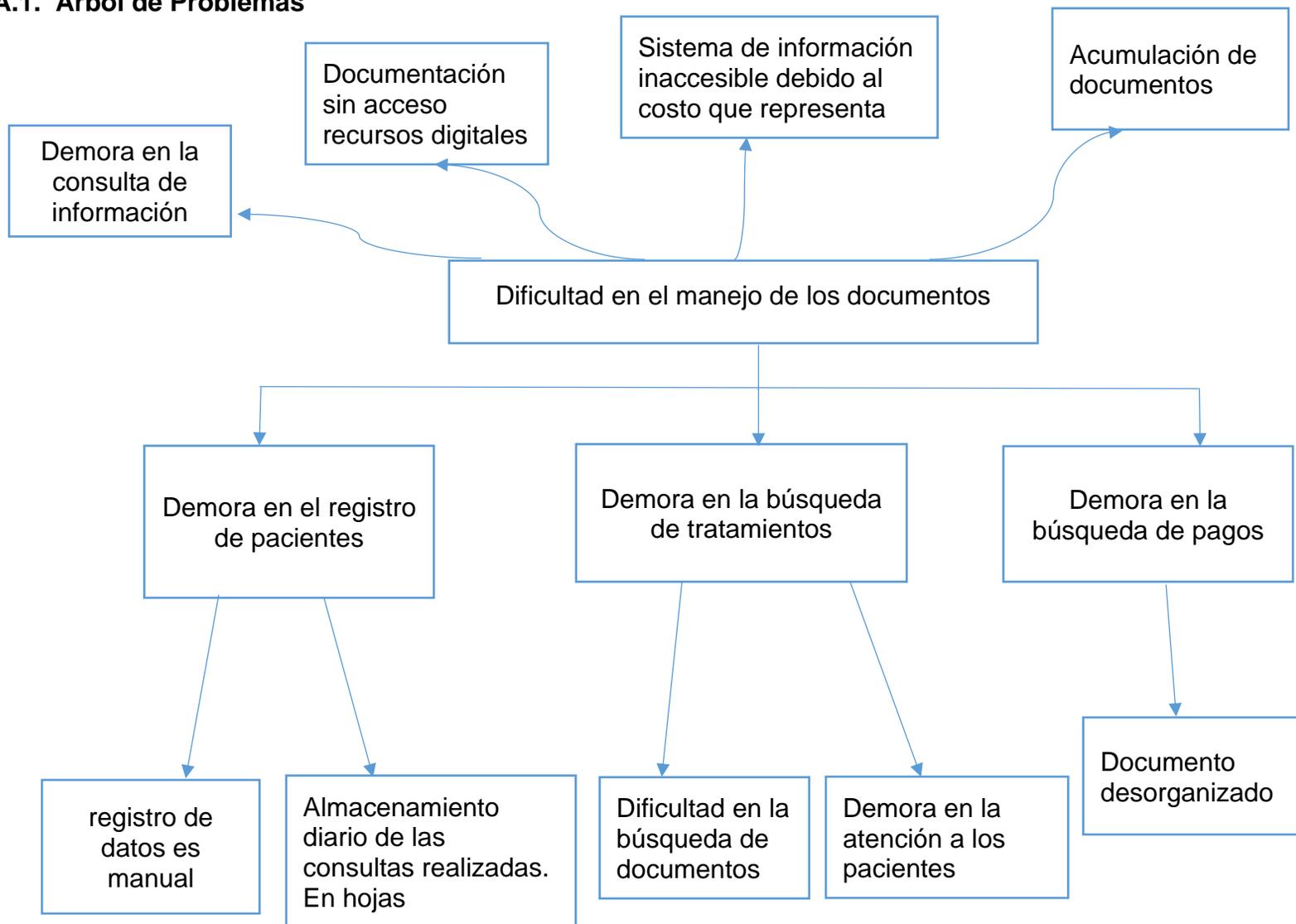
Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado: **"SISTEMA WEB PARA LA ADMINISTRACIÓN DE CONSULTAS MÉDICAS DENTALES "**, CASO: **CLINICA DENTAL PATZI EMANUEL** que propone el postulante Univ.: Israel Ivan Mendoza Apaza con cédula de identidad 6951613 expedido en la ciudad de La Paz y Reg. Univ.: 11005164 para su defensa pública, evaluación correspondiente a la materia Taller de Licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

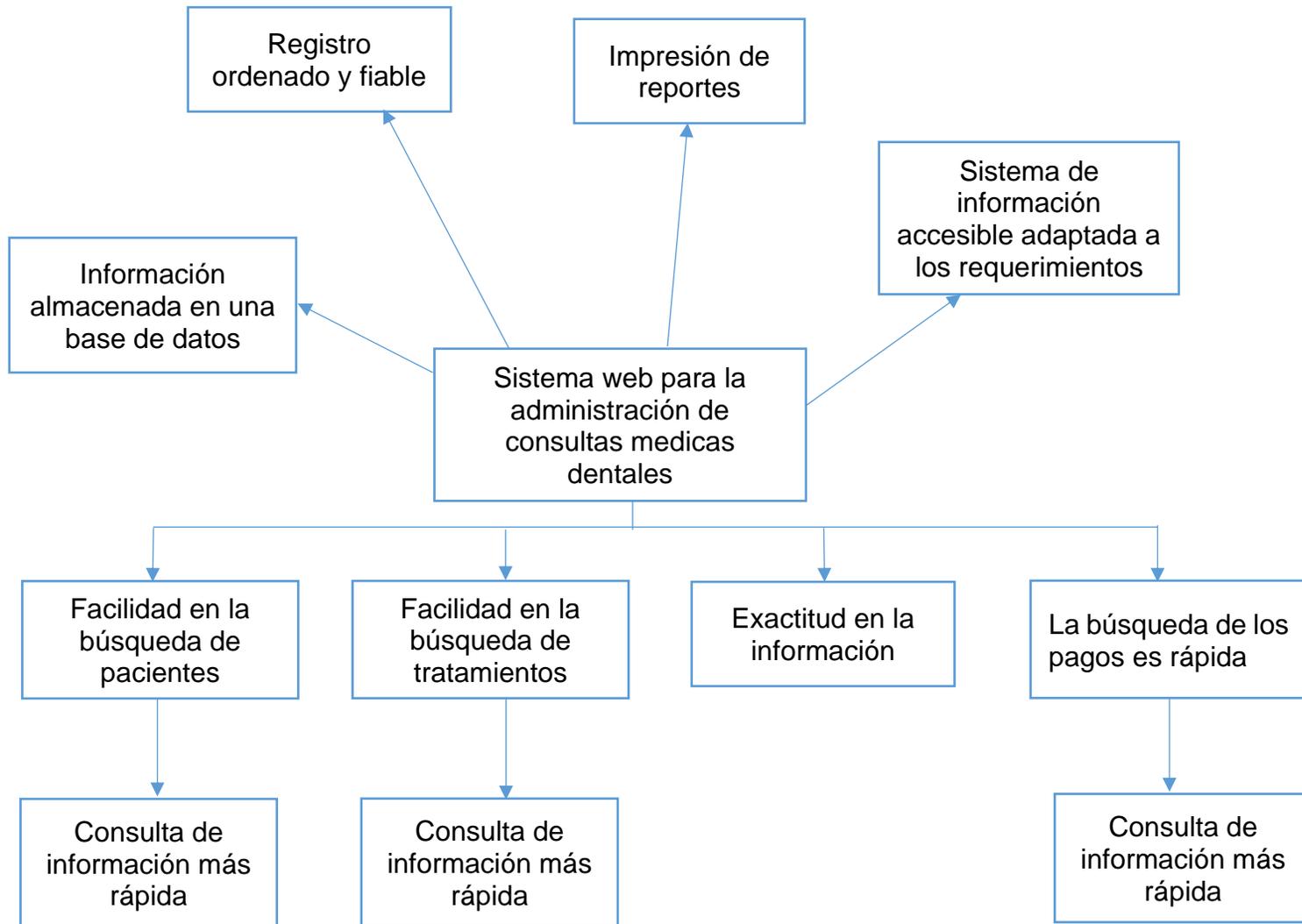
Atentamente:


.....
Ing.: Maricel Yarani Mamani
Docente: Taller de Licenciatura II

A.1. Árbol de Problemas



A.2. Árbol de Objetivos



**Captura de Requerimientos – Entrevista Realizada – Clínica Dental Patzi
Emanuel**

1. ¿Cuál es el nombre de la institución?
2. ¿Cuándo fue fundada la institución?
3. ¿Qué servicios ofrece la institución?
4. ¿Dónde está ubicado la institución?
5. ¿Describa el funcionamiento de su empresa?
6. ¿Describa los problemas más recurrentes que ocurren en los procesos registros?
7. ¿Cómo procede a la atención de un paciente?
8. ¿Cómo se procede a la búsqueda de un documento?