



**UNIVERSIDAD ESTATAL DE BOLÍVAR**  
**FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN**  
**EMPRESARIAL E INFORMÁTICA**

**CARRERA DE SOFTWARE**

**TRABAJO DE INTEGRACIÓN CURRICULAR**  
**PREVIO A LA OBTENCIÓN DEL TÍTULO DE**  
**INGENIERO EN SOFTWARE**

**FORMA: PROYECTO TECNOLÓGICO**

**TEMA:**

**SISTEMA WEB BASADO EN UNA METODOLOGÍA ÁGIL PARA GESTIONAR**  
**CITAS E HISTORIAS CLÍNICAS EN EL CENTRO ODONTOLÓGICO**  
**VELASTEGUÍ, CANTÓN ECHEANDÍA.**

**AUTORES:**

**JORGE ALEXANDER CÁRDENAS COLOMA**  
**LUIS ÁNGEL HERRERA CEDEÑO**

**DIRECTORA:**

**ING. GALUTH IRENE GARCÍA CAMACHO**  
**GUARANDA - ECUADOR**

**2024**

## **TEMA DEL PROYECTO TECNOLÓGICO**

**SISTEMA WEB BASADO EN UNA METODOLOGÍA ÁGIL PARA GESTIONAR CITAS E HISTORIAS CLÍNICAS EN EL CENTRO ODONTOLÓGICO VELASTEGUÍ, CANTÓN ECHEANDÍA.**

## **DEDICATORIA**

Este trabajo está dedicado, en primer lugar, a mis padres, Luis Calixto Herrera Camacho y María de los Ángeles Cedeño Guillen, quienes han sido pilares fundamentales en mi vida y a quienes debo gran parte de este logro. Su incansable dedicación, sacrificio y amor incondicional han sido la luz que ha guiado mi camino a lo largo de todo este proceso. A lo largo de los años, me han brindado no solo el apoyo material necesario, sino también la fortaleza emocional y moral para enfrentar cada obstáculo que la vida me ha presentado. Gracias a su ejemplo de responsabilidad, compromiso y perseverancia, he comprendido que no hay meta imposible cuando uno se entrega con pasión y esfuerzo. Este logro académico es tan suyo como mío, pues sin su apoyo inquebrantable, no habría sido posible.

A mis queridos hermanos, María Herrera y Carlos Herrera, quienes han sido mi constante fuente de apoyo y motivación, les agradezco profundamente. Su fe en mis capacidades ha sido un refugio en los momentos de duda, y su compañía me ha proporcionado el impulso necesario para seguir adelante, incluso cuando el camino parecía más difícil.

De manera especial, quiero dedicar este trabajo a mi abuela, Pola Guillen, quien desde el cielo continúa siendo una fuente de inspiración y fortaleza en mi vida. Aunque su presencia física ya no me acompaña, su ejemplo de vida y su legado permanecen duraderos en mi corazón y mente, guiando cada uno de mis pasos. Su amor y sabiduría siempre fueron un faro que iluminó mis decisiones, y este logro académico es también un homenaje a su memoria y al impacto que dejó en todos quienes tuvimos el privilegio de conocerla.

*Luis Herrera.*

¡Con mucho cariño y gratitud dedico el presente trabajo de titulación a Dios y a mi familia, porque siempre han sido, son y serán lo mejor que tengo en este mundo!

A mi madre, Ing. Marisol Coloma, que es una de las personas que más amo y respeto, la que desde que tengo uso de razón ha estado velando por mi bienestar e inculcándome a ser un buen hombre. De una y mil maneras ha sido un ejemplo a seguir, contribuyendo significativamente en mi camino y desarrollo profesional.

A mi esposa, Arq. Jessica Monar, quién nunca dudó de mí y siempre contribuyó de manera positiva en mi desempeño estudiantil. Una persona culta y llena de potencial que me hizo ver la vida de una manera diferente y me impulsó a seguir adelante tratando de ser mejor cada día. A mi hijo, Emir Cárdenas, alguien que aún no está consciente de la gran contribución que hizo para hacer posible este logro, pero a quién en el futuro tendré el orgullo de hacerle saber que fue una pieza fundamental para que su padre sea quien es ahora. El solo hecho de saber que pronto estaría en este mundo me llenaba de coraje y valentía para no desmoronarme en el camino y poder llegar a la meta, sin lugar a dudas fue el pilar fundamental para realizar este logro. Por él y para él.

A mi abuela, Elizabeth Ochoa y a mi tía, Lic. Lisbeth Coloma, porque desde la infancia fueron como mis segundas madres y siempre estuvieron a mi lado tratando de educarme de la mejor manera y esperando que algún día llegue a ser quien soy ahora.

*Jorge Cárdenas*

## **AGRADECIMIENTO**

Quiero expresar mi más sincero agradecimiento a todas las personas que han contribuido a la realización de este proyecto y a mi desarrollo personal y académico, a mi madre y a mi padre por su constante apoyo económico, su fe en mí y su sacrificio me han permitido perseguir mis sueños con determinación y confianza, gracias por ser mis guías y por siempre estar a mi lado en cada paso de este camino.

Quiero agradecer especialmente a mi directora de tesis, Ing. Galuth García. Aunque enfrenté un desafío al no aprobar una de sus materias, reconozco que esa experiencia fue fundamental para mi crecimiento académico. Gracias a ello, desarrollé un interés más profundo y un entendimiento sólido en programación y áreas relacionadas. Su enfoque riguroso y exigente ha sido una fuente de motivación para superarme y adquirir las habilidades necesarias para afrontar los desafíos de este proyecto.

Además, quiero extender mi más sincero agradecimiento al Ing. Danilo Barreno y al Ing. Christopher Espinosa, quienes han sido pares académicos esenciales y han sido fundamentales para enriquecer mi aprendizaje y comprensión de los conceptos técnicos relacionados con este proyecto. Gracias a su disposición para compartir su conocimiento y experiencia.

A mis amigos, Jorge Cárdenas, Sebastián García y Bryan Gaibor, quienes me ayudaron a que mi estancia en la ciudad de Guaranda fuera mucho más enriquecedora y amena. Su compañía y apoyo han hecho que esta etapa de mi vida sea aún más memorable, y estoy agradecido por cada momento compartido y cada conversación que me ha permitido crecer.

***Luis Herrera.***

No basta un simple agradecimiento a quienes han hecho posible este logro, porque sin lugar a duda todos ellos merecen mucho más de mi parte y me atrevo a decir que es lo que tendrán. Pero en esta ocasión quiero agradecerles y decirles que estoy comprometido a esforzarme para llenar de orgullo y satisfacción primeramente a Dios por darme la vida y ser mi guía en el transcurso de ella, por siempre darnos más de lo que merecemos y nunca negarnos su perdón ante las adversidades.

A mi familia, que con su profundo amor y sacrificio han logrado convertirme en la persona que soy. Agradezco cada acción y cada palabra que he tenido el honor de recibir, ya que es un privilegio que pocos tienen, el tener una familia llena de amor y ante todo fe en todos y cada uno de sus integrantes. Realmente estoy muy agradecido con su contribución por mínima que sea, pienso que hasta lo más mínimo es una grandeza si uno lo sabe valorar y retribuir de la mejor manera.

Quiero también extender mi más sincero agradecimiento a la Universidad Estatal de Bolívar, especialmente a la Carrera de Software y a todos los docentes y amistades que he tenido el gusto y honor de conocer. En todos estos años de estadía en esta noble institución me he encariñado muchísimo con el estilo de vida y evolución que proporciona el campus universitario, no hay duda que ha representado un segundo hogar para mí y estoy agradecido por todos los beneficios obtenidos.

Por último, también cabe mencionar mi grato agradecimiento al Consultorio Odontológico Velasteguí y todo su personal, por acogernos en sus instalaciones y siempre proporcionarnos de la mejor manera los requerimientos solicitados, no cabe la menor duda que ha sido una experiencia muy enriquecedora que ha fortalecido muchos aspectos de parte y parte.



*Jorge Cárdenas*

## CERTIFICADO DE VALIDACIÓN

ING.GALUTH GARCÍA, ING. DANILO BARRENO e ING. CHRISTOPHER ESPINOSA, en su orden Directora y Pares Académicos del Trabajo de Integración Curricular “SISTEMA WEB BASADO EN UNA METODOLOGÍA ÁGIL PARA GESTIONAR CITAS E HISTORIAS CLÍNICAS EN EL CENTRO ODONTOLÓGICO VELASTEGUÍ, CANTÓN ECHEANDÍA”, desarrollado por los estudiantes LUIS ÁNGEL HERRERA CEDEÑO y JORGE ALEXANDER CÁRDENAS COLOMA.

### CERTIFICAN

Que, luego de revisado el Trabajo de Integración Curricular en su totalidad, cumple con las exigencias académicas de la carrera SOFTWARE, por lo tanto, autorizamos su presentación y defensa.

Guaranda, 06 del 12 del 2024



GALUTH IRENE GARCIA  
GARCIA

ING. GALUTH GARCÍA

**Directora**



DANILO GIOVANNY  
BARRENO KARANJO

ING. DANILO BARRENO

**Par Académico**



CHRISTOPHER GABRIEL  
ESPINOSA RUIZ

ING. CHRISTOPHER ESPINOSA

**Par Académico**

**DERECHOS DE AUTOR**

Nosotros, **LUIS ÁNGEL HERRERA CEDEÑO** y **JORGE ALEXANDER CÁRDENAS COLOMA** portadores de la Cédula de Identidad No. **0705566537** y No. **0202233326** en calidad de autores y titulares de los derechos morales y patrimoniales del Trabajo de Titulación: **SISTEMA WEB BASADO EN UNA METODOLOGÍA ÁGIL PARA GESTIONAR CITAS E HISTORIAS CLÍNICAS EN EL CENTRO ODONTOLÓGICO VELASTEGUÍ, CANTÓN ECHEANDÍA**, modalidad Trabajo de Integración Curricular, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN, concedemos a favor de la Universidad Estatal de Bolívar, una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservo a mi favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo a la Universidad Estatal de Bolívar, para que realice la digitalización y publicación de este trabajo de titulación en el Repositorio Digital, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Los autores declaran que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

**LUIS ÁNGEL HERRERA CEDEÑO**  
C.I. 0705566537

**JORGE ALEXANDER CÁRDENAS COLOMA**  
C.I. 0202233326

## INDICE DE CONTENIDOS

TEMA DEL PROYECTO TECNOLÓGICO .....	ii
DEDICATORIA .....	ii
AGRADECIMIENTO .....	iv
CERTIFICADO DE VALIDACIÓN .....	v
DERECHOS DE AUTOR .....	vii
INDICE DE CONTENIDOS .....	viii
INDICE DE TABLAS .....	xii
INDICE DE FIGURAS.....	xiii
INTRODUCCIÓN .....	1
RESUMEN.....	3
ABSTRACT.....	4
CAPÍTULO I.....	5
FORMULACIÓN GENERAL DEL PROYECTO.....	5
1.1. TEMA .....	5
1.2. DESCRIPCIÓN DEL PROBLEMA.....	5
1.3. JUSTIFICACIÓN .....	5
1.4. OBJETIVOS .....	7
1.4.1. OBJETIVO GENERAL.....	7
1.4.2. OBJETIVO ESPECÍFICOS .....	7
CAPÍTULO II .....	8
MARCO TEÓRICO.....	8
2.1. ANTECEDENTES.....	8
2.2. CIENTÍFICO .....	9
2.2.1. ¿Qué es una metodología de desarrollo de software y para que se utiliza? .....	9
2.2.1. Metodologías ágiles .....	10
2.2.2. Selección de metodologías ágiles, por criterios de presencia .....	10
2.2.3. Selección de metodología, por criterios de conocimientos.....	10
2.2.4. Scrum .....	11

2.2.5. Cristal methodologies .....	11
2.2.6. Metodología extreme programming.....	12
2.3. CONCEPTUAL .....	12
2.3.1. Apache.....	12
2.3.2. Arquitectura de software .....	13
2.3.3. Api rest .....	13
2.3.4. Base de datos.....	13
2.3.5. Node.js .....	13
2.3.6. Angular.....	14
2.3.7. Frameworks.....	14
2.3.8. Gestor de Base de datos .....	14
2.3.9. Hosting .....	14
2.3.10. Laravel.....	15
2.3.11. Lenguaje de programación.....	15
2.3.12. Odontograma.....	16
2.3.13. Historia Clínica .....	16
2.3.14. Php .....	16
2.3.15. Stakeholders .....	17
2.3.16. WampServer.....	17
2.3.17. Insomnia.....	17
2.4. LEGAL.....	17
2.4.1. Ley de propiedad intelectual .....	17
2.4.2. Ley orgánica de protección de datos .....	18
2.4.3. Ley Orgánica del Sistema Nacional de Salud .....	19
2.4.4. Reglamento a la Ley Orgánica del Sistema Nacional de Salud .....	19
2.5. GEOREFERENCIAL .....	19
CAPITULO III .....	21
METODOLOGÍA .....	21
3.1. METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	21
3.1.1. Selección de la metodología ágil.....	21

3.1.2.	Revisión bibliográfica de las metodologías ágiles existentes. ....	21
3.1.3.	Metodologías ágiles existentes.....	23
3.1.4.	Programación Extrema (Extreme Programming, Xp).....	23
3.1.5.	Scrum .....	23
3.1.6.	Crystal Methodologies .....	24
3.1.7.	Kanban .....	25
3.1.8.	Lean.....	26
3.1.9.	Dynamic Systems Development Method (DSDM) .....	26
3.1.10.	Agile Unified Process (AUP).....	27
3.1.12.	Indicadores, métricas y criterios .....	29
3.2.	TÉCNICAS E INSTRUMENTOS DE RECOPIACIÓN DE DATOS.....	32
3.2.1.	Entrevistas .....	32
3.2.2.	Encuestas.....	32
CAPITULO IV .....		34
INGENIERÍA DEL PROYECTO.....		34
4.1.	Análisis.....	34
4.1.1.1.	Especificación de Requerimientos de Software.....	34
4.1.1.2.	Introducción .....	34
4.1.1.3.	Alcance del Producto .....	34
4.1.1.4.	Definiciones, Acrónicos y Abreviaturas. ....	35
4.1.1.5.	Referencias .....	36
4.1.2.	Descripción General del Producto .....	36
4.1.2.1.	Perspectiva del producto .....	36
4.1.2.2.	Funciones del producto .....	37
4.1.2.3.	Características de los usuarios .....	39
4.1.2.4.	Restricciones generales .....	39
4.1.3.	Requerimientos de Usuario .....	40
4.1.3.1.	Requerimientos funcionales .....	40
4.1.3.2.	Requerimientos no funcionales.....	41
4.1.3.3.	Requerimientos de interfaz .....	42

4.2.	Diseño .....	43
4.2.1.	Arquitectura del Software .....	43
4.2.2.	Diagramas .....	46
4.2.2.1.	Diagrama de Caso de Uso .....	46
4.2.2.2.	Diagramas de Secuencia .....	46
4.2.2.3.	Diagrama Entidad Relación .....	48
4.2.3.	Diseño de la Interfaz .....	48
4.2.3.1.	Patrón de navegación .....	49
4.2.3.2.	Interfaz de Pantallas .....	52
4.3.	Programación .....	56
4.3.1.	Definición del objetivo.....	56
4.3.2.	Análisis del problema.....	57
4.3.3.	Diseño del Algoritmo.....	57
4.3.4.	Codificación .....	60
4.4.	Pruebas .....	70
4.4.1.	Análisis de requisitos .....	70
4.4.2.	Planificación de pruebas .....	70
4.4.3.	Configuración del entorno de prueba.....	70
4.4.4.	Ejecución de las pruebas .....	71
4.4.5.	Pruebas Unitarias .....	71
4.4.6.	Pruebas de Integración .....	73
4.4.7.	Pruebas de Aceptación .....	75
4.4.8.	Prueba de cierre.....	76
	CONCLUSIONES .....	78
	RECOMENDACIONES .....	79
	BIBLIOGRAFÍA .....	80
	ANEXOS .....	82

**INDICE DE TABLAS**

Tabla 1. Características y componentes fundamentales de las metodologías ágiles.....	22
Tabla 2. Matriz comparativa .....	30
Tabla 3. Pruebas de Aceptación.....	75
Tabla 4. Ficha de Observación.....	77

**INDICE DE FIGURAS**

Figura 1.Georreferenciación .....	20
Figura 2.Georreferenciación (Calles).....	20
Figura 3.Arquitectura API REST .....	43
Figura 4.Diagrama de Casos de Uso .....	46
Figura 5.Diagrama de Secuencia: MANEJO DE LA INFORMACIÓN .....	47
Figura 6.Diagrama de Secuencia: REGISTRO DE DATOS .....	47
Figura 7.Diagrama de Secuencia: CITAS MÉDICA .....	48
Figura 8.Diagrama Entidad - Relación.....	48
Figura 9.Patrón de Navegación ADMINISTRADOR .....	49
Figura 10. Patrón de Navegación SECRETARIA .....	50
Figura 11. Patrón de Navegación DOCTOR.....	51
Figura 12. Patrón de Navegación PACIENTE.....	51
Figura 13. Paleta de Colores del Sistema.....	52
Figura 14. Logo del Sistema .....	53
Figura 15. Login.....	54
Figura 16. Panel Administrador .....	54
Figura 17. Panel Doctor .....	55
Figura 18.Panel secretaria .....	55
Figura 19. Panel Paciente.....	56
Figura 20. Cronograma pruebas .....	70
Figura 21. Registrar Paciente .....	71
Figura 22. Iniciar Sesión .....	72
Figura 23. Listado de Usuarios .....	72
Figura 24. Listado de Roles y Permisos.....	73
Figura 25. Servidor de Laravel.....	74
Figura 26. Servidor de Angular.....	74

## INTRODUCCIÓN

Los avances tecnológicos revolucionaron nuestro estilo de vida, la forma en que trabajábamos y cómo interactuábamos en la sociedad. Estas innovaciones tuvieron un impacto positivo, incrementando la eficiencia y precisión en los centros odontológicos, mejorando la gestión de citas e historias clínicas. A medida que avanzábamos, la tecnología continúa configurando nuestro futuro y planteando nuevas oportunidades y desafíos.

En el Centro Odontológico Velasteguí, el proceso de gestión de citas y registros clínicos se realizaba manualmente a través de redes sociales, mensajes de texto, llamadas telefónicas y agendamiento físico. Este sistema no satisfacía las necesidades de los pacientes, provocaba una desorganización en la coordinación del personal y afectaba el tiempo de espera y la eficiencia en la atención.

El sistema web implementado en el Centro Odontológico Velasteguí, permitió al personal administrativo registrar pacientes, gestionar el agendamiento y manejar las citas odontológicas de manera eficiente. Los odontólogos pueden visualizar las historias clínicas de los pacientes, así como agregar y actualizar diagnósticos y tratamientos. Por otra parte, los pacientes pueden registrarse en la aplicación, ver la disponibilidad de horarios de los médicos, y agendar o cancelar sus citas desde la comodidad de su hogar o cualquier lugar con conexión a internet.

Entre otras funcionalidades, el sistema contribuyó a disminuir las deficiencias en los procesos de atención a los pacientes del Centro Odontológico Velasteguí del cantón Echeandía, para el desarrollo del aplicativo se empleó la metodología ágil Extreme Programming (XP), la cual se ajustó a la lógica del negocio del Centro Odontológico Velasteguí, fue útil para identificar los riesgos principales y las expectativas de los involucrados, obteniendo como resultado la información y documentación necesaria para desarrollar el sistema de gestión de citas e historias clínicas. Con XP, se aseguró una colaboración continua entre el equipo de desarrollo, lo que facilitó la creación de un sistema que respondió adecuadamente a las necesidades del usuario.

El Capítulo I describe el problema que enfrenta el Centro Odontológico Velasteguí en su gestión de citas e historias clínicas; la justificación se centra en la necesidad de siempre mejorar de manera eficiente los procesos y mantener la calidad del servicio, utilizando la metodología ágil Extreme Programming (XP) que permitió el desarrollo eficaz del sistema. También se establecen los objetivos generales y específicos del proyecto.

En el Capítulo II, se explican los diferentes conceptos teóricos basados en autores relevantes, permitiendo un alto entendimiento del proceso de desarrollo del sistema. Además, se aborda la parte legal del proyecto, asegurando que se rija por las normativas correspondientes, y se presenta la ubicación del Centro Odontológico Velasteguí donde se implementó el sistema.

El Capítulo III detalla la metodología utilizada durante el desarrollo, así como los instrumentos de recolección de datos que permitieron establecer los requerimientos funcionales y no funcionales de la aplicación, tales como: la encuesta y la entrevista.

En el Capítulo IV se aplican las bases teóricas a la práctica, diferenciando a los usuarios del sistema, estableciendo los requerimientos necesarios para la aplicación, diseñando los diagramas según el tipo de arquitectura seleccionada y las vistas del sistema. Se explican brevemente los procesos de desarrollo de la aplicación y se presentan las interfaces de la misma.

## **RESUMEN**

El objetivo general de este trabajo fue desarrollar e implementar un sistema web para gestionar citas e historias clínicas en el Centro Odontológico Velasteguí, con el fin de mejorar la eficiencia en la atención el manejo de la información de los pacientes tales como la confusión en los horarios de citas, y tiempos de espera prolongados. La metodología aplicada fue Extreme Programming (XP), lo que facilitó una colaboración continua entre el equipo de desarrollo y los usuarios clave, adaptando el sistema a las necesidades específicas del centro. Durante la implementación, se creó una arquitectura basada en una API REST, utilizando Laravel para la lógica de negocio y MySQL como sistema de gestión de bases de datos. Angular se empleó para diseñar una interfaz de usuario moderna e intuitiva, mientras que los mecanismos de autenticación con JWT aseguraron la protección de los datos clínicos. Los resultados mostraron una mejora significativa en la organización de las citas y el acceso a las historias clínicas, con una disminución de errores y tiempos de espera. Los pacientes también experimentaron una mayor satisfacción al poder gestionar sus citas de manera autónoma. Finalmente, gracias a la implementación de este sistema web contribuyó a optimizar la gestión de la información en el Centro, elevando la calidad del servicio y la seguridad de los datos. Esta transición de un manejo manual a uno automatizado promete mejorar notablemente el control de las citas médicas, así como la gestión de los datos personales de los pacientes y otros usuarios involucrados.

**Palabras clave:** Citas Odontológicas, Historias Clínicas, Extreme Programming, API REST.

## **ABSTRACT**

The general objective of this work was to develop and implement a web system to manage appointments and clinical records at the Centro Odontológico Velasteguí, aiming to improve efficiency in patient care and information management, addressing issues such as appointment schedule conflicts and prolonged waiting times. The methodology applied was Extreme Programming (XP), which facilitated continuous collaboration between the development team and key users, adapting the system to the specific needs of the center. During implementation, an architecture based on a REST API was created, using Laravel for business logic and MySQL as the database management system. Angular was used to design a modern and intuitive user interface, while authentication mechanisms with JWT ensured the protection of clinical data. The results showed a significant improvement in the organization of appointments and access to clinical records, with a reduction in errors and waiting times. Patients also experienced greater satisfaction by being able to manage their appointments autonomously. Finally, the implementation of this web system contributed to optimizing information management at the Center, enhancing service quality and data security. This transition from manual to automated management promises to significantly improve the control of medical appointments, as well as the management of personal data for patients and other involved users.

**Keywords:**Dental Appointments, Medical Records, Extreme Programming, REST API.

## **CAPÍTULO I**

### **FORMULACIÓN GENERAL DEL PROYECTO**

#### **1.1.TEMA**

Sistema web basado en una metodología ágil para gestionar citas e historias clínicas en el centro odontológico Velasteguí, cantón Echeandía.

#### **1.2.DESCRIPCIÓN DEL PROBLEMA**

El Centro Odontológico Velasteguí, ubicado en el cantón Echeandía, fue creado el 6 de mayo del 2019 y desde entonces hasta la actualidad brinda sus servicios a la ciudadanía echeandiense. Hoy en día se encuentra enfrentando algunas complicaciones en los procesos de agendamiento de citas odontológicas, así como también en el seguimiento y manejo de historiales clínicos de los pacientes.

En lo que concierne a la reservación de citas, el proceso se lo realiza a través de redes sociales, mensajes de texto, llamadas telefónicas y agendamiento físico, lo que resulta poco eficiente en la calidad de servicio a los pacientes. Entre las dificultades que tiene el personal del centro Odontológico está la pérdida de información y el cruce de horarios de las citas lo que genera tiempos de espera prolongados por parte de los pacientes e insatisfacción con respecto a la calidad del servicio.

Por otro lado, la gestión del historial clínico de cada paciente se la realiza mediante una hoja de cálculo en Excel y uso de odontogramas en papel, lo que provoca desorganización y desactualización de la información contenida en esta, además de la dificultad en la búsqueda de información específica de los pacientes (tratamientos, diagnósticos, alergias, entre otros)

#### **1.3.JUSTIFICACIÓN**

Actualmente, los sistemas informáticos respaldan a las empresas al mejorar la eficiencia de los procesos y los recursos en todos los niveles de la organización.

A lo largo de los 5 años de servicio del Centro Odontológico Velasteguí, se ha experimentado un aumento de pacientes y por ende, también aumentó la cantidad de información que deben gestionar con respecto a los historiales clínicos y agendamiento de citas, procesos que actualmente se llevan de manera manual.

Al carecer de un sistema automatizado que optimice estos procesos y facilite el manejo de la información, se generan confusiones en los horarios de las citas, provocando molestias en los pacientes por los tiempos de espera prolongados. De igual manera se dificulta la búsqueda de información respectiva de los pacientes por parte del personal del centro odontológico, así como su organización y actualización oportuna.

El personal administrativo y de oficio del centro odontológico expresó en una entrevista la necesidad de adoptar un sistema informático que posibilite un procesamiento automatizado, ágil y eficiente de los datos de los pacientes y los procedimientos médicos, esto permitirá optimizar el tiempo y los recursos disponibles para brindar una atención de calidad. Por lo tanto, resulta adecuado desarrollar una aplicación web que permita un mejor control en el agendamiento de las citas médicas y llevar de manera sistemática y ordenada las historias clínicas de los pacientes.

De manera general la aplicación web permitirá al personal administrativo registrar pacientes, agendamiento y manejo de las citas odontológicas. Los odontólogos podrán visualizar la historia clínica del paciente, agregar y/o actualizar el diagnóstico o tratamiento. Por otra parte, el paciente podrá registrarse en la aplicación, ver la disponibilidad de horarios de los médicos, agendar o cancelar su cita desde la comodidad de su casa u otro lugar con conexión a internet. Entre otras funcionalidades que ayudarán a disminuir las deficiencias en los procesos de atención a los pacientes del Centro Odontológico Velasteguí del cantón Echeandía.

Este proyecto tecnológico aportará a la línea de investigación de la Carrera de Software denominada Ingeniería en Software, Redes y Telecomunicaciones, en la sub línea de Diseño e Implementación de sistemas de información.

## **1.4.OBJETIVOS**

### **1.4.1. OBJETIVO GENERAL**

Desarrollar un sistema web basado en una metodología ágil para la gestión de citas e historias clínicas en el Centro Odontológico Velasteguí, cantón Echeandía.

### **1.4.2. OBJETIVO ESPECÍFICOS**

- Seleccionar una metodología de desarrollo ágil de acuerdo con las características del proyecto.
- Identificar los requerimientos funcionales y no funcionales del sistema, tomando en cuenta las necesidades específicas de los pacientes y profesionales del centro odontológico.
- Diseñar la arquitectura y la interfaz del sistema acorde a los requerimientos ya definidos, asegurando un sistema intuitivo para los diferentes usuarios.
- Implementar el sistema web en la Clínica Odontológica Velastegui, brindando capacitación y soporte técnico necesario para una adaptación adecuada por parte del personal.

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1.ANTECEDENTES

En el contexto de la gestión de citas e historias clínicas en centros odontológicos, se han desarrollado varios proyectos previos que han abordado esta problemática. Por ejemplo:

“Desarrollo de una aplicación WEB para la gestión de Historias Clínicas y control de citas odontológicas para la Clínica "Dental House" en el sector de Conocoto usando la Metodología XP” (Freire, 2018)

En este proyecto se desarrolló una aplicación web que optimiza la gestión de historias clínicas y manejo de citas odontológicas de la clínica “Dental House”, con el objetivo de mejorar la atención a los pacientes y poder administrar de mejor manera la información.

Para el desarrollo de esta aplicación se hizo uso de la metodología de desarrollo de software “Extreme Programming” (XP), debido a su ligereza, eficiencia y flexibilidad. Además, con esta metodología se logró un desarrollo ágil y disciplinado, de tal forma que se logró seguir el progreso de la planificación conforme a los cambios.

“Desarrollo De Una Aplicación Web Para La Gestión De Historias Clínicas Y Control De Citas Odontológicas Para El Consultorio Dental Rodental 3” (Malavé, 2022)

En este proyecto de tesis se planteó desarrollar una aplicación web que permita la gestión de las historias clínicas y citas odontológicas en el consultorio “Rodental 3”, haciendo uso de una metodología de desarrollo de software incremental, debido principalmente a que gracias a sus incrementos se pudo identificar los distintos errores en la fase de pruebas.

“Aplicación web móvil para la gestión integral de la clínica odontológica “Braulio Josué Rojas Dental Studio” ubicado en la ciudad de Ibarra.” (Pinto, 2023)

Como estrategia para agilizar los diferentes servicios que ofrece la clínica dental al público en general, se propuso el desarrollo de una aplicación web móvil para la gestión integral de la clínica y de esta manera facilitar la programación de citas de clientes, así como la visualización de sus datos y respectivas historias clínicas.

Haciendo uso de la metodología de desarrollo de software “Extreme Programming” (XP), se obtuvo una mejor gestión de cada uno de los servicios prestados por parte de la clínica odontológica, dando un óptimo servicio a los pacientes.

Estos antecedentes demuestran la vital importancia del uso de la innovación tecnológica en la gestión eficiente de citas e historias clínicas en el campo odontológico. De la mano con estas tendencias, el presente estudio se enfoca en el desarrollo de un sistema web basado en una metodología ágil que permita una gestión óptima de citas e historias clínicas en el Centro Odontológico Velasteguí, brindando así una solución moderna, sistemática y eficaz, además de contribuir significativamente a la calidad de los servicios, experiencia y confort de los pacientes.

## **2.2. CIENTÍFICO**

### **2.2.1. ¿Qué es una metodología de desarrollo de software y para que se utiliza?**

La metodología se refiere a un grupo de procedimientos lógicos empleados para cumplir un propósito que demanda habilidades y conocimientos específicos. Constituye una fase clave dentro de un proyecto o trabajo, basada en un enfoque teórico que lleva a la elección de técnicas o métodos específicos para alcanzar los objetivos establecidos. Es el sistema de métodos empleados en una actividad particular con el fin de estructurarla y mejorar su eficiencia. Define las etapas a seguir y la forma de ejecutarlas para completar una tarea con éxito. (Pacienza, 2015)

### **2.2.1. Metodologías ágiles**

Las metodologías ágiles se entienden como un concepto general que abarca diversas prácticas de gestión, como Scrum, Kanban y Lean. Estas metodologías facilitan la evolución de requisitos y soluciones mediante la colaboración de equipos autoorganizados y multidisciplinarios. Según este enfoque, se fomenta la planificación flexible, el desarrollo progresivo, entregas tempranas y la mejora constante, estableciendo además una capacidad de respuesta ágil y adaptable hacia los clientes. (Denning, 2015)

### **2.2.2. Selección de metodologías ágiles, por criterios de presencia**

Los diseñadores de software suelen preferir metodologías que estén bien documentadas, ya que estas facilitan el acceso a la información necesaria. Asimismo, es importante que dichas metodologías cuenten con algún tipo de certificación y opciones de formación que las respalden. Considerando estos criterios, se han identificado seis categorías que permiten elegir la metodología más adecuada, basándose en su posición dentro de una evaluación acumulativa final. (Gómez, 2018)

#### **Las clasificaciones son:**

- La metodología con mayor presencia en Internet.
- La metodología mejor documentada.
- Metodologías certificadas y con formación.
- Metodologías con comunidades.
- Metodología más utilizada por empresas.
- Presencia empresarial

### **2.2.3. Selección de metodología, por criterios de conocimientos**

Así mismo dice que en función del grupo de trabajo o de diseño, se consideran los siguientes criterios en función de los conocimientos que el equipo de desarrollo tenga de las metodologías a evaluar. (Gómez, 2018)

**Estos criterios son:**

- Grado de conocimiento
- Soporte orientado a objetos
- Adaptable a cambios
- Basado en casos de uso
- Posee documentación adecuada
- Facilita la integración entre las etapas de desarrollo
- Relación con UML
- Permite desarrollo software sobre cualquier tecnología

Existen varias metodologías ágiles en la actualidad, entre las cuales es posible citar las siguientes:

**2.2.4. Scrum**

El marco de Scrum está compuesto por los Equipos Scrum, que incluyen roles, eventos, artefactos y reglas que los conectan. Cada elemento del marco tiene un propósito claro y es fundamental para garantizar el éxito de su implementación. Las reglas de Scrum regulan las relaciones e interacciones entre eventos, roles y artefactos, y están detalladas en este documento. Por otro lado, las estrategias para aplicar Scrum varían ampliamente y se desarrollan en otros recursos. (Sutherland, 2018)

**2.2.5. Cristal methodologies**

Las metodologías creadas por Alistair Cockburn se centran en las personas del equipo de desarrollo y buscan reducir al mínimo los artefactos generados, considerando el desarrollo de software como un proceso cooperativo de invención y comunicación limitado por los recursos disponibles. Dado que el equipo es clave, se promueve el fortalecimiento de sus habilidades y la implementación de políticas de trabajo

adaptadas al tamaño del grupo, como Crystal Clear para equipos pequeños (3 a 8 miembros) y Crystal Orange para equipos más grandes (25 a 50 integrantes). (Penadés, 2016)

### **2.2.6. Metodología extreme programming**

Extreme Programming (XP) es una metodología ágil que pone énfasis en las relaciones humanas como factor clave para el éxito en el desarrollo de software. Fomenta el trabajo en equipo, el aprendizaje continuo de los desarrolladores y un ambiente laboral positivo. XP se apoya en la retroalimentación constante entre el cliente y el equipo, una comunicación efectiva, la simplicidad en las soluciones y la valentía para adaptarse a los cambios. Es ideal para proyectos con requisitos poco definidos o en constante evolución, y con un alto nivel de riesgo técnico. Su nombre proviene de llevar al extremo principios y prácticas comunes. Kent Beck, creador de XP, expone su filosofía sin profundizar en detalles técnicos, tarea que ha sido desarrollada posteriormente en diversas publicaciones. A continuación, se describen sus elementos principales: historias de usuario, roles, procesos y prácticas. (Penadés, 2016)

## **2.3. CONCEPTUAL**

### **2.3.1. Apache**

El servidor web Apache, desarrollado por la Apache Software Foundation y lanzado en 1995, es un software de código abierto distribuido bajo licencia libre y ampliamente utilizado en Internet. Gran parte de los sitios web que visitamos diariamente funcionan gracias a este servidor, debido a diversas ventajas. Es gratuito, lo que elimina costos asociados a su uso; además, destaca por su robustez y alta seguridad frente a ataques en línea. Su facilidad de configuración es otra de sus fortalezas, ya que los ajustes necesarios se encuentran en un archivo de texto plano, lo que simplifica su manejo. (López, 2007)

### **2.3.2. Arquitectura de software**

La arquitectura no se refiere al software operativo, sino a una representación que permite evaluar la efectividad del diseño frente a los requisitos establecidos, explorar alternativas arquitectónicas en una etapa temprana donde los cambios aún son manejables y minimizar los riesgos relacionados con el desarrollo del software. Esta definición destaca la importancia de los "componentes del software" dentro de cualquier representación arquitectónica. Estos componentes pueden abarcar desde módulos de programa o clases orientadas a objetos hasta bases de datos y middleware que faciliten la configuración de redes cliente-servidor. (Pressman, 2010)

### **2.3.3. Api rest**

Es un conjunto de normas que permite a los desarrolladores crear servicios web que se comunican de forma eficiente mediante el protocolo HTTP. Utiliza operaciones estándar como obtener datos (GET), enviar datos (POST), actualizar datos (PUT) y eliminar datos (DELETE) a través de URLs específicas. Esto facilita la creación de aplicaciones web y móviles que pueden interactuar entre sí de manera flexible y escalable. (Fielding., 2010)

### **2.3.4. Base de datos**

Un sistema de bases de datos consiste en una serie de archivos interconectados y un conjunto de programas que facilitan a los usuarios el acceso y la modificación de estos archivos. Uno de sus principales objetivos es ofrecer a los usuarios una representación abstracta de los datos, es decir, el sistema oculta los detalles sobre cómo se almacenan y gestionan los datos. (Silberschatz , 2012)

### **2.3.5. Node.js**

Es un entorno de ejecución para JavaScript que permite ejecutar código JavaScript en el servidor. Los desarrolladores pueden utilizar JavaScript para crear aplicaciones del lado del servidor con alta eficiencia y escalabilidad, aprovechando el motor V8 de

Google, que convierte el código JavaScript en código de máquina. (Ethan Brown, 2019)

### **2.3.6. Angular**

Angular es un framework de desarrollo de aplicaciones web mantenido por Google que permite crear aplicaciones web dinámicas y de alto rendimiento proporciona una arquitectura clara y modular que facilita la gestión del código y la escalabilidad del proyecto. Angular también ofrece un potente sistema de datos y un sistema de inyección de dependencias, lo que permite gestionar eficientemente los servicios y otros recursos compartidos en la aplicación. (Ethan Brown, 2019)

### **2.3.7. Frameworks**

Es un conjunto de herramientas, directrices y estructuras predefinidas utilizadas para desarrollar y organizar software de manera eficiente. Actúa como una "base" o plataforma sobre la cual los programadores pueden construir y personalizar sus aplicaciones. Proporciona prácticas estandarizadas y componentes reutilizables que agilizan el proceso de desarrollo. (Meijomil, 2024)

### **2.3.8. Gestor de Base de datos**

Consiste en un conjunto de datos interconectados y un conjunto de programas para acceder a esos datos. La colección de datos, comúnmente conocida como base de datos, contiene información relevante para una organización. El objetivo principal de un SGBD es ofrecer una forma práctica y eficiente de almacenar y recuperar la información de la base de datos. Los sistemas de bases de datos están diseñados para manejar grandes volúmenes de datos, lo que implica tanto la definición de estructuras para almacenarlos como la provisión de herramientas para su manipulación. (a, 2012)

### **2.3.9. Hosting**

Según (Denning, 2015) El hosting, o alojamiento web, se refiere a los servicios proporcionados por empresas especializadas para almacenar y mantener los sitios

web de sus clientes en servidores conectados a Internet. Algunas de las principales funciones del hosting incluyen:

- Proporcionar el espacio en servidores para almacenar los archivos y datos del sitio web.
- Garantizar la disponibilidad y accesibilidad del sitio web las 24 horas del día, los 7 días de la semana.
- Brindar herramientas y recursos para la administración y gestión del sitio web.
- Asegurar la seguridad y protección de los datos del sitio web.

### **2.3.10. Laravel**

Laravel facilita la escritura de código en un entorno de bajo acoplamiento. La plataforma se enfoca en el desarrollo de clases con un único propósito, evitando la codificación en los módulos de nivel superior y permitiendo que los contenedores IoC manejen abstracciones específicas. Los módulos de nivel superior son independientes de los de nivel inferior, lo que hace que el proceso de programación sea más agradable (Suharjito, 2019)

### **2.3.11. Lenguaje de programación**

En su libro "Introducción a la programación con Java", (Dean & Dean Raymond, 2017) define un lenguaje de programación como "un lenguaje que utiliza palabras, sintaxis y puntuación especialmente definidas que las computadoras pueden entender". Estos lenguajes proporcionan una forma de comunicación entre los programadores y las computadoras, permitiéndoles escribir instrucciones y algoritmos que serán ejecutados por la máquina, cada lenguaje de programación tiene reglas específicas y una estructura definida que debe seguirse para que las instrucciones sean comprensibles y ejecutables por el sistema informático.

### **2.3.12. Odontograma**

El odontograma es una herramienta diseñada para identificar el estado de cada diente, lo que permite a los profesionales de la odontología realizar diagnósticos, establecer planes de tratamiento y valorar la evolución de los casos. Además, puede ser utilizado en investigaciones forenses y epidemiológicas. Su uso distingue a la odontología de otras áreas de la salud, ya que solo los odontólogos certificados están capacitados para implementarlo. Se sostiene que el odontograma se aprende simultáneamente en prácticas, como sostener formatos con las manos, ubicarse en coordenadas distintas a las habituales, escribir sobre pantallas e imaginar una letra dentro de un hueso. Finalmente, se ofrecen algunas reflexiones sobre la elaboración de odontogramas, considerando los conceptos de políticas ontológicas y afecto. (Daza-Cardona, 2021)

### **2.3.13. Historia Clínica**

La historia clínica en odontología es un documento esencial que registra de forma detallada y cronológica la información sobre la salud bucodental del paciente. Contiene antecedentes médicos y odontológicos, diagnósticos, tratamientos realizados y el seguimiento de la evolución del paciente, lo que permite al odontólogo tomar decisiones fundamentadas y proporcionar una atención continua y de calidad. (Ortún, 2017)

### **2.3.14. Php**

El lenguaje de programación PHP es robusto y cuenta con todo lo necesario para desarrollar sistemas web. Actualmente, se utiliza comúnmente en el desarrollo backend (lado del servidor). La versión 8 de PHP, lanzada el 26 de noviembre de 2020, introdujo características clave como el compilador JIT, tipos de unión y operadores NullSafe, siendo el compilador JIT la característica más destacada de esta versión (Engebret, 2010)

### **2.3.15. Stakeholders**

En el ámbito empresarial, los stakeholders son individuos o grupos que tienen un interés y un impacto en una organización y en los resultados de sus acciones. Entre los ejemplos más comunes de stakeholders se encuentran los empleados, accionistas, clientes, proveedores, gobiernos y comunidades. (Bello, 2021)

### **2.3.16. WampServer**

WampServer es un entorno de desarrollo web para Windows que facilita la creación y gestión de aplicaciones web locales. Integra Apache como servidor web, MySQL/MariaDB como sistema de gestión de bases de datos y PHP como lenguaje de programación. WampServer permite a los desarrolladores configurar rápidamente un entorno completo de desarrollo web en sus computadoras, lo que lo convierte en una herramienta ideal para el desarrollo y las pruebas de sitios y aplicaciones web. (Smith, 2021)

### **2.3.17. Insomnia**

Insomnia es una aplicación de escritorio utilizada por desarrolladores de software y profesionales de pruebas para facilitar la creación, envío y depuración de solicitudes HTTP y API. Es una herramienta que simplifica la interacción con servicios web y API, permitiendo a los desarrolladores probar y depurar de manera eficiente cómo sus aplicaciones se comunican con otros servicios en la web. (Inc, 2023)

## **2.4. LEGAL**

El desarrollo del proyecto tecnológico se fundamenta en las siguientes leyes:

### **2.4.1. Ley de propiedad intelectual**

#### **Sección V**

**Art. 28.-** Los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya

sea en forma legible por el hombre (código fuente) o en forma legible por máquina (código objeto), ya sean programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa. (p. 11)

**Art. 29.-** Es titular de un programa de ordenador, el productor, esto es la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se considerará titular, salvo prueba en contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual.

Dicho titular está además legitimado para ejercer en nombre propio los derechos morales sobre la obra, incluyendo la facultad para decidir sobre su divulgación. El productor tendrá el derecho exclusivo de realizar, autorizar o prohibir la realización de modificaciones o versiones sucesivas del programa, y de programas derivados del mismo. (p.11).

#### **2.4.2. Ley orgánica de protección de datos**

Ley Orgánica de Protección de Datos. Constitución de la República del Ecuador. (2008). Artículo 66, numeral 19. "Garantía y reconocimiento del derecho a la protección de datos de carácter personal, que incluye el acceso y la toma de decisiones sobre dicha información y datos y su adecuada protección. La recopilación, archivo, procesamiento, difusión o difusión de dichos datos personales requiere el permiso del titular o el permiso conforme a la ley."

Constitución de la República del Ecuador. (2008). Artículo 92. "Derecho de acceso a documentos, datos genéticos, archivos bancarios o de datos personales e informes relativos a sí mismo o a sus bienes en poder de entidades públicas o privadas en soporte físico o electrónico. Conocimiento del uso, finalidad, origen y destino de los datos personales. Autorización de divulgación de información archivada. Derecho de acceso gratuito, actualización, rectificación, supresión o cancelación de datos. Medidas de seguridad para datos sensibles. Posibilidad de recurso ante juez o magistrado. Derecho a indemnización para los afectados."

Constitución de la República del Ecuador. (2008). Artículo 385, literal 2. "Desarrollar tecnologías e innovaciones que impulsen la producción nacional, eleven la eficiencia y productividad, mejorar la calidad de vida, contribuir a la realización de una vida mejor."

### **2.4.3. Ley Orgánica del Sistema Nacional de Salud**

**Art. 11.-** “El Sistema establecerá los mecanismos para que las instituciones garanticen su operación en redes y aseguren la calidad, continuidad y complementariedad de la atención, para el cumplimiento del Plan Integral de Salud de acuerdo con las demandas y necesidades de la comunidad.”

Estos mecanismos incluyen:

d) “Un conjunto común de datos básicos para la historia clínica;”

**Art. 27.-** “El Ministerio de Salud Pública, con el apoyo del Consejo Nacional de Salud, implantará y mantendrá un sistema común de información sectorial que permitirá conocer la situación de salud, identificar los riesgos para la salud de las personas y el ambiente, dimensionar los recursos disponibles y la producción de los servicios para orientar las decisiones políticas y gerenciales en todos los niveles”

### **2.4.4. Reglamento a la Ley Orgánica del Sistema Nacional de Salud**

**Art. 77.-** De la Historia Clínica Única. - El Ministerio de Salud Pública, en su calidad de autoridad sanitaria, revisará y actualizará los formularios básicos y de especialidades de la historia clínica única para la atención de los usuarios, los mismos que serán posteriormente concertados y difundidos por el Consejo Nacional de Salud en todas las instituciones prestadoras de salud del sector público, semipúblico y privado.

## **2.5. GEOREFERENCIAL**

- **Cantón:** Echeandía
- **Provincia:** Bolívar
- **País:** Ecuador

El cantón Echeandía se encuentra ubicado en la región subtrópico de la provincia, en el centro-norte del país. Limita al norte con el cantón Las Naves, al sur el cantón Caluma, al este con la ciudad de Ventanas provincia de Los Ríos y al oeste con Guaranda. El Centro Odontológico Velasteguí está ubicado en el centro de este cantón, calle Simón Bolívar y Genaro Viscarra, a pocos pasos de la Unidad Educativa Adolfo Páez. Latitud y longitud (-1.4313127903495282, -79.27744575154888).

Figura 1. Georreferenciación



Elaborado por: Jorge Cárdenas - Luis Herrera  
Fuente: OpenStreetMap

Figura 2. Georreferenciación (Calles)



Elaborado por: Jorge Cárdenas - Luis Herrera  
Fuente: OpenStreetMap

## CAPITULO III

### METODOLOGÍA

#### 3.1.METODOLOGÍA DE DESARROLLO DE SOFTWARE

##### 3.1.1. Selección de la metodología ágil.

Las actividades que se llevarán a cabo en esta etapa son:

- Revisión bibliográfica de las metodologías ágiles existentes.
- Determinación de indicadores, métricas o criterios específicos para evaluar y comparar las distintas metodologías.

##### 3.1.2. Revisión bibliográfica de las metodologías ágiles existentes.

La adopción de metodologías ágiles en el desarrollo de software ha ganado un reconocimiento significativo en la última década debido a su capacidad para mejorar la flexibilidad y la calidad del producto final. Según (Beck, 2021), las metodologías ágiles se fundamentan en la colaboración continua y la entrega incremental de software, lo que permite a los equipos adaptarse de manera eficiente a los cambios en los requisitos del cliente. Además, (Sutherland S. y., 2017) sostienen que metodologías ágiles como Scrum facilitan la comunicación abierta y transparente entre desarrolladores y clientes, garantizando que el producto desarrollado cumpla con las expectativas y necesidades reales del usuario.

En el caso del proyecto de desarrollo del sistema web para el Centro Odontológico Velasteguí, estas características de las metodologías ágiles son particularmente relevantes. La necesidad de gestionar de manera eficiente las citas e historias clínicas requiere un sistema que no solo sea robusto y funcional, sino también flexible y capaz de adaptarse a las demandas cambiantes del entorno. A continuación, se detallan las características y componentes fundamentales de las metodologías ágiles, proporcionando una base teórica para la implementación exitosa de este proyecto.

Tabla 1. Características y componentes fundamentales de las metodologías ágiles

<b>Metodologías ágiles</b>	
<b>Hipótesis fundamental</b>	El software adaptativo de alta calidad es desarrollado por pequeños equipos que utilizan el principio de mejora continua del diseño y las pruebas basadas en una rápida respuesta y cambio.
<b>Estilo de gestión</b>	Liderazgo y colaboración
<b>Conocimiento administrativo</b>	Tácito
<b>Comunicación</b>	Formal
<b>Modelo de desarrollo</b>	Modelo evolutivo de entrega
<b>Estructura organizacional</b>	Orgánico (flexible y participativo,
<b>Control de calidad</b>	Control permanente de requisitos, diseño y soluciones. Pruebas permanentes.
<b>Requisitos de usuario</b>	Entrada interactiva
<b>Costo de reinicio</b>	Bajo
<b>Pruebas</b>	Cada iteración
<b>Participación del cliente</b>	Alto
<b>Desarrolladores</b>	Ágil, con conocimientos avanzados y cooperativos.
<b>Clientes</b>	Dedicado, informado, cooperativo, representativo y capacitado.

Elaborado por: Jorge Cárdenas – Luis Herrera

### 3.1.3. Metodologías ágiles existentes

### 3.1.4. Programación Extrema (Extreme Programming, Xp)

XP (Extreme Programming) es una metodología ágil enfocada en fortalecer las relaciones interpersonales como clave para el éxito en el desarrollo de software. Promueve el trabajo en equipo, se preocupa por el aprendizaje constante de los desarrolladores y fomenta un ambiente laboral positivo.

#### **Ventajas:**

- **Calidad del software:** Prácticas como TDD y refactorización continua mejoran la calidad del código.
- **Entrega frecuente:** Entregas frecuentes permiten recibir feedback constante del cliente.
- **Colaboración:** Promueve una estrecha colaboración entre los desarrolladores y los clientes.

#### **Desventajas:**

- **Requiere alta disciplina:** Necesita un equipo altamente disciplinado y comprometido.
- **Intensivo en recursos:** La programación en parejas y las pruebas continuas pueden ser intensivas en recursos y tiempo.
- **Complejidad inicial:** Implementar todas las prácticas de XP desde el inicio puede ser desafiante.

### 3.1.5. Scrum

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle, Scrum es una metodología ágil que define un marco para la gestión de proyectos y ha sido utilizada con éxito durante más de 10 años. Está especialmente indicada para proyectos con cambios rápidos en los requisitos.

### **Ventajas:**

- **Flexibilidad:** Permite adaptarse rápidamente a los cambios en los requisitos del cliente.
- **Transparencia:** Reuniones diarias (Scrum meetings) facilitan la comunicación constante.
- **Mejora continua:** Cada sprint concluye con una retrospectiva para identificar áreas de mejora.

### **Desventajas:**

- **Requiere compromiso:** Necesita una participación constante y comprometida de todos los miembros del equipo.
- **Estructura rígida de roles:** La rigidez en la definición de roles (Scrum Master, Product Owner, Development Team) puede ser una limitación en equipos pequeños.
- **Dependencia del Product Owner:** El éxito del proyecto puede depender en gran medida de la disponibilidad y la capacidad del Product Owner.

### **3.1.6. Crystal Methodologies**

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos

### **Ventajas:**

- **Adaptabilidad:** Diferentes versiones (Clear, Yellow, Orange, Red) permiten seleccionar la metodología más adecuada según el tamaño y la criticidad del proyecto.
- **Énfasis en la comunicación:** Fomenta la comunicación continua y cercana entre los miembros del equipo.

- **Flexibilidad en procesos:** No tiene un conjunto fijo de prácticas, lo que permite a los equipos adaptar las prácticas según sus necesidades.

#### **Desventajas:**

- **Falta de estandarización:** La falta de un conjunto fijo de prácticas puede llevar a inconsistencias en la implementación.
- **Dependencia de la experiencia:** Requiere equipos con experiencia para seleccionar y adaptar las prácticas adecuadas.
- **Difícil de escalar:** Puede ser más difícil de escalar en proyectos grandes debido a su naturaleza flexible.

#### **3.1.7. Kanban**

Kanban es una metodología ágil que utiliza un tablero visual para gestionar el flujo de trabajo. Fue desarrollado por Toyota como una forma de mejorar la eficiencia de su producción. La palabra "Kanban" significa "tarjeta visual" en japonés, y el método se basa en mover tarjetas a través de columnas que representan diferentes estados del trabajo, desde "Por hacer" hasta "Hecho".

#### **Ventajas:**

- **Transparencia:** Proporciona una visualización clara del flujo de trabajo y del estado de las tareas.
- **Flexibilidad:** Permite cambios en cualquier momento sin necesidad de seguir un ciclo de iteraciones fijas.
- **Mejora continua:** Fomenta la identificación y eliminación de cuellos de botella para mejorar el flujo de trabajo.

#### **Desventajas:**

- **Dependencia de la disciplina del equipo:** Requiere que el equipo se adhiera a la actualización constante del tablero.

- **Menos estructura:** La falta de un marco estructurado puede ser difícil para algunos equipos que prefieren más orientación.
- **Difícil de escalar:** La simplicidad del método puede no ser suficiente para gestionar proyectos muy grandes y complejos.

### 3.1.8. Lean

Lean es una metodología ágil basada en los principios de Lean Manufacturing, desarrollada por Toyota. Su objetivo es maximizar el valor entregado al cliente al mismo tiempo que se minimizan los desperdicios. Se centra en la eficiencia del proceso y la entrega rápida de valor.

#### Ventajas:

- **Eliminación de desperdicios:** Se enfoca en eliminar actividades que no añaden valor.
- **Entrega rápida:** Fomenta la entrega rápida de valor al cliente mediante ciclos de desarrollo más cortos.
- **Mejora continua:** Promueve una cultura de mejora continua y adaptación.

#### Desventajas:

- **Requiere un cambio cultural:** La implementación exitosa de Lean puede requerir un cambio significativo en la cultura de la organización.
- **Dependencia de la experiencia:** Necesita equipos con experiencia en identificar y eliminar desperdicios.
- **Difícil de implementar completamente:** Puede ser difícil implementar todos los principios de Lean en ciertas organizaciones o proyectos.

### 3.1.9. Dynamic Systems Development Method (DSDM)

DSDM es una metodología ágil que se centra en la entrega dentro de plazos y presupuestos fijos. Utiliza un ciclo de vida iterativo y proporciona un marco

estructurado para la planificación, ejecución y entrega de proyectos. Es uno de los métodos más antiguos y bien establecidos en el ámbito ágil.

**Ventajas:**

- **Estructura clara:** Proporciona un marco claro y detallado para la gestión de proyectos.
- **Control sobre plazos y presupuestos:** Enfatiza la entrega dentro de los límites de tiempo y presupuesto establecidos.
- **Flexibilidad en la entrega:** Permite ajustar los requisitos y la funcionalidad en cada iteración.

**Desventajas:**

- **Complejidad:** Puede ser más complejo de implementar debido a su estructura detallada y rigurosa.
- **Requiere compromiso del cliente:** Necesita una participación activa y continua del cliente a lo largo del proyecto.
- **Rígido en comparación con otros métodos ágiles:** Aunque flexible en la entrega, puede ser menos adaptable en términos de procesos y estructura.

**3.1.10. Agile Unified Process (AUP)**

AUP es una simplificación del Rational Unified Process (RUP) que aplica prácticas ágiles a un enfoque más estructurado y formal. AUP es una metodología iterativa y centrada en la arquitectura, que busca combinar la flexibilidad del desarrollo ágil con la disciplina de los procesos más formales.

**Ventajas:**

- **Equilibrio entre agilidad y estructura:** Combina la flexibilidad de los métodos ágiles con la disciplina de RUP.

- **Iterativo y centrado en la arquitectura:** Permite la adaptación continua y la mejora de la arquitectura del sistema.
- **Documentación suficiente:** Mantiene un nivel adecuado de documentación para garantizar la claridad y el control del proyecto.

**Desventajas:**

- **Complejidad:** Puede ser más complejo de implementar debido a la combinación de prácticas ágiles y formales.
- **Requiere experiencia:** Necesita equipos con experiencia para equilibrar adecuadamente agilidad y estructura.
- **Menos flexible:** Puede ser menos flexible en comparación con otras metodologías ágiles puras debido a sus raíces en RUP.

**3.1.11. Determinación de indicadores, métricas o criterios específicos para evaluar y comparar las distintas metodologías.**

Se eligió Extreme Programming (XP), Scrum y Crystal Methodologies para elaborar una tabla comparativa porque cada una ofrece elementos distintivos que son relevantes para el desarrollo de un sistema web ágil en el Centro Odontológico Velasteguí. XP se enfoca en mejorar la calidad del software mediante prácticas como TDD y refactorización continua, lo cual es esencial para mantener un sistema confiable y sin errores en un entorno clínico. Scrum aporta flexibilidad y estructura mediante sprints iterativos, permitiendo adaptarse rápidamente a los cambios y facilitando la entrega temprana y frecuente de incrementos funcionales, lo que asegura una retroalimentación constante del cliente. Crystal Methodologies, con su énfasis en la comunicación y su adaptabilidad según el tamaño del equipo, garantiza un ambiente de trabajo colaborativo y eficiente, proporcionando la flexibilidad necesaria para seleccionar la metodología más adecuada según las características específicas del proyecto. La combinación de estos enfoques permite crear una

comparación detallada que ayuda a identificar la metodología más adecuada para gestionar citas e historias clínicas de manera eficiente en el centro odontológico

### **3.1.12. Indicadores, métricas y criterios**

Para evaluar y comparar las distintas metodologías ágiles en el contexto del desarrollo del sistema web para la gestión de citas e historias clínicas en el Centro Odontológico Velasteguí, se deben considerar una serie de indicadores, métricas y criterios específicos. Estos factores ayudarán a determinar cuál metodología se adapta mejor a las necesidades del proyecto y del equipo de desarrollo. A continuación, se presentan los criterios más relevantes:

#### **Eficiencia del Equipo**

- Velocidad de entrega (historias de usuario/completadas por período de tiempo).
- Capacidad de adaptación del equipo a cambios en los requisitos del cliente.

#### **Calidad del Producto**

- Porcentaje de defectos encontrados durante las pruebas.
- Tiempo medio de resolución de problemas o bloqueos.

#### **Transparencia y Comunicación**

- Frecuencia y efectividad de las reuniones de equipo.
- Claridad en la comunicación de objetivos y requerimientos.

Tabla 2. Matriz comparativa

<b>Métricas / Metodologías</b>	<b>Extreme Programming (XP)</b>	<b>Scrum</b>	<b>Crystal Methodologies</b>
<b>Eficiencia del Equipo</b>	Alta (90%)	Alta (70%)	Moderada (60%)
	Capacidad de adaptación del equipo a cambios en los requisitos del cliente (90%).	Alta capacidad de adaptarse rápidamente a los cambios en los requisitos del cliente (70%).	Dependiente de la experiencia del equipo para seleccionar y adaptar las prácticas adecuadas (65%).
	Velocidad de entrega de historias de usuario/completadas por período de tiempo (90%)	Capacidad de adaptación del equipo a cambios en los requisitos del cliente (85%).	Énfasis en la comunicación continua y cercana entre los miembros del equipo (70%).
<b>Calidad del Producto</b>	Alta (90%)	Alta (70%)	Moderada (70%)
	Porcentaje de defectos encontrados durante las pruebas (90%).	Porcentaje de defectos encontrados durante las pruebas (70%).	Falta de estandarización que puede llevar a inconsistencias en la implementación (70%).
	Tiempo medio de resolución de problemas o bloqueos (90%).	Tiempo medio de resolución de problemas o bloqueos (75%).	Dependencia de la experiencia del equipo para seleccionar y adaptar las prácticas adecuadas (70%).
<b>Transparencia y Comunicación</b>	Alta (90%)	Alta (85%)	Alta (80%)
	Frecuencia y efectividad de las reuniones de equipo (95%).	Frecuencia y efectividad de las reuniones de equipo (90%).	Fomenta la comunicación continua y cercana entre los miembros del equipo (80%).
	Claridad en la comunicación de objetivos y requerimientos (90%).	Claridad en la comunicación de objetivos y requerimientos (85%).	No tiene un conjunto fijo de prácticas, lo que permite a los equipos adaptar las prácticas según sus necesidades (75%).

Fuente: Casierra Cavada, J., & Quiñonez

La metodología Extreme Programming (XP) emerge como la opción más sólida y adecuada. Los resultados muestran que XP lidera en prácticamente todas las métricas evaluadas, demostrando una eficiencia superior en la entrega de proyectos, una alta adaptabilidad a cambios en los requisitos del cliente, y una calidad del producto destacada. Además, XP obtiene el promedio general más alto entre las metodologías analizadas, lo que refuerza su posición como la elección más recomendada para el desarrollo del sistema web del Centro Odontológico Velastegui.

## **3.2. TÉCNICAS E INSTRUMENTOS DE RECOPIACIÓN DE DATOS**

### **3.2.1. Entrevistas**

Se llevaron a cabo entrevistas con el director de la clínica, los doctores y la secretaria (repcionista) recopilando información relevante para el estudio en cuestión. Durante estas entrevistas, se obtuvo una comprensión detallada de cómo se manejaban los procesos internos y se identificaron los medios de comunicación utilizados para las reservas. Las entrevistas permitieron descubrir las deficiencias del sistema manual, como la duplicación de citas, la falta de una base de datos y las dificultades en la coordinación del personal.

El director de la clínica expresó su preocupación por la ineficiencia del sistema anterior y su impacto en la satisfacción de los pacientes. Los doctores, por su parte, señalaron las dificultades que enfrentaban para acceder a las historias clínicas y actualizar la información de los pacientes en tiempo real. La secretaria destacó los desafíos diarios en la gestión de citas y la coordinación con los demás miembros del personal.

Además, se documentaron las distintas formas de comunicación utilizadas para las reservas, incluyendo redes sociales, mensajes de texto y llamadas telefónicas. Se identificaron problemas comunes, como la pérdida de información, la duplicación de registros y el tiempo excesivo dedicado a la administración de citas. Esta recopilación de datos fue fundamental para el desarrollo de un sistema que abordara las necesidades específicas del Centro Odontológico Velasteguí y mejorara significativamente la eficiencia y la calidad del servicio ofrecido a los pacientes.

### **3.2.2. Encuestas**

La encuesta es una técnica de recopilación de datos que permite obtener información de una muestra de individuos a través de preguntas estructuradas. En el contexto del Centro Odontológico Velasteguí, la encuesta fue utilizada con el objetivo de recopilar opiniones y percepciones de los pacientes sobre su experiencia en el proceso de agendamiento de citas, tiempo de respuesta, satisfacción con el servicio, y la implementación de nuevas tecnologías en la administración del centro.

El instrumento utilizado fue un cuestionario estructurado de siete preguntas dirigido a los pacientes del centro. Este cuestionario fue diseñado para explorar tanto aspectos cuantitativos como cualitativos, abordando temas como la frecuencia de las visitas, los métodos preferidos para agendar citas, la satisfacción con el tiempo de espera, y la importancia del uso de tecnologías automatizadas.

La elección de un cuestionario como instrumento de recopilación de datos responde a la necesidad de obtener información precisa y directa de los pacientes, en un formato que facilite el análisis estadístico y la interpretación de resultados. El cuestionario permitió a los pacientes expresar sus opiniones de manera clara y concisa, garantizando que se pudieran identificar tendencias y áreas de mejora.

### **Población y Muestra**

La población objetivo del estudio incluyó a todos los pacientes activos del Centro Odontológico Velasteguí, con un tamaño total estimado de 800 pacientes. Esta población fue seleccionada debido a su relevancia directa para el estudio, ya que se busca mejorar su experiencia en la interacción con el centro odontológico. Dado que estos pacientes son los principales usuarios de los servicios ofrecidos por el centro, su participación en la encuesta proporcionó información valiosa para evaluar la efectividad del sistema web desarrollado y su impacto en la satisfacción de los usuarios.

### **Para mayor información sobre el cálculo de la muestra ver anexo N°3**

En resumen, los resultados de la encuesta han puesto en evidencia la clara necesidad y aceptación de un sistema web para la gestión de citas e historiales clínicos en el Centro Odontológico Velasteguí. La implementación de este sistema ha respondido a la alta demanda de servicios continuos y a la preferencia por métodos de agendamiento más modernos. La nueva plataforma ha optimizado el tiempo de respuesta, mejorando significativamente la eficiencia y reduciendo los tiempos de espera. Además, ha abordado eficazmente las preocupaciones sobre la puntualidad y la gestión del tiempo. La integración de funciones de seguimiento y recordatorio mediante correo electrónico y WhatsApp ha facilitado una mejor comunicación con los pacientes.

### **Para mayor detalle de los resultados de la encuesta ver el anexo N°4**

## **CAPITULO IV**

### **INGENIERÍA DEL PROYECTO**

#### **4.1.Análisis**

##### **4.1.1.1.Especificación de Requerimientos de Software**

##### **4.1.1.2.Introducción**

En el presente documento se identificaron y analizaron los requerimientos para el desarrollo del sistema SIWCH-OV, el cual fue implementado en el Centro Odontológico Velasteguí en el cantón Echeandía, tomando como referencia la norma IEEE 830.

##### **4.1.1.3.Alcance del Producto**

El sistema SIWCH-OV fue desarrollado para cubrir de manera integral las necesidades de gestión de citas e historias clínicas en el Centro Odontológico Velasteguí, garantizando una administración eficiente y procesos ágiles de información. Este sistema incluye funcionalidades que permiten a distintos tipos de usuarios (administradores, doctores, secretarias y pacientes) interactuar de manera efectiva con la plataforma.

Los usuarios podrán iniciar sesión en el sistema de forma segura, asegurando que solo el personal autorizado y los pacientes registrados tengan acceso a la información pertinente. El registro de pacientes y doctores, así como la gestión de sus horarios y especialidades, permitirá una administración centralizada y eficiente de la información.

El sistema también facilita la visualización y gestión de listados de pacientes, doctores y especialidades, asegurando que todos los usuarios relevantes tengan acceso a la información actualizada y necesaria para la toma de decisiones.

La funcionalidad de gestión de citas médicas incluye el registro, listado, cancelación y re-agendamiento de citas, lo que permite una coordinación efectiva entre los pacientes y el personal del centro

odontológico. Además, se enviarán notificaciones a los usuarios para recordarles sus citas, minimizando la posibilidad de ausencias y optimizando la utilización del tiempo de los doctores.

En cuanto al manejo de historias clínicas, el sistema permite que los administradores, doctores y secretarías accedan y actualicen la información médica de los pacientes, asegurando que los registros sean completos y estén siempre disponibles para consultas futuras.

En resumen, el sistema SIWCH-OV cubre todos los aspectos necesarios para una gestión eficiente y efectiva de las citas e historias clínicas en el Centro Odontológico Velasteguí, mejorando la experiencia tanto para el personal como para los pacientes, y garantizando una operación más organizada y profesional del centro.

#### **4.1.1.4. Definiciones, Acrónimos y Abreviaturas.**

**Proceso:** Conjunto de pasos estructurados para alcanzar un objetivo.

**Roles:** Conjunto de permisos, derechos y funciones asignados a un usuario

**ERS:** Especificación de Requerimientos de Software.

**SIWCH-OV:** Sistema web para gestionar citas e historias clínicas" en el "Centro Odontológico Velasteguí".

**URL:** "Uniform Resource Locator", dirección de un recurso en internet, como una página web específica.

**CRUD:** Conjunto de operaciones básicas para manejar información en una base de datos: Crear (Create), Leer (Read), Actualizar (Update) y Eliminar (Delete).

**JASON:** Formato ligero de intercambio de datos, fácil de leer y escribir para los humanos, y sencillo de procesar para las máquinas.

**GET:** Método HTTP utilizado para solicitar datos del servidor.

**POST:** Método HTTP utilizado para enviar datos al servidor.

**DELETE:** Método HTTP utilizado para eliminar un recurso en el servidor.

**UPDATE:** Acción de modificar o actualizar la información existente en una base de datos

**HTTP:** Protocolo de transferencia de hipertexto

**API:** Interfaz de programación de aplicaciones, un conjunto de reglas que permite que diferentes aplicaciones se comuniquen entre sí.

**TDD:** “Test Driven Development”, Desarrollo Guiado por Pruebas.

#### **4.1.1.5.Referencias**

IEEE. (22 de octubre de 2008). Especificación de Requisitos según el estándar de IEE 830. IEEE Std. 830 - 1998. Obtenido de <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

### **4.1.2. Descripción General del Producto**

#### **4.1.2.1.Perspectiva del producto**

El sistema SIWCH-OV se implementó en el Centro Odontológico Velasteguí con el objetivo de optimizar la gestión de citas e historias clínicas de los pacientes.

Desde la perspectiva del personal administrativo, el sistema proporcionó una herramienta centralizada para manejar la información de los distintos usuarios que podría tener el Centro Odontológico, tales como secretarias, pacientes, doctores, entre otros. Permitiendo registrar y actualizar datos, programar horarios y gestionar especialidades de manera más eficiente.

Para los doctores, el sistema facilitó el acceso a las historias clínicas de los pacientes, permitiendo visualizar diagnósticos y tratamientos previos, y actualizar la información en tiempo real. Además, el sistema permite el manejo digital de la odontograma de

cada paciente. Esto ayudó a brindar una atención más informada y eficiente.

Desde la perspectiva de los pacientes, el sistema mejoró la accesibilidad y conveniencia, permitiéndoles gestionar sus citas en línea. Los pacientes pudieron iniciar sesión con su propio perfil para realizar distintas acciones como verificar cuales son los doctores y especialidades disponibles, además de agendar citas desde cualquier lugar con conexión a internet. Las notificaciones automáticas de citas contribuyeron a una mejor organización y puntualidad.

El sistema también mejoró la comunicación entre todas las partes involucradas, asegurando que tanto los pacientes como el personal estuvieran al tanto de las citas programadas y cualquier cambio en la agenda, lo que redujo el riesgo de ausencias.

En términos de impacto organizacional, la implementación del sistema SIWCH-OV ayudó al Centro Odontológico Velasteguí a modernizarse y mejorar la eficiencia en sus operaciones. El sistema permitió una mejor organización de los recursos internos y facilitó la coordinación entre el personal. Esto resultó en una percepción más positiva por parte de los pacientes respecto a la calidad del servicio.

#### **4.1.2.2. Funciones del producto**

El sistema SIWCH-OV incluye las siguientes funciones clave:

##### **1. Iniciar Sesión**

**Descripción:** Permite a los usuarios (administrador, doctor, secretaria) acceder al sistema mediante un proceso de autenticación seguro.

##### **2. Registro de Pacientes**

**Descripción:** Facilita la creación y almacenamiento de nuevos registros de pacientes, incluyendo información personal y médica.

##### **3. Listado de Pacientes**

**Descripción:** Proporciona una vista completa y filtrable de todos los pacientes registrados en el sistema.

#### **4. Registro de Doctores**

**Descripción:** Permite ingresar y gestionar la información de los doctores, incluyendo datos personales.

#### **5. Listado de Doctores**

**Descripción:** Ofrece una vista detallada de todos los doctores disponibles en el centro, accesible para administrador y secretaria.

#### **6. Registro de Horarios de los Doctores**

**Descripción:** Permite al personal administrativo y a las secretarias ingresar y actualizar los horarios de disponibilidad de los doctores.

#### **7. Registro de Especialidades**

**Descripción:** Facilita la entrada y gestión de las especialidades médicas ofrecidas por el centro.

#### **8. Listado de Especialidades**

**Descripción:** Muestra un listado de todas las especialidades disponibles en el centro, accesible para administradores, doctores y secretarias.

#### **9. Registro de Citas Médicas**

**Descripción:** Permite la programación de citas médicas, por el personal administrativo.

#### **10. Listado de Citas Médicas**

**Descripción:** Proporciona una vista detallada de todas las citas médicas programadas, accesible para administradores, doctores y secretarias.

#### **11. Cancelación de Citas Médicas**

**Descripción:** Permite la cancelación de citas médicas ya programadas, realizada por administradores, doctores y secretarias.

#### **12. Re-agendamiento de Citas Médicas**

**Descripción:** Facilita el cambio de la fecha y hora de las citas médicas previamente agendadas.

### **13. Notificación de Citas Médicas**

**Descripción:** Envía recordatorios y notificaciones sobre las citas médicas a todos los usuarios relevantes.

### **14. Manejo de Historias Clínicas**

**Descripción:** Permite a los administradores, doctores y secretarias acceder, actualizar y gestionar la información contenida en las historias clínicas de los pacientes, asegurando que toda la información médica esté disponible y actualizada.

#### **4.1.2.3. Características de los usuarios**

##### **Doctores**

**Conocimientos Básicos de Informática:** Los doctores tienen un nivel básico de habilidades informáticas, lo que les permite utilizar aplicaciones y sistemas para acceder a información médica y actualizar historias clínicas.

**Navegación:** Los doctores están familiarizados con la navegación básica en aplicaciones y sitios web. Pueden buscar información, visualizar horarios y acceder a las historias clínicas de manera eficiente.

##### **Secretaria**

**Conocimientos Básicos de Informática:** La secretaria posee un nivel básico a intermedio de conocimientos informáticos. Es capaz de utilizar software para gestionar datos, programar citas en el sistema.

**Navegación:** La secretaria tiene experiencia en la navegación de aplicaciones y plataformas en línea. Es capaz de realizar búsquedas, administrar registros, y manejar la programación de citas médicas con eficiencia.

#### **4.1.2.4. Restricciones generales**

##### **Conexión a Internet**

**Descripción:** La operación del sistema SWCH-OV requería una conexión a internet estable. La funcionalidad del sistema dependía del acceso continuo a la red para que los usuarios pudieran acceder y utilizar las herramientas disponibles, así como para la sincronización de datos y la comunicación en tiempo real.

#### **Pago del Alojamiento**

**Descripción:** La implementación y el funcionamiento del sistema estaban sujetos al pago del alojamiento del sitio web. Se necesitaba asegurar la cobertura financiera para mantener el servicio de hospedaje.

### **4.1.3. Requerimientos de Usuario**

#### **4.1.3.1. Requerimientos funcionales**

##### **Iniciar Sesión**

Usuarios: Administrador, doctor, secretaria, paciente.

##### **Registro de Pacientes**

Usuarios: Administrador, secretaria.

##### **Listado de Pacientes**

Usuarios: Administrador, doctor, secretaria.

##### **Registro de Doctores**

Usuarios: Administrador, secretaria.

##### **Listado de Doctores**

Usuarios: Administrador, secretaria, paciente.

##### **Registro de Horarios de los Doctores**

Usuarios: Administrador, secretaria.

##### **Registro de Especialidades**

Usuarios: Administrador, secretaria.

##### **Listado de Especialidades**

Usuarios: Administrador, doctor, secretaria, paciente.

### **Registro de Citas Médicas**

Usuarios: Administrador, secretaria, paciente.

### **Listado de Citas Médicas**

Usuarios: Administrador, doctor, secretaria, paciente.

### **Cancelación de Citas Médicas**

Usuarios: Administrador, doctor, secretaria.

### **Re-agendamiento de Citas Médicas**

Usuarios: Administrador, doctor, secretaria.

### **Notificación de Citas Médicas**

Usuarios: Administrador, doctor, secretaria, paciente.

### **Manejo de Historias Clínicas**

Usuarios: Administrador, doctor, secretaria.

## **4.1.3.2.Requerimientos no funcionales**

### **Rendimiento**

Descripción: El sistema SWCH-OV debía ofrecer tiempos de respuesta rápidos para todas las operaciones, incluyendo la gestión de citas, el acceso a historias clínicas y la actualización de datos. Se requería que el sistema pudiera manejar múltiples solicitudes simultáneas sin afectar su rendimiento.

### **Escalabilidad**

Descripción: El sistema debía ser escalable para adaptarse al crecimiento del Centro Odontológico Velasteguí. Esto incluía la capacidad de manejar un aumento en el número de usuarios, pacientes y registros sin comprometer la funcionalidad.

### **Seguridad**

Descripción: Se requería que el sistema implementara medidas de seguridad robustas para proteger la información sensible de los

pacientes y los datos del centro. Esto incluía el uso de cifrado para la transmisión de datos y la autenticación segura de usuarios.

### **Disponibilidad**

Descripción: El sistema debía estar disponible en todo momento durante las horas de operación del centro. Se necesitaban mecanismos para garantizar la continuidad del servicio y minimizar el tiempo de inactividad, incluyendo la realización de copias de seguridad periódicas.

### **Usabilidad**

Descripción: El sistema debía ser fácil de usar para todos los perfiles de usuario, con una interfaz intuitiva y accesible que permitiera realizar las tareas de manera eficiente. Se requería una curva de aprendizaje mínima para los usuarios.

### **Mantenimiento**

Descripción: El sistema debía permitir un mantenimiento sencillo y efectivo, con la capacidad de realizar actualizaciones y correcciones sin interrumpir el servicio. Se necesitaba un diseño modular para facilitar la gestión y la implementación de cambios.

#### **4.1.3.3.Requerimientos de interfaz**

##### **Interfaz de Usuario Web**

Descripción: El sistema debía ofrecer una interfaz web que se adaptara a diferentes dispositivos y navegadores. Esto incluía compatibilidad con navegadores comunes como Chrome, Firefox y Safari.

##### **Interfaz de Acceso**

Descripción: Se necesitaba una interfaz de inicio de sesión que permitiera a los usuarios autenticar sus credenciales de manera segura y eficiente. La interfaz debía ofrecer opciones para recuperar contraseñas y gestionar credenciales de forma sencilla.

## Interfaz de Gestión de Datos

Descripción: La interfaz para el registro y manejo de datos (pacientes, doctores, citas) debía permitir la entrada, modificación y visualización de información de manera efectiva.

Para mayor detalle ver documento SRS en el anexo N°5

## 4.2.Diseño

### 4.2.1. Arquitectura del Software

Figura 3.Arquitectura API REST



Elaborado por: Jorge Cárdenas - Luis Herrera

La arquitectura del sistema de gestión de citas e historias clínicas del Centro Odontológico Velasteguí se basa en una arquitectura API REST implementada con Laravel como framework de desarrollo para la capa de aplicación, MySQL como sistema de gestión de bases de datos relacional para la capa de datos y para la capa de presentación se usó Angular como framework de desarrollo.

### 15. Capa de Datos

- Base de Datos MySQL:

Descripción: Se utiliza MySQL como sistema de gestión de bases de datos, donde se almacena toda la información relacionada con los pacientes, doctores, citas, especialidades, entre otras.

- Estructura:
  - Tablas: Se incluyen tablas para Pacientes, Doctores, Citas, Especialidades, Horarios, Historias Clínicas, entre otras.
  - Relaciones: Las tablas están relacionadas mediante claves foráneas para mantener la integridad referencial, siguiendo un modelo entidad-relación (ER).
  - Optimización: Índices y procedimientos almacenados se implementan para optimizar consultas frecuentes y mantener el rendimiento.

## 16. Capa de Lógica de Negocio

- Framework Laravel:
 

Descripción: Laravel es el framework elegido para implementar la lógica de negocio del sistema, aprovechando su arquitectura modular y soporte nativo para la creación de APIS REST. El en centro del sistema está la API en Laravel que es la encargada de manejar la información, primero almacenándola en la base de datos para luego devolverla hacia el usuario.
- Componentes:
  - Rutas y Controladores: Se utilizan rutas definidas en web.php y api.php para manejar tanto las interfaces web como las solicitudes API. Los controladores gestionan la lógica de negocio, procesando las solicitudes y respondiendo con los datos necesarios.
  - Modelos Eloquent: Laravel utiliza Eloquent ORM para interactuar con la base de datos de manera eficiente, permitiendo operaciones CRUD sobre las tablas mediante un enfoque orientado a objetos.
  - Servicios: Se implementan servicios para encapsular la lógica de negocio.

## 17. Capa de Presentación

- Framework Angular:

Descripción: Angular CLI es un framework que facilita el desarrollo de aplicaciones ofreciendo comandos que automatizan tareas como la creación de componentes, servicios, compilación, entre otros. Se usó esta herramienta para crear una interfaz centralizada y profesional desde donde los administradores, doctores y secretarias pueden gestionar todos los procesos que ofrece el sistema satisfactoriamente.

- Componentes:

- Vistas Blade: Laravel utiliza Blade como motor de plantillas para renderizar las vistas en el frontend, permitiendo una integración eficiente con los datos provenientes del backend.
- JavaScript y AJAX: Para mejorar la experiencia del usuario, se utilizan técnicas de AJAX que permiten la actualización de contenido sin necesidad de recargar la página.

## 18. Integración y Comunicación

- API REST:

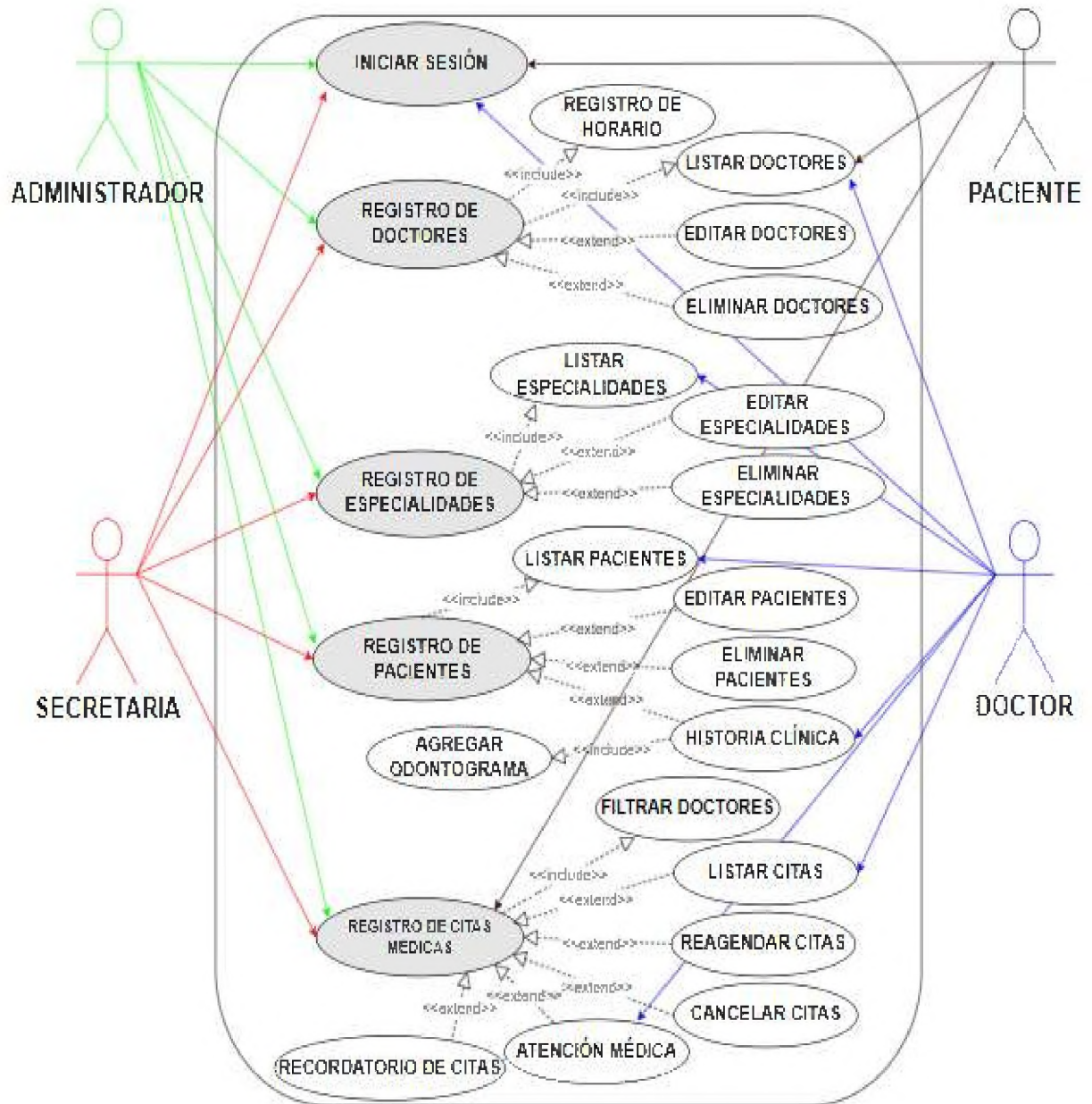
Descripción: La aplicación expone una API REST que permite la interacción con la capa de datos y lógica de negocio de manera uniforme y escalable.

- Operaciones: La API soporta operaciones CRUD para realizar peticiones y obtener respuestas de la base de datos, permitiendo obtener todos los recursos clave del sistema.
- Autenticación: Se implementa autenticación mediante tokens JWT (JSON Web Tokens) los que permiten controlar las solicitudes de información con respecto a la identidad y privilegios del usuario.

## 4.2.2. Diagramas

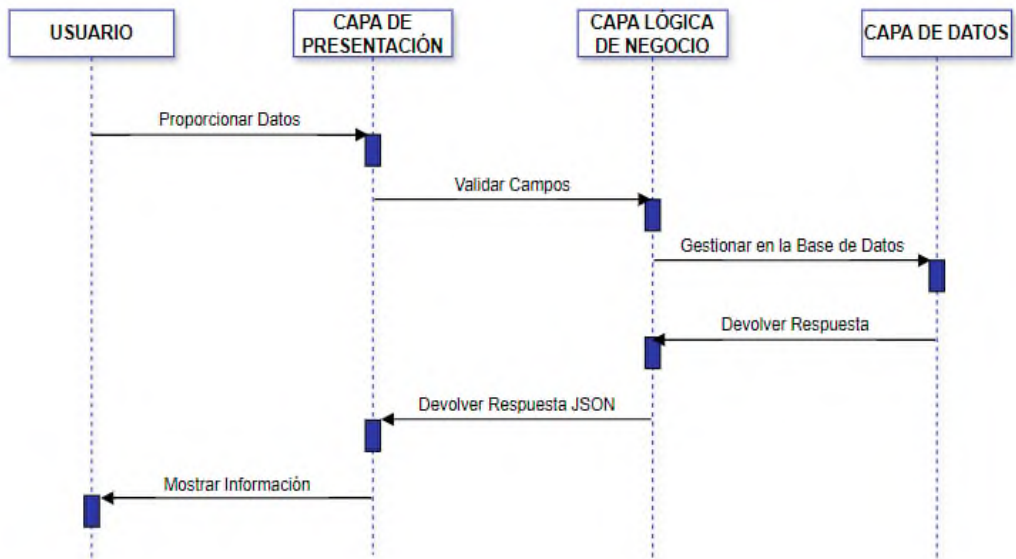
### 4.2.2.1. Diagrama de Caso de Uso

Figura 4. Diagrama de Casos de Uso



Elaborado por: Jorge Cárdenas - Luis Herrera

Figura 5. Diagrama de Secuencia: MANEJO DE LA INFORMACIÓN

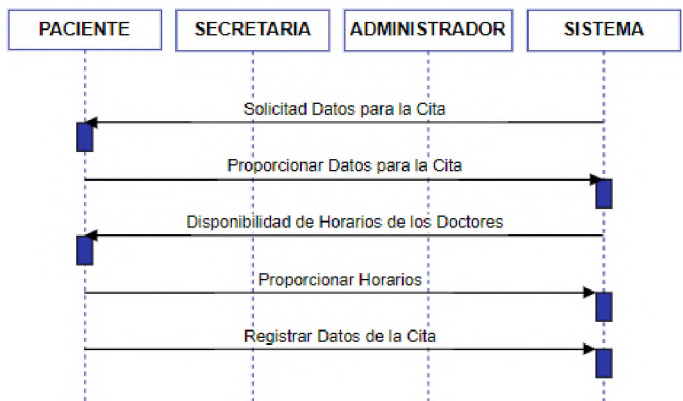


Elaborado por: Jorge Cárdenas - Luis Herrera

Figura 6. Diagrama de Secuencia: REGISTRO DE DATOS



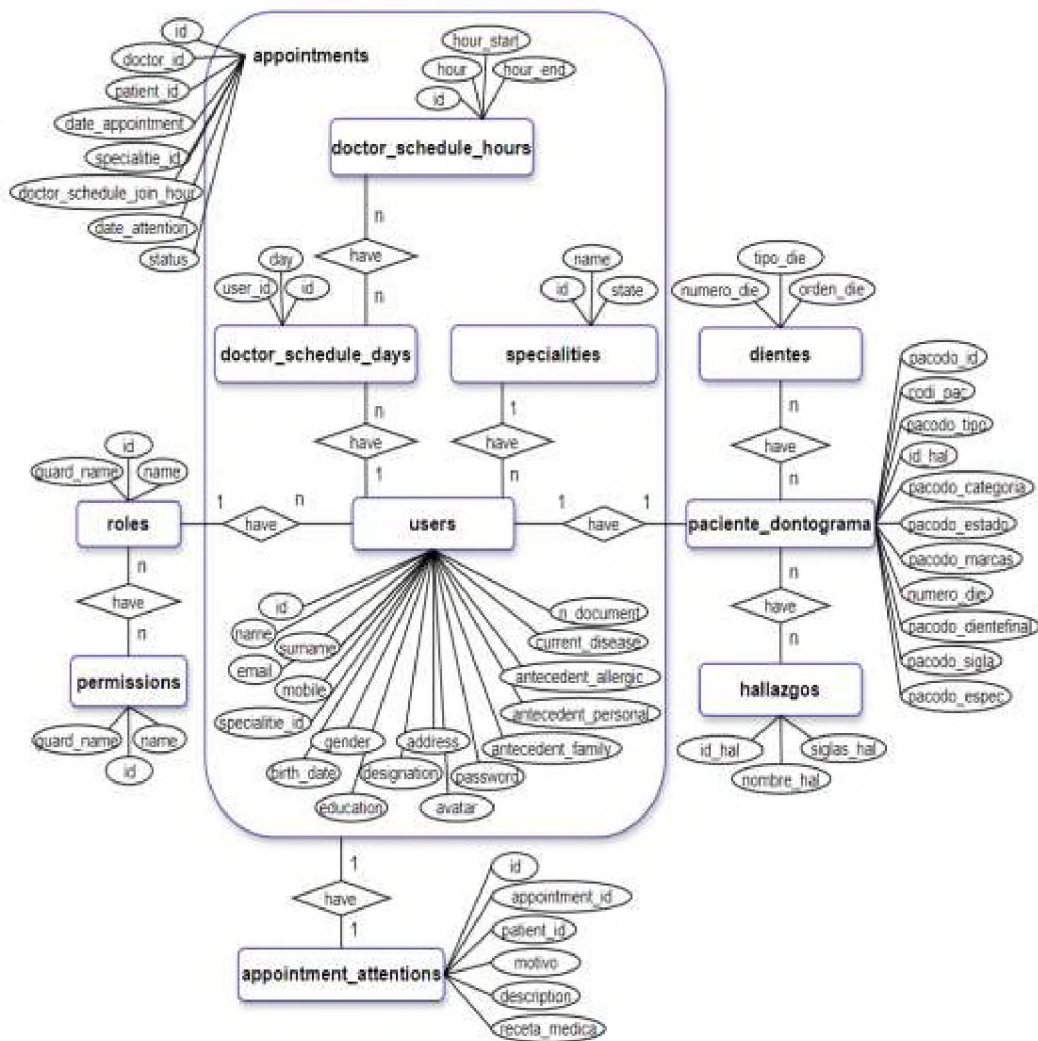
Figura 7. Diagrama de Secuencia: CITAS MÉDICA



Elaborado por: Jorge Cárdenas - Luis Herrera

### 4.2.2.3. Diagrama Entidad Relación

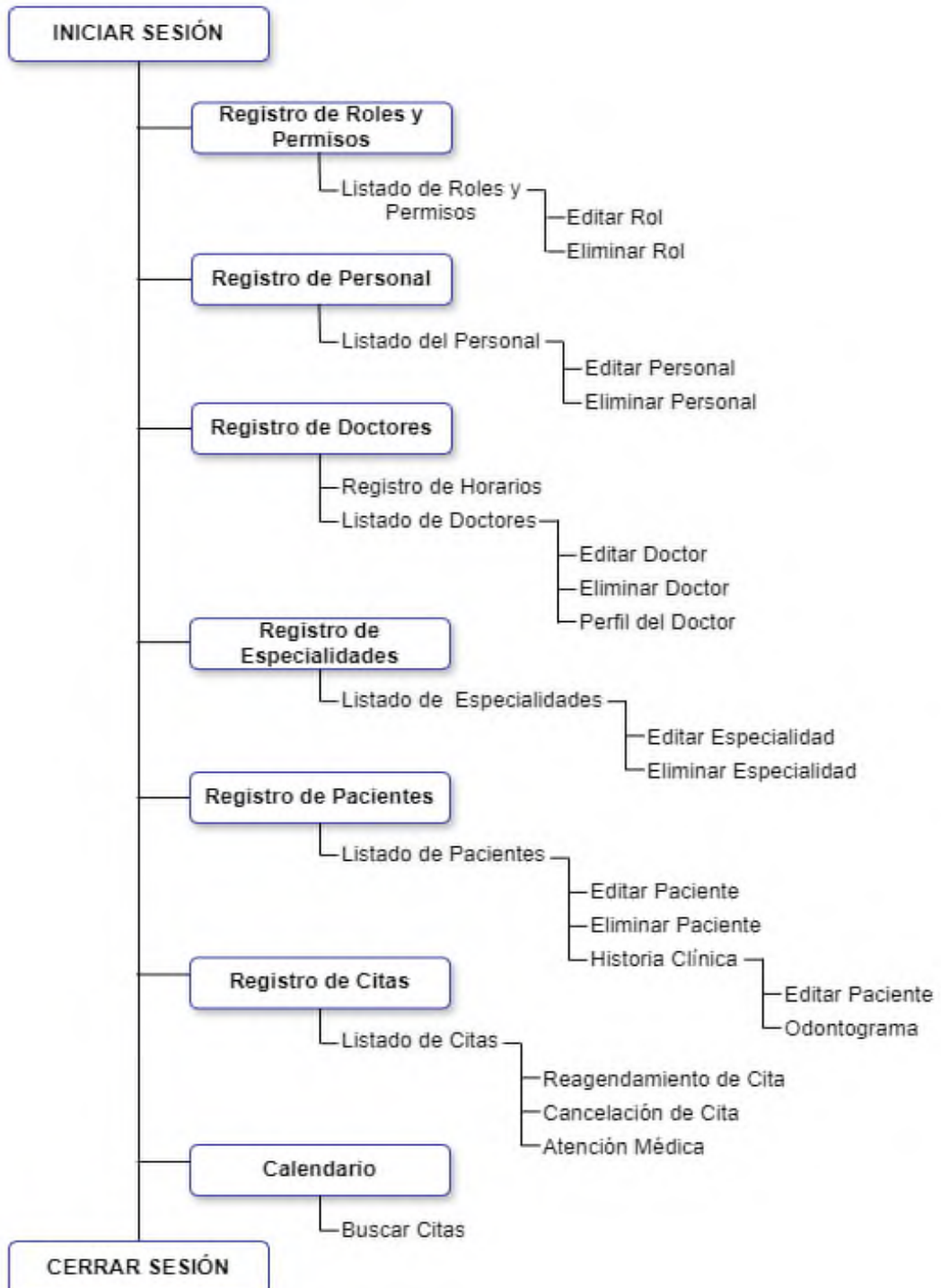
Figura 8. Diagrama Entidad - Relación



Elaborado por: Jorge Cárdenas - Luis Herrera

### 4.2.3.1. Patrón de navegación

Figura 9. Patrón de Navegación ADMINISTRADOR



Elaborado por: Jorge Cárdenas - Luis Herrera

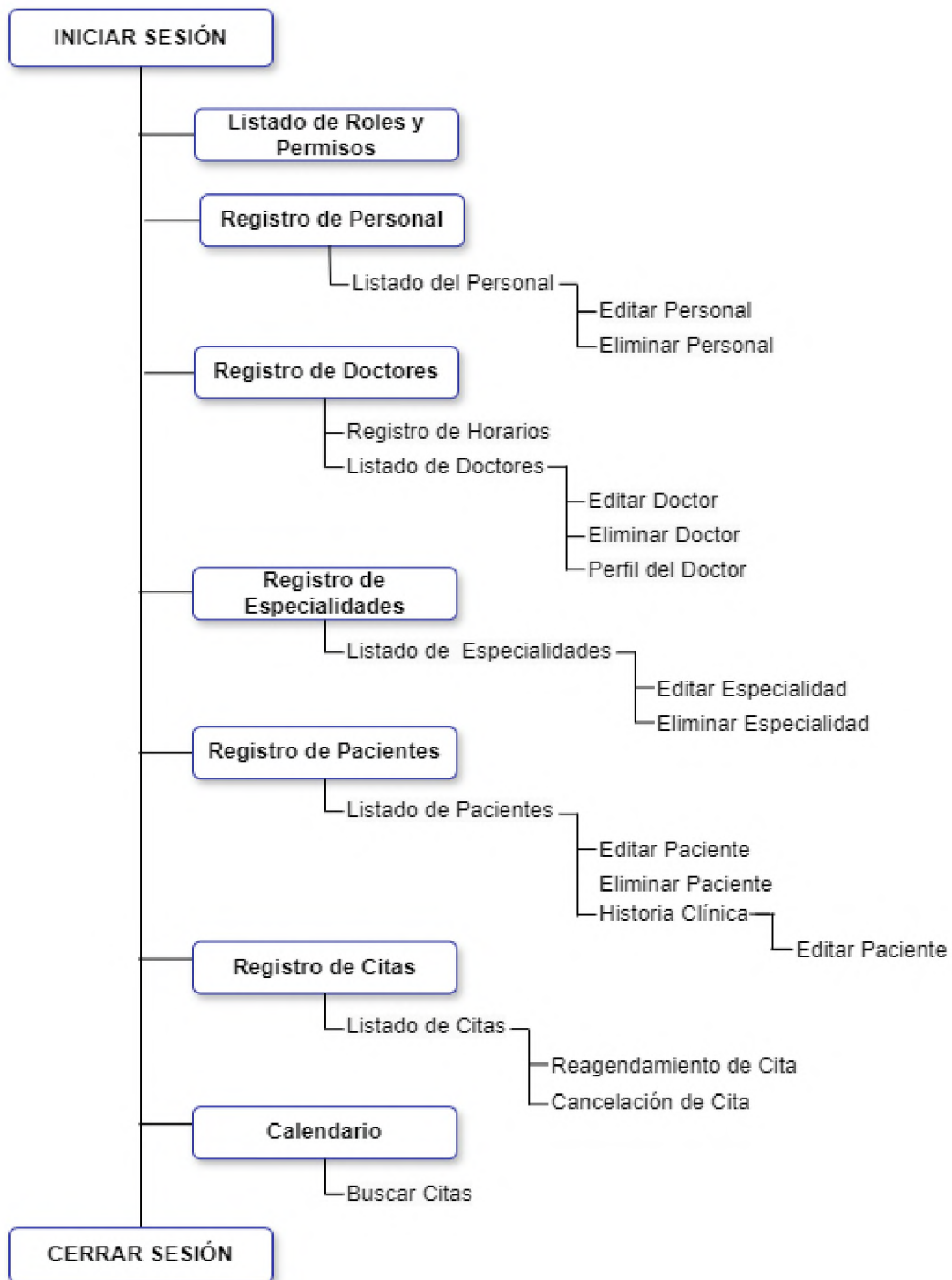
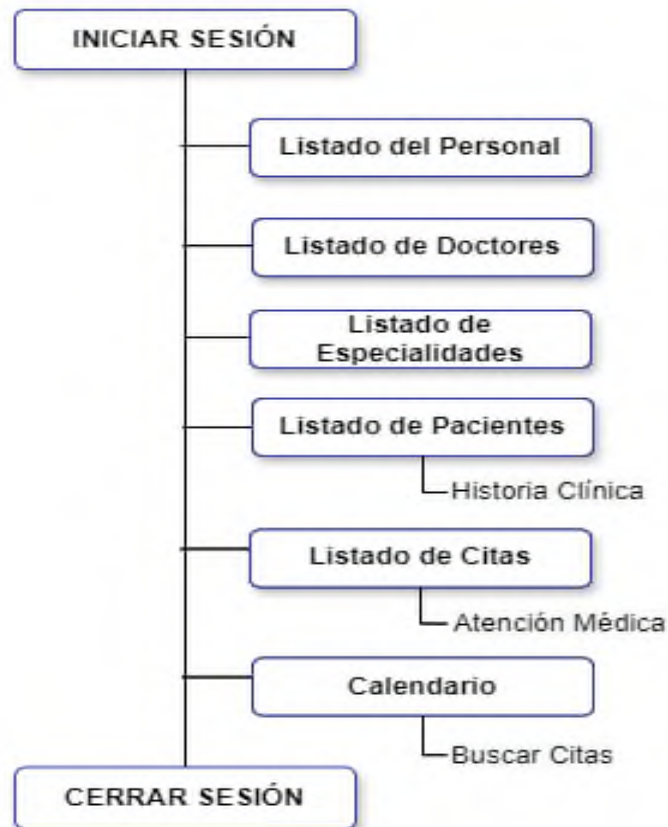


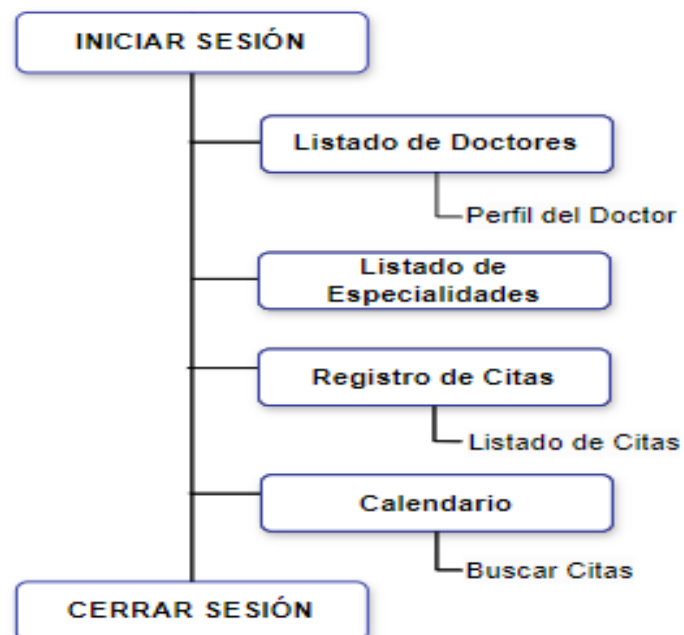
Figura 10. Patrón de Navegación SECRETARIA

Figura 11. Patrón de Navegación DOCTOR



Elaborado por: Jorge Cárdenas - Luis Herrera

Figura 12. Patrón de Navegación PACIENTE



Elaborado por: Jorge Cárdenas - Luis Herrera

#### 4.2.3.2. Interfaz de Pantallas

- **Color principal**

Blanco

Código hexadecimal: #FFFFFF

- **Color secundario**

Azure

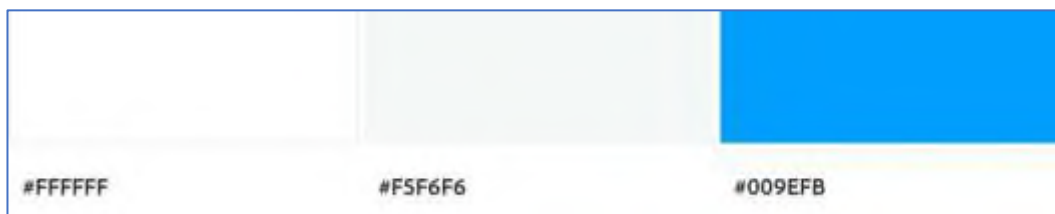
Código hexadecimal: #F5F6F6

- **Color complementario**

Azul Gema

Código hexadecimal: #2B8DFE

Figura 13. Paleta de Colores del Sistema



Elaborado por: Jorge Cárdenas - Luis Herrera

#### **Tipografía:**

Las tipografías seleccionadas para el desarrollo del sistema de gestión de citas médicas y manejo de historias clínicas del Centro Odontológico Velasteguí han sido cuidadosamente elegidas para garantizar una óptima legibilidad y una apariencia profesional, alineándose con los estándares modernos de diseño web.

Las características de las tipografías empleadas son las siguientes:

- **Tipografía:** Poppins
- **Clasificación:** Sans-serif.
- **Aplicación:**

**Títulos Principales:** La tipografía Poppins se utiliza predominantemente para los títulos principales en las distintas páginas del sistema, proporcionando una apariencia moderna y limpia que facilita la lectura y mejora la estética visual del sistema.

- **Tamaños:**  
**Títulos:** 18 px.  
**Texto Regular:** 14 px.
- **Justificación del Uso:**  
**Estética y Consistencia:** El uso de una única tipografía para tanto títulos como texto regular contribuye a una estética uniforme en todo el sistema, lo que es crucial para la experiencia del usuario.
- **Estándares de Diseño Web:**

**Accesibilidad:**

El uso de una tipografía sans-serif como Poppins se alinea con las mejores prácticas de accesibilidad en diseño web, facilitando la lectura para usuarios con discapacidades visuales.

**Responsive Design:** Los tamaños de fuente elegidos (18 px para títulos y 14 px para texto regular) han sido seleccionados considerando la adaptabilidad del sistema a diferentes resoluciones y tamaños de pantalla, garantizando una experiencia de usuario óptima en dispositivos móviles y de escritorio.

**Logo del sistema**

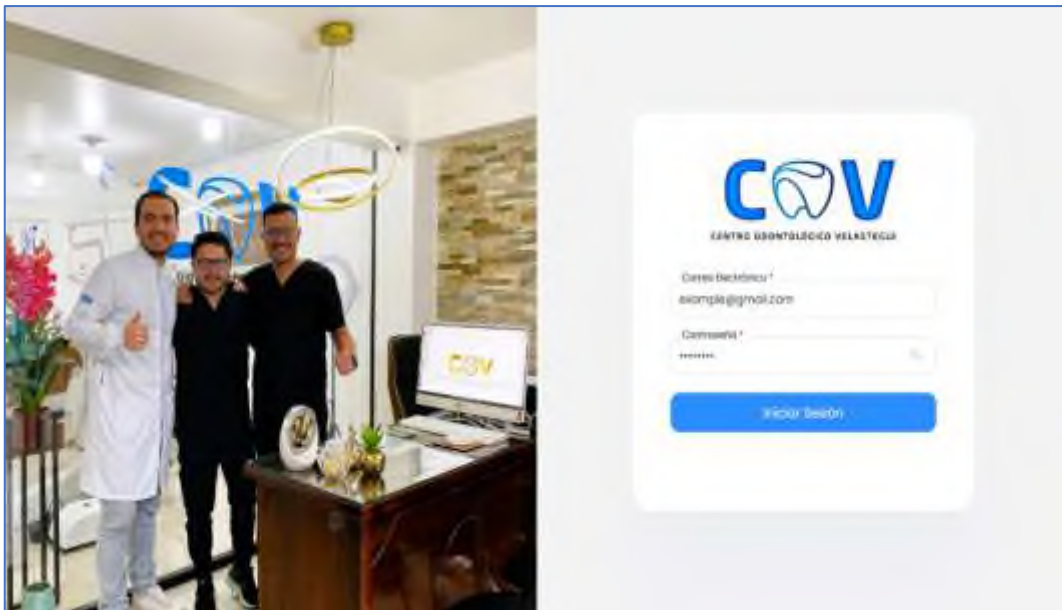
*Figura 14. Logo del Sistema*



*Elaborado por: Jorge Cárdenas - Luis Herrera*

## Login

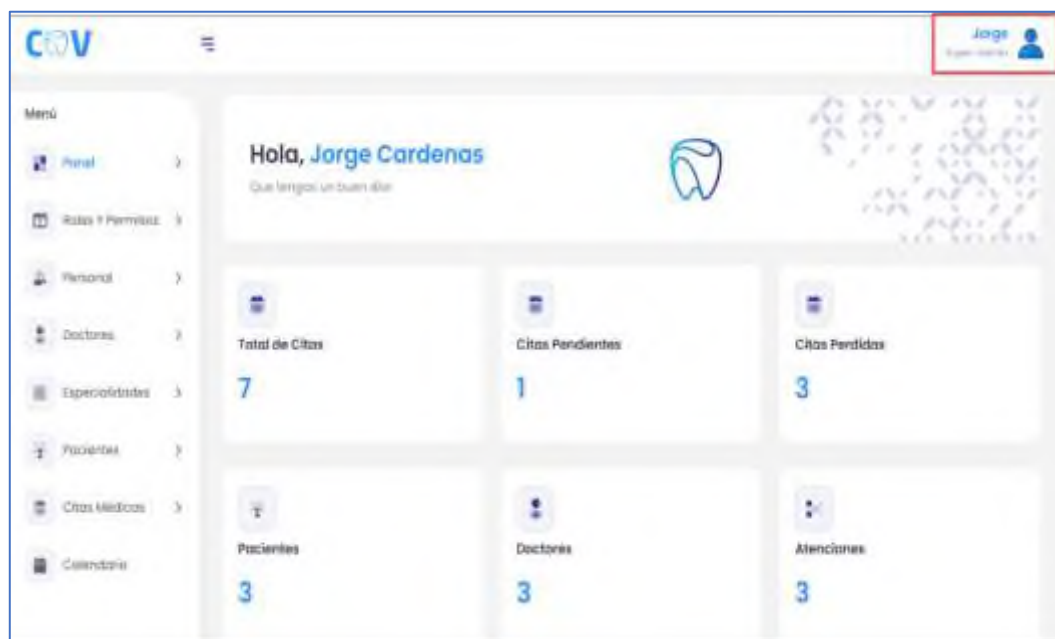
Figura 15. Login



Elaborado por: Jorge Cárdenas - Luis Herrera

## Panel del administrador

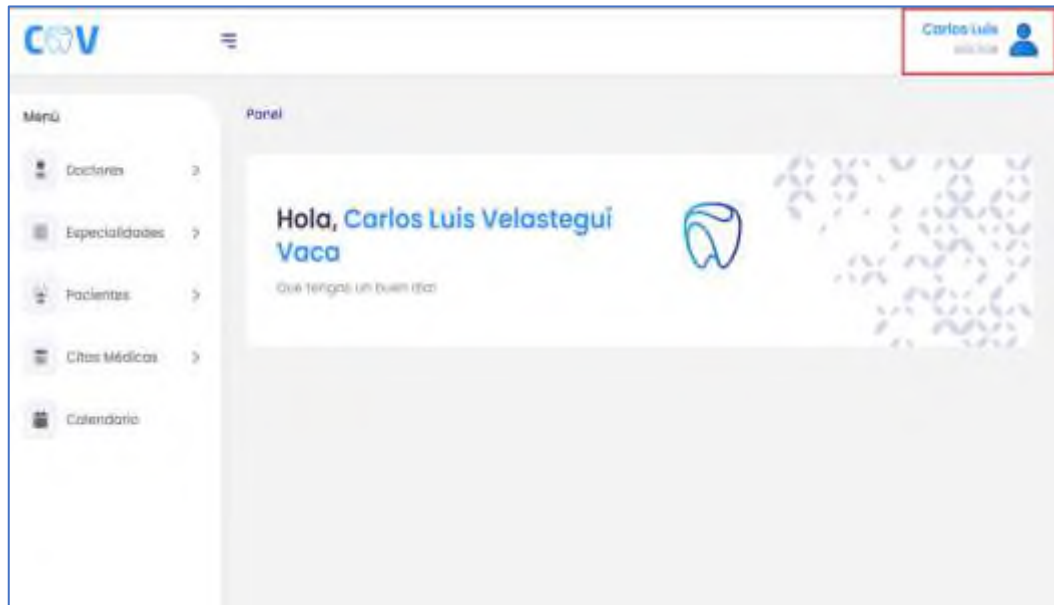
Figura 16. Panel Administrador



Elaborado por: Jorge Cárdenas - Luis Herrera

## Panel del doctor

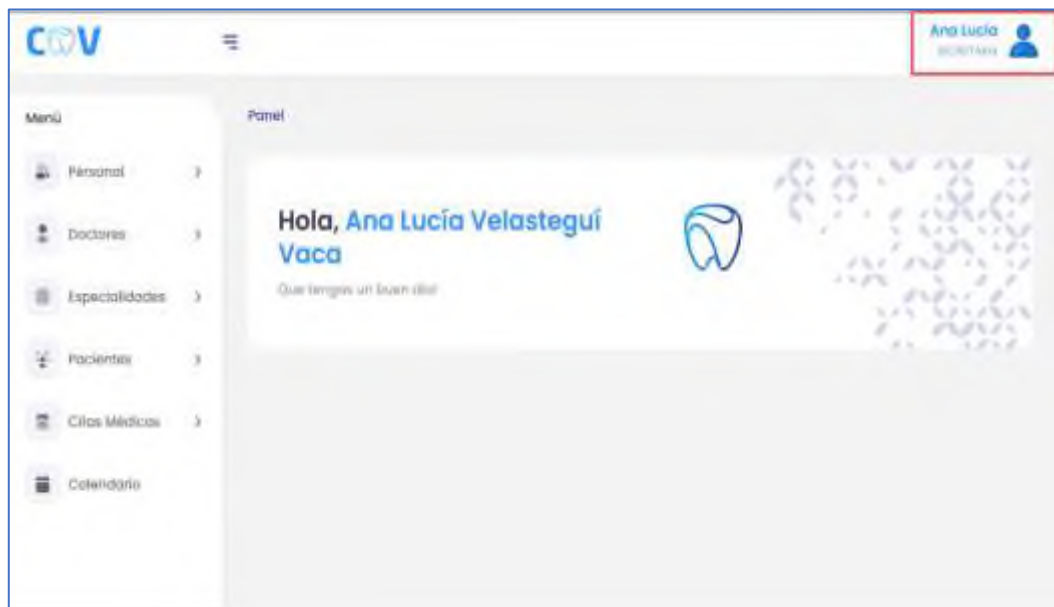
Figura 17. Panel Doctor



Elaborado por: Jorge Cárdenas - Luis Herrera

## Panel de la secretaria

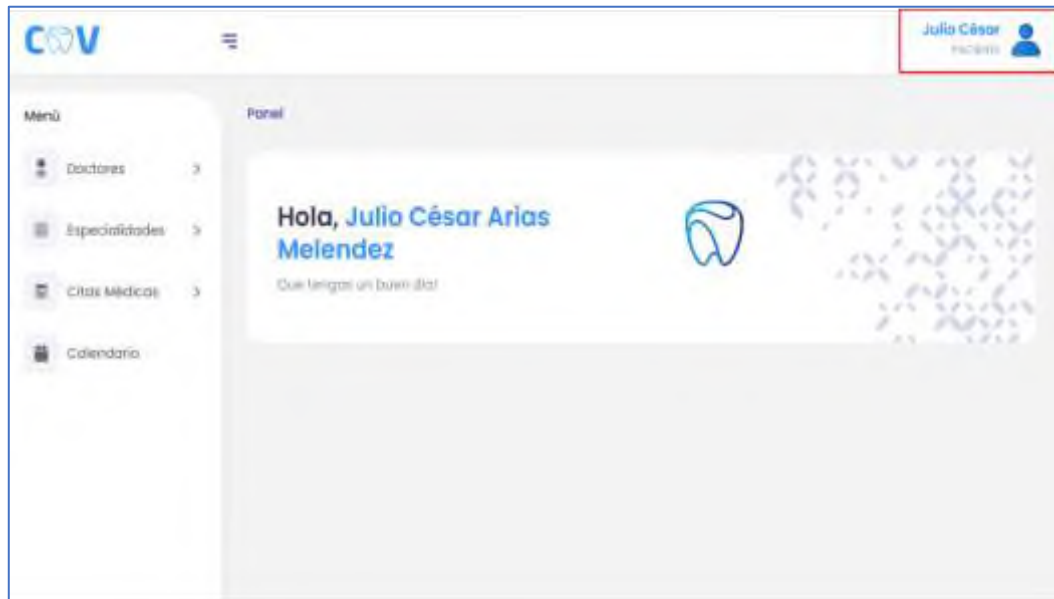
Figura 18. Panel secretaria



Elaborado por: Jorge Cárdenas - Luis Herrera

## Panel del paciente

Figura 19. Panel Paciente



Elaborado por: Jorge Cárdenas - Luis Herrera

## 4.3. Programación

### 4.3.1. Definición del objetivo

El sistema de gestión de citas e historias clínicas SIWCH-OV está estructurado haciendo uso de una API REST (Representational State Transfer), siguiendo un conjunto de principios arquitectónicos para la comunicación entre sistemas a través de HTTP. Aquí, los recursos se identifican mediante URL, y se usan métodos HTTP como GET, POST, PUT, DELETE para realizar las operaciones necesarias sobre estos recursos.

En la capa de presentación (Angular), es donde el cliente envía peticiones HTTP a la API y muestra los resultados. En el núcleo del sistema, en la capa de lógica de negocio (Laravel), esta API es la encargada de la comunicación con la base de datos y contiene la lógica para crear, actualizar, leer o eliminar la información. Aquí se define rutas y controladores que manejan las peticiones HTTP.

### **4.3.2. Análisis del problema**

Durante mucho tiempo el manejo de la información de los pacientes del Centro Odontológico Velasteguí y la gestión de sus citas odontológicas se ha estado realizando de manera manual, lo que ha ocasionado claros inconvenientes de horarios para las atenciones a los pacientes y pérdida de su información.

De manera que se aborden estos problemas adecuadamente, se desarrolló un sistema web capaz de gestionar de mejor manera los procesos de agendamientos de citas y manejo de la información, aportando satisfactoriamente a la calidad de servicio brindado a la ciudadanía del cantón Echeandía.

### **4.3.3. Diseño del Algoritmo**

- **Inicio de Sesión**

1. Inicio del algoritmo.
2. Solicitar al usuario que ingrese nombre de usuario y contraseña.
3. Validar si los campos necesarios son completados.
4. Si los campos no se han completado, mostrar mensaje de advertencia y volver al paso 2.
5. Si los campos son completados, proceder a consultar a la base de datos si los datos son correctos.
6. En el caso de ser correctos, dar acceso.
7. En el caso de ser incorrectos mostrar mensaje de advertencia y volver al paso 2.
8. Fin del algoritmo.

- **Registro Roles y Permisos**

1. Inicio del algoritmo.
2. Insertar los datos del rol y asignar los permisos necesarios.
3. Validar si los campos necesarios son completados.
4. Si los campos no se han completado, mostrar mensaje de advertencia y volver al paso 2.
5. Si los campos son completados, registrar rol.
6. Fin del algoritmo.

- **Registro de Personal**

1. Inicio del algoritmo.
2. Insertar los datos del usuario.
3. Validar si los campos necesarios son completados.
4. Si los campos no se han completado, mostrar mensaje de advertencia y volver al paso 2.
5. Si los campos son completados, proceder.
6. Validar si los campos de contraseña coinciden.
7. Si los campos no coinciden, mostrar mensaje de advertencia y volver al paso 2.
8. Si los campos coinciden, registrar usuario.
9. Fin del algoritmo.

- **Registro de Especialidades**

1. Inicio del algoritmo.
2. Insertar los datos de la especialidad.
3. Validar si los campos necesarios son completados.
4. Si los campos no se han completado, mostrar mensaje de advertencia y volver al paso 2.
5. Si los campos son completados, registrar especialidad.
6. Fin del algoritmo.

- **Registro de Doctores**

1. Inicio del algoritmo.
2. Insertar los datos del doctor.
3. Validar si los campos necesarios son completados.
4. Si los campos no se han completado, mostrar mensaje de advertencia y volver al paso 2.
5. Si los campos son completados, proceder.
6. Validar si los campos de contraseña coinciden.
7. Si los campos no coinciden, mostrar mensaje de advertencia y volver al paso 2.
8. Si los campos coinciden, proceder.
9. Validar si el horario es completado.

10. Si el horario no es completado, dar mensaje de advertencia y volver al paso 2.

11. Si el horario es completado, registrar doctor.

12. Fin del algoritmo.

- **Registro de Pacientes**

1. Inicio del algoritmo.

2. Insertar los datos del paciente.

3. Validar si los campos necesarios son completados.

4. Si los campos no se han completado, mostrar mensaje de advertencia y volver al paso 2.

5. Si los campos son completados, proceder.

6. Validar si los campos de contraseña coinciden.

7. Si los campos no coinciden, mostrar mensaje de advertencia y volver al paso 2.

8. Si los campos coinciden, registrar usuario.

9. Fin del algoritmo.

- **Registro de Citas Médicas**

1. Inicio del algoritmo.

2. Insertar los datos para la cita.

3. Verificar la disponibilidad del doctor en la fecha, hora y especialidad asignada.

4. Si el doctor no está disponible dar un mensaje de alerta y volver al paso 2.

5. Si el doctor está disponible, mostrar sus turnos disponibles en la hora seleccionada.

6. Insertar datos de turno para la cita.

7. Validar si los campos necesarios son completados.

8. Si los campos no se han completado, mostrar mensaje de advertencia y volver al paso 6.

9. Si los datos fueron completados, proceder.

10. Insertar los datos del paciente.

11. Validar si los campos necesarios son completados.

12. Si los campos no se han completado, mostrar mensaje de advertencia y volver al paso 10.

13. Si los campos fueron completados, registrar cita médica.

14. Fin del algoritmo.

- **Registro de Atención Médica**

1. Inicio del algoritmo.

2. Insertar datos de motivo de consulta, diagnóstico y receta médica.

3. Validar si los campos necesarios son completados.

4. Si los campos no se han completado, mostrar mensaje de advertencia y volver al paso 2.

5. Si los campos fueron completados, registrar atención médica.

6. Fin del algoritmo.

- **Registro de Odontogramas**

1. Inicio del Algoritmo.

2. Seleccionar el procedimiento odontológico a realizar.

3. Seleccionar el diente o los dientes a los que se va a aplicar el procedimiento.

4. De ser necesario, seleccionar el estado del procedimiento, las partes del diente y las especificaciones.

5. Si los campos fueron completados, registrar odontograma.

6. Fin del algoritmo.

#### **4.3.4. Codificación**

##### **REGISTRAR CITA MÉDICA**

##### **Controlador para la lógica de registro de una cita médica**

```
class AppointmentController extends Controller
```

```
{
```

```
/**
```

```
 * Método para listar las citas
```

```
*/
```

```
public function index(Request $request)
```

```
{
```

```
    $this->authorize('viewAny', Appointment::class);
```

```

$specialitie_id = $request->specialitie_id;
$name_doctor = $request->search;
$date = $request->date;
$user = auth("api")->user();
$appointments = Appointment::filterAdvance($specialitie_id, $name_doctor,
$date, $user)
    ->orderBy("id", "desc")
    ->paginate(10);

// Cambiamos el estado de la cita a 'Perdida' si ya ha pasado su fecha y hora
$currentDateTime = Carbon::now();
$appointments->each(function($appointment) use ($currentDateTime) {
    $appointmentDateTime = Carbon::parse($appointment->date_appointment);
    if ($appointmentDateTime->lt($currentDateTime) && $appointment->status
== 1) {
        $appointment->status = 3;
        $appointment->save();
    }
});
return response()->json([
    "total" => $appointments->total(),
    "appointments" => AppointmentCollection::make($appointments),
]);
}

/**
 * Método que configura los horarios con especialidades
 */
public function config(){
    $hours = [
        [
            "id" => "08",
            "name" => "8:00 AM",
        ],
        [
            "id" => "09",

```

```
"name" => "9:00 AM",
],
[
  "id" => "10",
  "name" => "10:00 AM",
],
[
  "id" => "11",
  "name" => "11:00 AM",
],
[
  "id" => "12",
  "name" => "12:00PM",
],
[
  "id" => "13",
  "name" => "01:00 PM",
],
[
  "id" => "14",
  "name" => "02:00 PM",
],
[
  "id" => "15",
  "name" => "03:00 PM",
],
[
  "id" => "16",
  "name" => "04:00 PM",
],
[
  "id" => "17",
  "name" => "05:00 PM",
],
```

```

];
$specialities = Specialitie::where("state",1)->get();
return response()->json([
    "specialities" => $specialities,
    "hours" => $hours,
]);
}
/**
 * Método para filtrar la disponibilidad del doctor
 */
public function filter(Request $request) {
    $this->authorize('filter',Appointment::class);
    $date_appointment = $request->date_appointment;
    $hour = $request->hour;
    $specialitie_id = $request->specialitie_id;
    date_default_timezone_set('America/Guayaquil');
    Carbon::setLocale('es');
    DB::statement("SET lc_time_names = 'es_ES'");
    $name_day = Carbon::parse($date_appointment)->dayName;
    $doctor_query =
DoctorScheduleDay::where("day","like","%".$name_day."%")
        ->whereHas("doctor",function($q) use($specialitie_id){
            $q->where("specialitie_id",$specialitie_id);
        }->whereHas("schedules_hours",function($q)
use($hour) {
            $q->whereHas("doctor_schedule_hour",function($qs)
use($hour) {
                $qs->where("hour",$hour);
            });
        }->get());
    $doctors = collect([]);
    // Revisamos la disponibilidad horaria de los doctores y proporcionamos sus
    turnos disponibles
    foreach ($doctor_query as $doctor_q) {

```

```

$segments =
DoctorScheduleJoinHour::where("doctor_schedule_day_id",$doctor_q->id)
    ->whereHas("doctor_schedule_hour",function($q)
use($hour) {
        $q->where("hour",$hour);
    }->get());
// Controlamos los turnos ocupados
$doctors->push([
    "doctor" => [
        "id" => $doctor_q->doctor->id,
        "full_name" => $doctor_q->doctor->name .' '.$doctor_q->doctor-
>surname,
        "specialitie" => [
            "id" => $doctor_q->doctor->specialitie->id,
            "name" => $doctor_q->doctor->specialitie->name,
        ],
    ],
    "segments" => $segments->map(function($segment)
use($date_appointment,$doctor_q){
        $appointment =
Appointment::whereHas("doctor_schedule_join_hour",function($q)
use($segment,$doctor_q){
            $q->where("doctor_schedule_hour_id",$segment-
>doctor_schedule_hour_id);
        })
        -
>whereDate("date_appointment",Carbon::parse($date_appointment)->format("Y-
m-d"))
            ->where("doctor_id",$doctor_q->doctor->id)
            ->first();
    return [
        "id" => $segment->id,
        "doctor_schedule_day_id" => $segment->doctor_schedule_day_id,
        "doctor_schedule_hour_id" => $segment-
>doctor_schedule_hour_id,
        "is_appointment" => $appointment ? true : false,

```

```

        "format_segment" => [
            "id" => $segment->doctor_schedule_hour->id,
            "hour_start" => $segment->doctor_schedule_hour->hour_start,
            "hour_end" => $segment->doctor_schedule_hour->hour_end,
            "format_hour_start" => Carbon::parse(date("Y-m-d")).'
            '.$segment->doctor_schedule_hour->hour_start->format("h:i A"),
            "format_hour_end" => Carbon::parse(date("Y-m-d")).'
            '.$segment->doctor_schedule_hour->hour_end->format("h:i A"),
            "hour" => $segment->doctor_schedule_hour->hour,
        ]
    ];
}
});
}
return response()->json([
    "doctors" => $doctors,
]);
}
/**
 * Método para configurar el calendario
 */
public function calendar(Request $request) {
    $specialitie_id = $request->specialitie_id;
    $search_doctor = $request->search_doctor;
    $search_patient = $request->search_patient;
    $user = auth("api")->user();
    $appointments =
Appointment::filterAdvancePay($specialitie_id,$search_doctor,$search_patient,n
ull,null,$user)
        ->orderBy("id","desc")
        ->get();

    // Cambiamos el estado de la cita a 'Perdida' si ya ha pasado su fecha y hora
    $appointments->each(function($appointment) {
        if (Carbon::now()->gt(Carbon::parse($appointment->date_appointment))
        && $appointment->status == 1) {

```

```

        $appointment->status = 3;
        $appointment->save();
    }
});
return response()->json([
    "appointments" => $appointments->map(function($appointment) {
        return [
            "id" => $appointment->id,
            "title" => "CITA MEDICA - " . ($appointment->doctor->name. '
            . $appointment->doctor->surname). " - " . $appointment->specialitie->name,
            "start" => Carbon::parse($appointment->date_appointment)-
            >format("Y-m-d")."T".$appointment->doctor_schedule_join_hour-
            >doctor_schedule_hour->hour_start,
            "end" => Carbon::parse($appointment->date_appointment)-
            >format("Y-m-d")."T".$appointment->doctor_schedule_join_hour-
            >doctor_schedule_hour->hour_end,
        ];
    })
]);
}
/**
 * Método para buscar a un paciente
 */
public function query_patient(Request $request){
    $n_document = $request->get("n_document");
    $patient = user::where("n_document", $n_document)->first();
    if(!$patient){
        return response()->json([
            "message" => 403,
        ]);
    }
    return response()->json([
        "message" => 200,
        "name" => $patient->name,
        "surname" => $patient->surname,
    ]);
}

```

```

        "mobile" => $patient->mobile,
        "n_document" => $patient->n_document,
    ];
}
/**
 * Método para registrar una cita médica
 */
public function store(Request $request)
{
    $this->authorize('create', Appointment::class);
    $patient = User::where("n_document", $request->n_document)->first();
    if (!$patient) {
        // Mostramos un mensaje de advertencia si el paciente no existe
        return response()->json([
            "message" => "El paciente no existe en la base de datos. Por favor, registre
al paciente antes de agendar la cita."
        ], 400);
    }
    $segment = DoctorScheduleJoinHour::find($request-
>doctor_schedule_join_hour_id);
    if (!$segment) {
        return response()->json([
            "message" => "El segmento de hora seleccionado no es válido."
        ], 400);
    }
    // Combinamos la fecha de la cita con la hora de inicio del turno
    $dateTime = Carbon::parse($request->date_appointment)->format('Y-m-d') . ' '
. $segment->doctor_schedule_hour->hour_start;
    $appointment = Appointment::create([
        "doctor_id" => $request->doctor_id,
        "patient_id" => $patient->id,
        "date_appointment" => $dateTime,
        "specialitie_id" => $request->specialitie_id,
    ]);
}

```

```

        "doctor_schedule_join_hour_id" => $request-
->doctor_schedule_join_hour_id,
        "user_id" => auth("api")->user()->id,
    ];
    return response()->json([
        "message" => "Cita registrada con éxito."
    ]);
}
/**
 * Método para mostrar los registros
 */
public function show(string $id)
{
    $appointment = Appointment::findOrFail($id);
    $this->authorize('view', $appointment);
    // Obtenemos la hora actual del sistema
    $currentDateTime = Carbon::now();
    $appointmentDateTime = Carbon::parse($appointment->date_appointment);
    \Log::info('Visualización de cita', [
        'appointment_id' => $appointment->id,
        'currentDateTime' => $currentDateTime,
        'date_appointment' => $appointmentDateTime,
        'status' => $appointment->status
    ]);
    // Verificamos y actualizamos el estado si es necesario
    if ($currentDateTime->greaterThan($appointmentDateTime) &&
    $appointment->status == 1) {
        $appointment->status = 3;
        $appointment->save();
        \Log::info('Estado actualizado a Perdida en show', ['appointment_id' =>
    $appointment->id]);
    }
    return response()->json([
        "appointment" => AppointmentResource::make($appointment)
    ]);
}

```

```

    );
}
/**
 * Método para actualizar los registros
 */
public function update(Request $request, string $id)
{
    $appointment = Appointment::findOrFail($id);
    $this->authorize('update',$appointment);
    $appointment->update([
        "doctor_id" => $request->doctor_id,
        "date_appointment" => Carbon::parse($request->date_appointment)-
>format("Y-m-d h:i:s"),
        "specialitie_id" => $request->specialitie_id,
        "doctor_schedule_join_hour_id" => $request-
>doctor_schedule_join_hour_id,
    ]);
    return response()->json([
        "message" => 200,
    ]);
}
/**
 * Método para eliminar los registros
 */
public function destroy(string $id)
{
    $appointment = Appointment::findOrFail($id);
    $this->authorize('delete',$appointment);
    $appointment->delete();
    return response()->json([
        "message" => 200,
    ]);
}
}

```

#### 4.4.Pruebas

##### 4.4.1. Análisis de requisitos

En la fase de análisis de requisitos, se llevó a cabo una recopilación exhaustiva de las necesidades funcionales y no funcionales del sistema web para gestionar las citas e historias clínicas en el Centro Odontológico Velasteguí. Se realizaron reuniones con las partes interesadas clave, como doctores, administradores y personal de soporte, para identificar los requerimientos esenciales y asegurarse de que el sistema cubriera todas las expectativas. Estos requisitos fueron documentados y validados antes de proceder con las fases de desarrollo y despliegue, realizando los debidos controles de las operaciones que se van a realizar entre ellos. Para estos monitoreos operativos se hizo uso de varias herramientas, tales como, Insomnia y Cmd, que ayudaron de manera significativa a tener acceso con antelación a cada uno de los eventos que se dan en el sistema, para así poder controlarlos con anterioridad.

##### 4.4.2. Planificación de pruebas

La planificación de las pruebas se realizó previamente en el diagrama de Gantt (ver anexo N°1) en función de los módulos del sistema, las fechas y los recursos disponibles. Se organizó en tres tipos de pruebas que fueron distribuidas en un cronograma detallado para cumplir con los tiempos establecidos de desarrollo y validación, como se muestra en la siguiente figura:

Figura 20. Cronograma pruebas

5 Pruebas	10 días	lun 9/9/24	vie 20/9/24
5.1 Pruebas de unitarias	3 días	lun 9/9/24	mié 11/9/24
5.2 Pruebas de integración	4 días	jue 12/9/24	mar 17/9/24
5.3 Pruebas de aceptación	3 días	mié 18/9/24	vie 20/9/24

Elaborado por: Jorge Cárdenas - Luis Herrera

##### 4.4.3. Configuración del entorno de prueba

Para garantizar que todos los aspectos funcionales y no funcionales del sistema fueran probados de manera adecuada, se desarrollaron casos de prueba específicos, con sus respectivos datos de entrada y resultados esperados. Los casos de prueba incluyeron pruebas automatizadas para verificar directamente el

correcto funcionamiento de un caso de prueba específico, como por ejemplo el listado de citas con su respectiva información.

El entorno de prueba se configuró para reflejar lo más fielmente posible la lógica de producción del sistema. Se implementaron instancias de MySQL, Laravel y Angular, permitiendo que los procesos se ejecuten de manera óptima. Además, se utilizó a Insomnia como herramienta de testing automatizada para realizar pruebas de regresión y validación de datos continuamente durante el proceso de desarrollo.

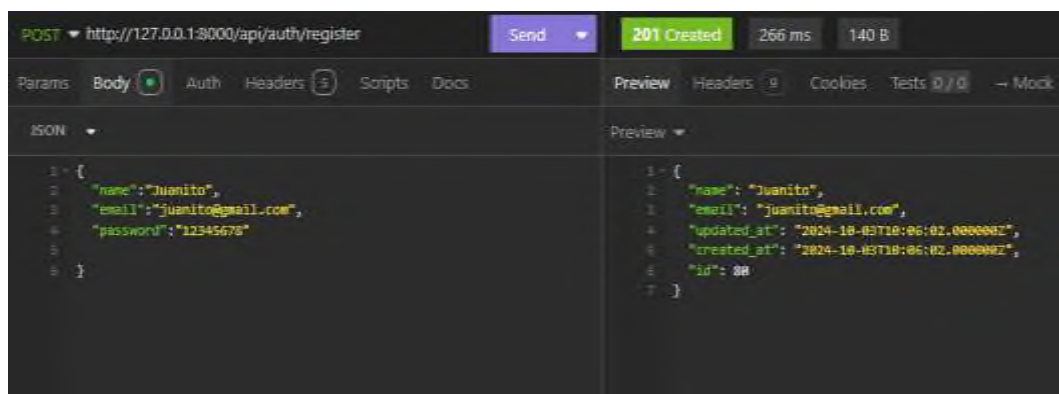
#### 4.4.4. Ejecución de las pruebas

#### 4.4.5. Pruebas Unitarias

Se ejecutaron las pruebas unitarias con el objetivo de llevar a cabo un control preventivo para todos los módulos del sistema y así tener la capacidad de controlar la información del sistema antes de ser puesta en producción.

Para el caso de prueba base “iniciar sesión” se realizaron algunas peticiones a nuestra API empezando por el registro de un nuevo usuario. Mediante el método POST y la función “register” vamos a enviar los parámetros de tipo JSON para registrar un nuevo paciente.

Figura 21. Registrar Paciente

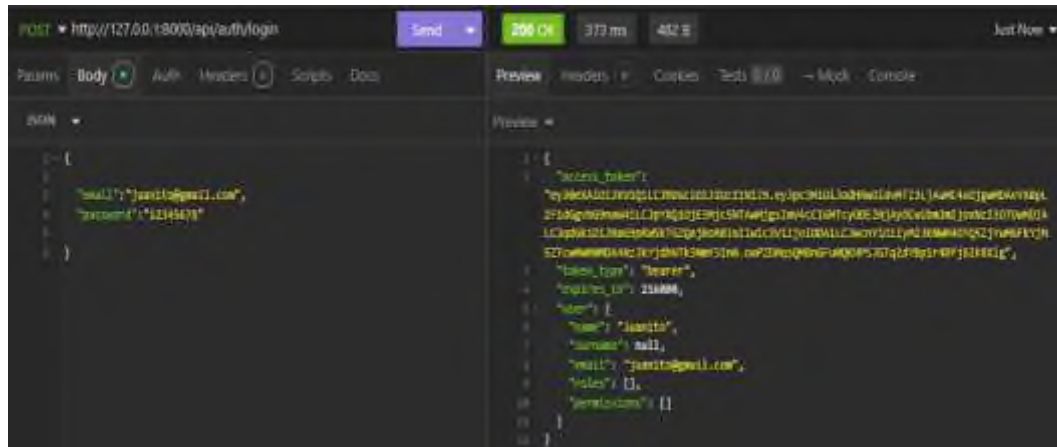


Elaborado por: Jorge Cárdenas - Luis Herrera

Una vez registrado el nuevo paciente, que a su vez es un usuario de nuestro sistema, podemos realizar una petición de tipo POST a nuestra API para autenticar a este usuario, es decir, realizar el inicio de sesión con los datos del paciente con nuestra función “login”. Así, obtendremos el token de autenticación

de este usuario para poder asociarlo con los permisos determinados que pueda tener dicho usuario.

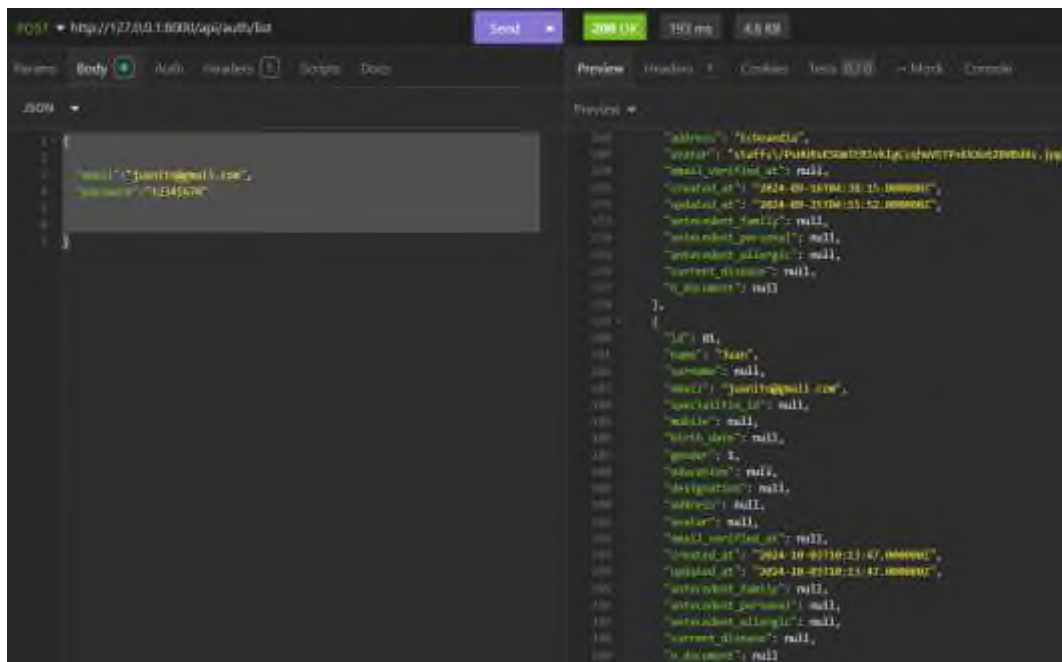
Figura 22. Iniciar Sesión



Elaborado por: Jorge Cárdenas - Luis Herrera

Cuando ya se ha realizado la autenticación del usuario se puede realizar una prueba para ver si en realidad el usuario ha sido registrado correctamente y aparece en la lista de registros. Entonces vamos a hacer uso de nuestra función “list” para listar a los usuarios registrados en nuestra base de datos.

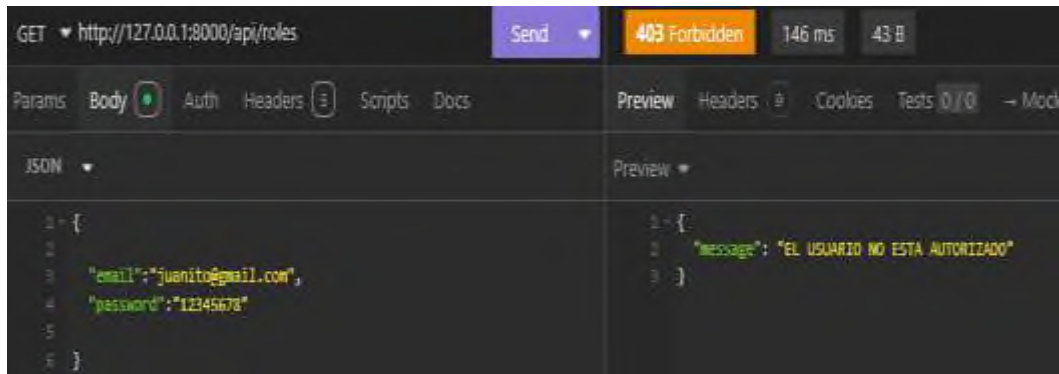
Figura 23. Listado de Usuarios



Elaborado por: Jorge Cárdenas - Luis Herrera

Cuando el usuario intente realizar una petición para la que no está autorizado, va a visualizar un mensaje de alerta, por ejemplo:

Figura 24. Listado de Roles y Permisos



Elaborado por: Jorge Cárdenas - Luis Herrera

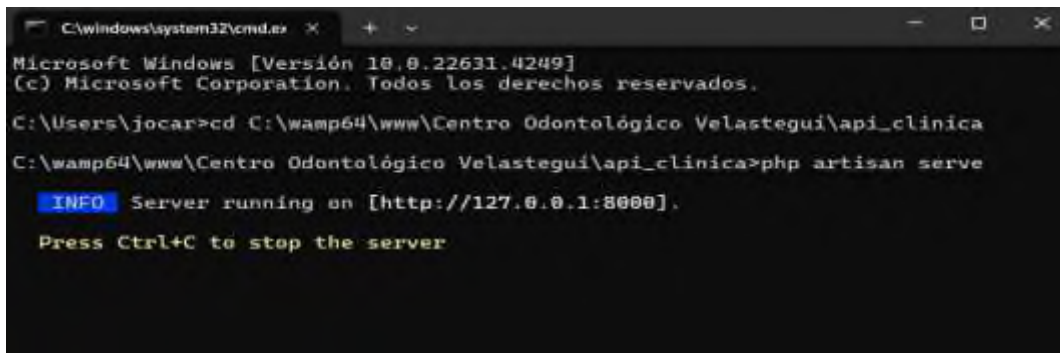
#### 4.4.6. Pruebas de Integración

Este tipo de pruebas están destinadas a monitorear la interacción entre los componentes de nuestro sistema. Por una parte, se encuentra nuestra API en Laravel haciendo de intermediario entre la información y el cliente. Por otro lado, están las vistas de nuestro sistema, donde se puede apreciar en la interfaz, la información almacenada en nuestra base de datos.

Para llevar un correcto desarrollo y encaminar nuestro proyecto a producción, se trabajó localmente en el servidor de WAMP, que nos brinda acceso al servidor web Apache para manejar las solicitudes HTTP, además del sistema de gestión de base de datos MySQL que es donde se va a almacenar nuestra información y PHP que es el lenguaje de servidor que usa Laravel.

Para la comunicación entre el backend (Laravel) y el frontend (Angular), se levantó un servidor de desarrollo en el puerto 8000 de la máquina local, el cual se encarga de escuchar las peticiones que entran a ese servidor y devolver la respuesta JSON correspondiente.

Figura 25. Servidor de Laravel



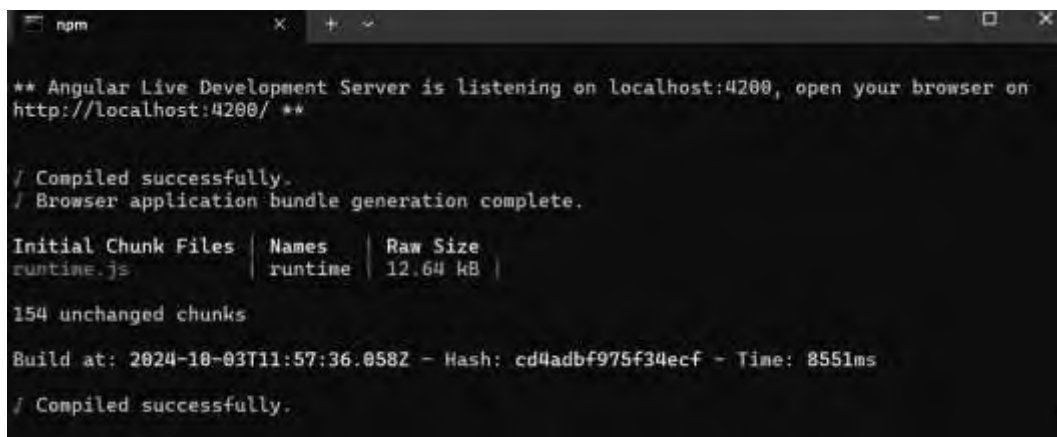
```
C:\windows\system32\cmd.exe X + -
Microsoft Windows [Versión 10.0.22631.4249]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jocar>cd C:\wamp64\www\Centro Odontológico Velastegui\api_clinica
C:\wamp64\www\Centro Odontológico Velastegui\api_clinica>php artisan serve
INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server
```

Elaborado por: Jorge Cárdenas - Luis Herrera

Por otro lado, para Angular que actúa como cliente del sistema haciendo las peticiones a Laravel, se levantó un servidor de desarrollo en el puerto 4200, este servidor facilita a grandes rasgos el desarrollo ya que maneja el término “hot-reload” (recarga en caliente), para cada cambio que se haga en el código

Figura 26. Servidor de Angular



```
npm X + -
** Angular Live Development Server is listening on localhost:4200, open your browser on
http://localhost:4200/ **

/ Compiled successfully.
/ Browser application bundle generation complete.

Initial Chunk Files | Names | Raw Size
runtime.js          | runtime | 12.64 kB |

154 unchanged chunks

Build at: 2024-10-03T11:57:36.058Z - Hash: cd4adb975f34ecf - Time: 8551ms

/ Compiled successfully.
```

Elaborado por: Jorge Cárdenas - Luis Herrera

Entonces, Angular se ejecuta en el puerto 4200 y es el encargado de manejar la interfaz de usuario y realizar las peticiones al servidor de Laravel, que está en el puerto 8000.

Ejemplo, cuando se quiere registrar una cita, Angular hace la petición POST a la API de Laravel, este método envía los datos del formulario de la cita al backend, en donde se procesa esta petición, se realizan operaciones en la base de datos y se devuelve un mensaje que Angular recibe y renderiza en la interfaz del usuario.

#### 4.4.7. Pruebas de Aceptación

Tabla 3. Pruebas de Aceptación

<b>PRUEBAS DE ACEPTACIÓN</b>			
<b>Involucrados:</b> Jorge Cárdenas, Luis Herrera			
<b>Propósito:</b> Realizar un análisis del comportamiento de los distintos usuarios del Centro Odontológico Velasteguí dentro del SIWCH-OV.			
N°	Aspecto Evaluado	Cumplimiento	
		SI	NO
<b>Lenguaje y ortografía</b>			
1	¿Los términos usados dentro del sistema, son entendibles para los distintos tipos de usuarios existentes?	X	
2	¿La información está estructurada correctamente y tiene una lógica entendible de acceso?	X	
3	¿Se usa una correcta ortografía?	X	
<b>Visibilidad y estética</b>			
4	¿El diseño de la interfaz es el adecuado para un Centro Odontológico?	X	
5	¿Son totalmente legibles los textos en todas las interfaces del sistema?	X	
6	¿El menú de despliegue funciona correctamente?	X	
7	¿Los gráficos muestran de manera oportuna la información requerida?	X	
8	¿Se muestra al usuario la posición en la que se encuentra, es decir, el módulo al que está accediendo?	X	
9	¿El sistema muestra un título del módulo en cuestión?	X	
<b>Navegación</b>			

<b>10</b>	¿Funciona a cabalidad la navegación entre los componentes del sistema?	X	
<b>11</b>	¿Existen accesos directos que vinculan a otras pantallas minimizando clics?	X	
<b>Alertas de seguridad y ayuda</b>			
<b>12</b>	¿Existen mensajes de alerta y error para las funcionalidades del sistema?	X	
<b>13</b>	¿Los mensajes de error desaparecen al actualizar la información correctamente?	X	
<b>14</b>	¿Existe paneles de búsqueda?	X	
<b>Personalización</b>			
<b>15</b>	¿El usuario puede personalizar la interfaz del sistema a su gusto?		X
<b>Tiempos de respuesta</b>			
<b>16</b>	¿El tiempo de respuesta para cada petición es aceptable?	X	

*Elaborado por: Jorge Cárdenas - Luis Herrera*

#### **4.4.8. Prueba de cierre**

Las pruebas de cierre consistieron en validar el sistema de manera global, asegurando que todas las funcionalidades trabajen en conjunto sin errores ni inconvenientes. Se realizó una revisión final de todos los casos de prueba y se garantizó que el sistema estuviera preparado para su despliegue en producción. Una vez que todas las pruebas fueron aprobadas, se emitió un informe de cierre detallando los resultados y aprobaciones obtenidas.

Tabla 4. Ficha de Observación

<b>FICHA DE OBSERVACIÓN</b>				
<b>Involucrados:</b> Jorge Cárdenas, Luis Herrera				
<b>Fecha:</b> 29/09/2024			<b>Hora:</b> 22:00 PM	
<b>Propósito:</b> Plasmar una descripción general de los resultados de las pruebas realizadas en el Sistema de Gestión de Citas Médicas e Historias Clínicas.				
1=Regular		2=Bueno		3=Muy Bueno
N°	Características	Métrica Interna	Valoración Promedio	Porcentaje Promedio
1	Pruebas Unitarias	Monitorear el comportamiento de las funcionalidades principales del Sistema	3	100%
2	Pruebas de Integración	Verificar que los componentes del sistema tienen un correcto método de comunicación	3	100%
3	Pruebas de Aceptación	Asegurar el correcto manejo de las distintas funcionalidades de los usuarios del Centro Odontológico Velasteguí	2	100%

Elaborado por: Jorge Cárdenas - Luis Herrera

## CONCLUSIONES

- La implementación del sistema web utilizando Extreme Programming (XP) resultó ser adecuada para la naturaleza del proyecto, permitiendo una alta adaptabilidad y entrega continua de mejoras. El enfoque iterativo y la constante retroalimentación con los usuarios clave fueron fundamentales para identificar y mitigar riesgos, garantizando que el sistema respondiera de manera eficiente a los requerimientos del Centro Odontológico Velasteguí.
- El análisis de los requerimientos permitió el desarrollo de un sistema que cumple con las necesidades específicas tanto de los pacientes como de los profesionales del centro. La identificación clara de las funcionalidades críticas, tales como la gestión eficiente de citas y el manejo de historias clínicas, fue esencial para mejorar la calidad del servicio y optimizar los procesos internos.
- La arquitectura propuesta, basada en una estructura modular que utiliza Laravel, MySQL y Angular, facilitó la creación de un sistema robusto y flexible. Esta arquitectura permitió una separación eficiente entre la lógica de negocio y la presentación de los datos, garantizando la seguridad, integridad y disponibilidad de la información, además de proporcionar una experiencia de usuario intuitiva y fluida.
- El proceso de implementación del sistema en el Centro Odontológico Velasteguí fue satisfactorio. La capacitación brindada al personal facilitó una transición adecuada del proceso manual a uno automatizado, mejorando significativamente la gestión de citas y la actualización de las historias clínicas, lo que a su vez impactó de manera positiva en la atención a los pacientes.

## RECOMENDACIONES

- Se recomienda que el Centro Odontológico Velasteguí asegure el pago anual del servicio de alojamiento web (hosting). Este pago es esencial para garantizar que el sistema web continúe siendo accesible y funcional de manera ininterrumpida. El centro debe priorizar este gasto en su presupuesto operativo, ya que la funcionalidad del sistema depende directamente de la disponibilidad continua del hosting, asegurando que tanto los pacientes como el personal puedan acceder y utilizar el sistema de manera eficiente durante todo el año.
- Es fundamental que el personal del Centro Odontológico Velasteguí reciba capacitación continua para optimizar el uso del sistema. Esto permitirá no solo la correcta gestión de las citas y las historias clínicas, sino también aprovechar todas las funcionalidades que ofrece la aplicación, con el fin de mejorar la atención a los pacientes.
- Se recomienda establecer un plan de mantenimiento periódico para asegurar que el sistema web continúe operando de manera eficiente y que se actualice las funcionalidades según sea necesario. Esto permitirá mantener la vigencia tecnológica y la adaptabilidad del sistema a posibles cambios en la normativa odontológica o en las necesidades del centro.

## BIBLIOGRAFÍA

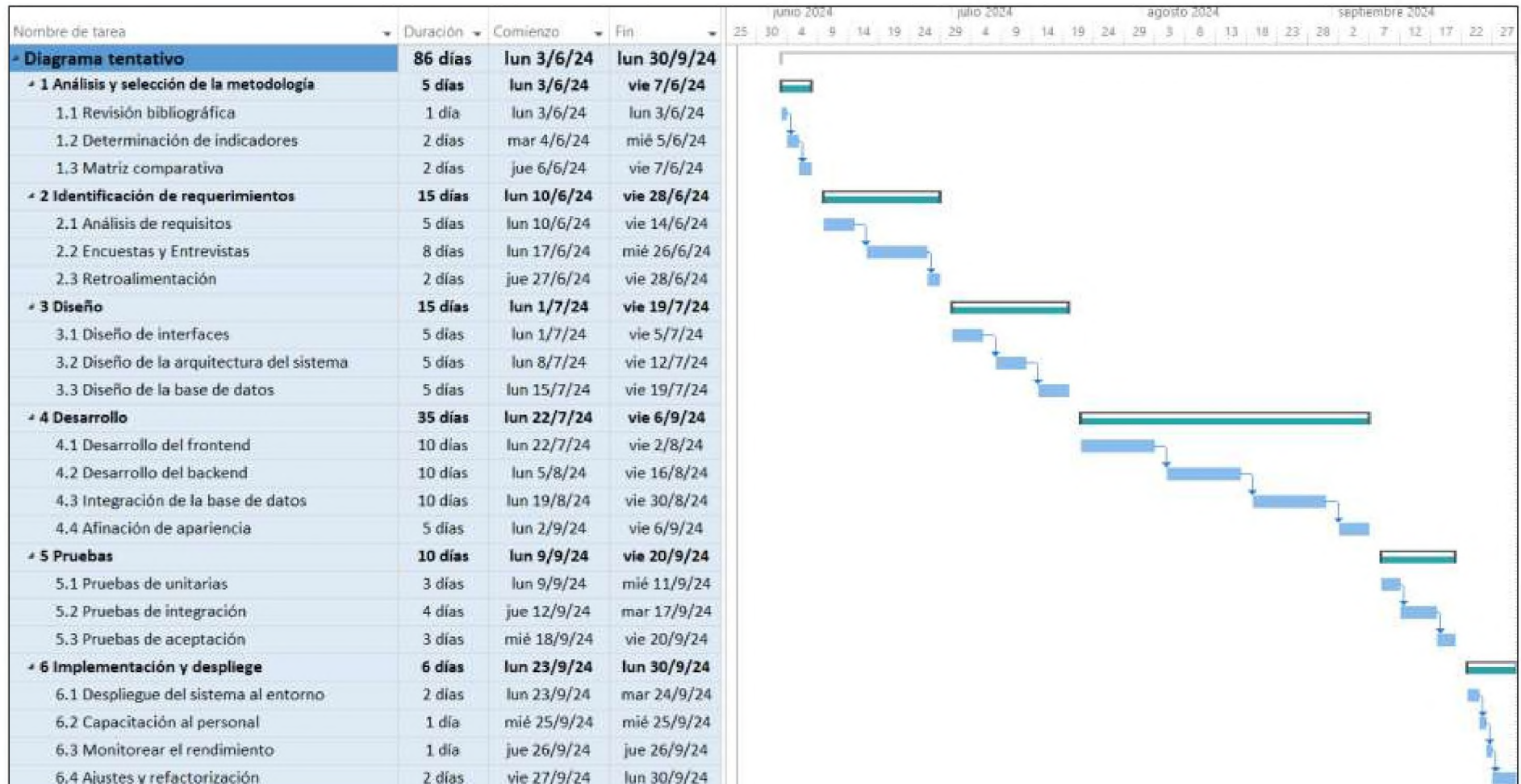
- Bello, E. (2021). Stakeholders: quiénes son, por qué son importantes y cómo gestionarlos. *Thinking for Innovation*.  
<https://www.iebschool.com/blog/stakeholders-quienes-son-digital-business/>
- Cabello, J. P. (2021, mayo 19). Conceptos básicos de Agile: cómo adaptarse a esta metodología y no morir en el intento. *Conferencia virtual "Jornadas Trabajo IT & Sistemas"*.
- Daza-Cardona, J. A., Vargas-Ramírez, J., & Guapacha-Sánchez, M. A. (2021). Doing odontograms and dentists in the classroom. Materiality and affect in dental education. *Tapuya: Latin American Science, Technology and Society*, 4(1).  
<https://doi.org/10.1080/25729861.2021.1968635>
- Dean, J., & Dean Raymond. (2017). *Introducción a la programación con java* (McGraw-Hil, Vol. 2).
- Denning, S. (2015). Agile: It's time to put it to use to manage business complexity. *Strategy & Leadership*, 10-17.
- Engbreth, G. (s. f.). *PHP 8 Revealed: Use Attributes, the JIT Compiler, Union Types, and More for Web Development?* .
- Freire, M. (2018). *DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE HISTORIAS CLÍNICAS Y CONTROL DE CITAS ODONTOLÓGICAS PARA LA CLÍNICA "DENTAL HOUSE" EN EL SECTOR DE CONOCOTO USANDO LA METODOLOGÍA XP*. Universidad Tecnológica Israel.
- Hostinger. (2023). *Guía Básica de Hosting* (Hostinger).
- Malavé, K. J. (2022). *DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE HISTORIAS CLÍNICAS Y CONTROL DE CITAS ODONTOLÓGICAS PARA EL CONSULTORIO DENTAL "RODENTAL 3"*. Universidad Estatal Península de Santa Elena.
- Letelier, P., & Carmen Penadés, M. (2006). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. [www.agileuniverse.com](http://www.agileuniverse.com).
- López, J. (2007). *Domine PHP y MySQL (programación dinámica en el lado del servidor)* (1.ª ed.).
- Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA SANTA MARIA DE LOS BUENOS AIRES.
- Meijomil, S. (2024). ▷ *Qué es un framework, para qué sirve y ejemplos / InboundCycle*.  
<https://www.inboundcycle.com/diccionario-marketing-online/framework>
- Pinto, M. (2023). *APLICACIÓN WEB MÓVIL PARA LA GESTIÓN INTEGRAL DE LA CLÍNICA ODONTOLÓGICA "BRAULIO JOSUÉ ROJAS DENTAL STUDIO" UBICADO EN LA CIUDAD DE IBARRA*.

- Pressman, R. S. (2010). *Ingeniería del Software. Un Enfoque Practico*. 7. [www.FreeLibros.me](http://www.FreeLibros.me)
- Sánchez, A., & Para, L. (s. f.). *CALDAS-FACULTAD TECNOLÓGICA*.
- Schwaber, K., & Sutherland, J. (1991). *La Guía de Scrum TM*.
- Silberschatz, Abraham., Korth, H. F., & Sudarshan, S. (2002a). *Database system concepts*. McGraw-Hill.
- Silberschatz, Abraham., Korth, H. F., & Sudarshan, S. (2002b). *Database system concepts*. McGraw-Hill.
- Soler, I. (2019). *Bootstrap 4: Guía práctica para el desarrollo web responsivo* (Leanpub).
- Sunardi, A., & Suharjito. (2019). MVC architecture: A comparative study between laravel framework and slim framework in freelancer project monitoring system web based. *Procedia Computer Science*, 157, 134-141. <https://doi.org/10.1016/j.procs.2019.08.150>
- Tinoco Gómez, I. O., Pedro, I., Rosales López, P., & Salas Bacalla, I. J. (2010). SISTEMA E INFORMÁTICA. *Ind. data*, 13(2).

# **ANEXO N°1**

---

## **CRONOGRAMA TENTATIVO (GANTT)**



# **ANEXO N°2**

---

**PRESUPUESTO EJECUTADO**

<b>N°</b>	<b>Recursos</b>	<b>Cantidad</b>	<b>Tiempo (Meses)</b>	<b>Precio unitario</b>	<b>Total</b>
<b>1</b>	Computadora	2	4	\$80	\$160
<b>2</b>	Hosting	1	12	\$48	\$48
<b>3</b>	Impresora	1	4	\$50	\$50
<b>4</b>	Internet	1	4	\$30	\$120
<b>5</b>	Papelería y Suministros de Oficina	1	4	\$30	\$30
<b>6</b>	Movilización	1	4	\$50	\$50
	<b>Total</b>				<b>\$458</b>

# **ANEXO N°3**

---

**CARTA DE ACEPTACIÓN POR PARTE DEL  
CENTRO ODONTOLÓGICO VELASTEGUÍ**



**CENTRO ODONTOLÓGICO VELASTEGUI  
DR. CARLOS VELASTEGUI**

**Echeandía, 06 de mayo del 2024**

Señores

Jorge Alexander Cárdenas Coloma

Luis Ángel Herrera Cedeño

**Estudiantes de la Universidad Estatal de Bolívar**

Presente:

De mi consideración:

Nos complace hacer llegar la confirmación de la propuesta presentada por ustedes, para la implementación del **SISTEMA WEB BASADO EN UNA METODOLOGÍA ÁGIL PARA GESTIONAR CITAS E HISTORIAS CLÍNICAS EN EL CENTRO ODONTOLÓGICO VELASTEGUÍ, CANTÓN ECHEANDÍA.**

Confiamos en la experiencia y el profesionalismo por parte de su equipo de desarrollo de software, y esperamos que su propuesta contribuya significativamente al éxito de nuestra labor profesional brindada a la ciudadanía echeandiense. Extendemos un agradecimiento sincero por su compromiso y dedicación para llevar a cabo este proyecto de manera efectiva y eficiente. De la misma manera estamos dispuestos a trabajar en estrecha colaboración con su usted para lograr los objetivos establecidos.

Atentamente,

**Jr. Carlos Velastegui V.**  
ODONTOLOGO  
REGISTRO SENESCYE  
1005-2017-1860280

**Dr. Carlos Velastegui**  
C. I: 0202244539

**Propietario del Centro Odontológico Velastegui**

# **ANEXO N°4**

---

## **CÁLCULO DE LA MUESTRA**

Para calcular el tamaño de la muestra se usó la siguiente fórmula:

$$n = \frac{N \times Z^2 \times p \times (1 - p)}{(N - 1) \times E^2 + Z^2 \times p \times (1 - p)}$$

**Donde:**

- n = Tamaño de la muestra
- N = Tamaño de la población
- Z = Valor Z para el nivel de confianza (para un 90% es aproximadamente 1.645)
- p = Proporción estimada (si no se conoce, se usa 0.5 para maximizar el tamaño de la muestra)
- E = Margen de error (en decimal, por ejemplo, 10% = 0.10)

**Cálculo:**

Datos:

- Población (N) = 800
- Nivel de confianza = 90% (Valor Z = 1.645)
- Margen de error (E) = 10% (0.10)
- Proporción estimada (p) = 0.5 (por defecto)

**Fórmula aplicada:**

$$n = \frac{800 \times 1.654^2 \times 0.5 \times (1 - 0.5)}{(800 - 1) \times 0.10^2 + 1.654^2 \times 0.5 \times (1 - 0.5)}$$

$$n = \frac{800 \times 2.706025 \times 0.25}{799 \times 0.01 + 2.706025 \times 0.25}$$

$$n = \frac{800 \times 0.67650625}{7.99 + 0.67650625}$$

$$n = \frac{5.41.205}{8.6665065}$$

$$n = 62.5$$

Redondeando, por lo tanto, el tamaño de la muestra es de 63 personas.

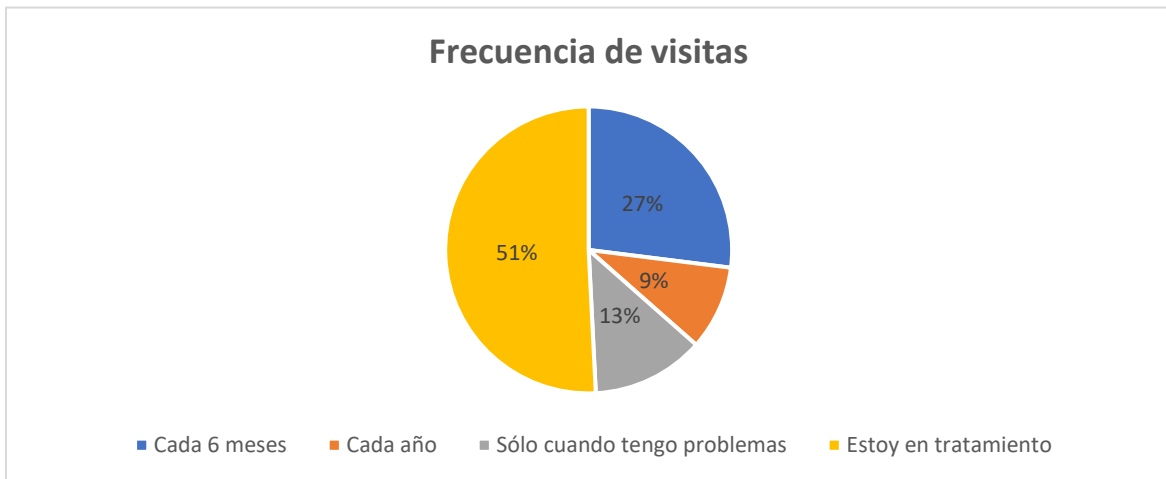
# **ANEXO N°5**

---

**ANÁLISIS Y TABULACIÓN DE RESULTADOS**

## Frecuencia de visitas

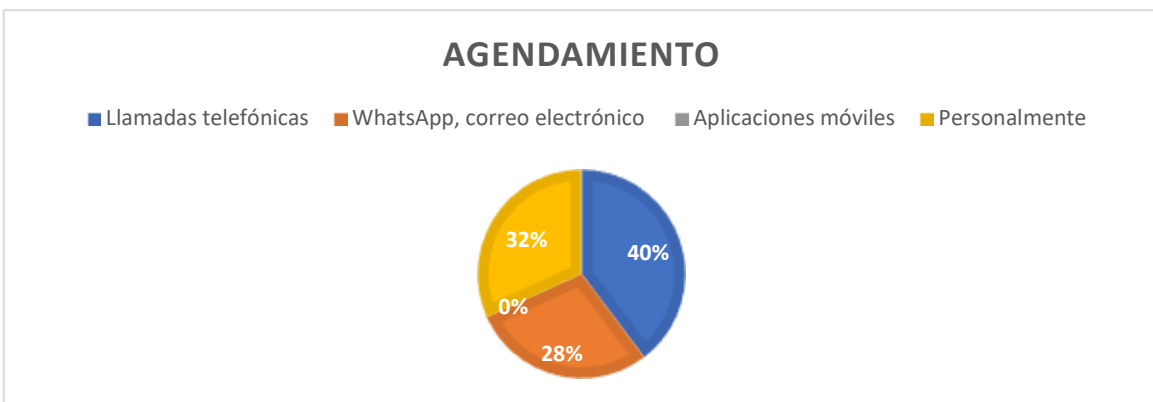
- ¿Con qué frecuencia visita el Centro Odontológico Velastegui?



La mayoría de los encuestados (51%) está en tratamiento activo, lo que indica una alta demanda de servicios continuos en el centro. Un 27% de los pacientes sigue un régimen regular de cuidado preventivo dental, visitando el centro cada seis meses. Un 9% visita el centro anualmente, mientras que un 13% solo busca atención dental en situaciones de problemas específicos. Estos datos reflejan una combinación de pacientes con necesidades de tratamiento continuo y aquellos que mantienen visitas preventivas regulares.

## Proceso de agendamiento

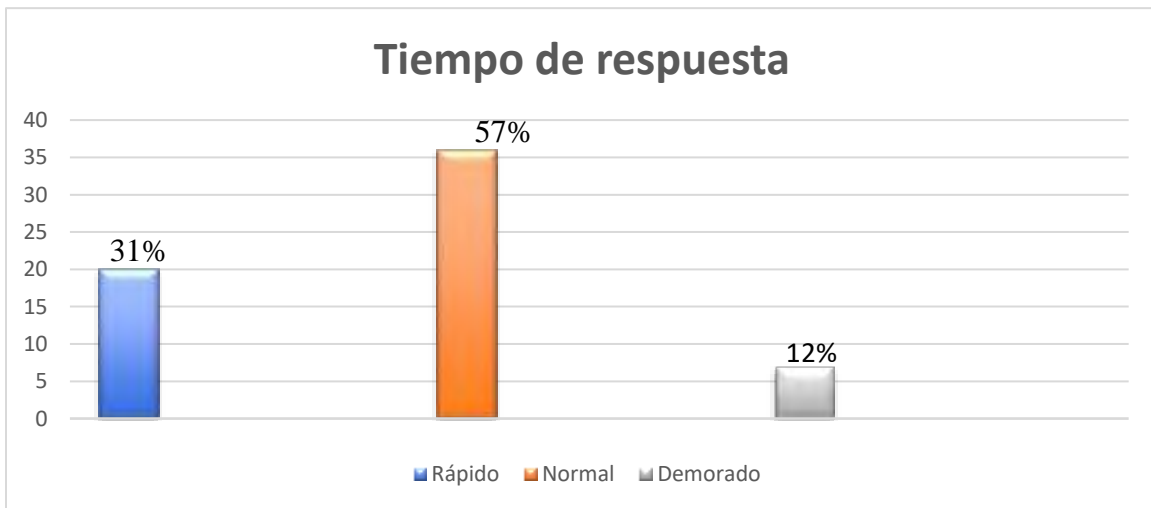
- ¿Cómo realiza el proceso de agendamiento de una cita odontológica?



La mayoría de los encuestados 40% prefiere utilizar llamadas telefónicas para agendar sus citas odontológicas, seguido por un 32% que prefiere hacerlo personalmente. Un 28% de los encuestados utiliza medios digitales como WhatsApp y correo electrónico para agendar sus citas. Ninguno de los encuestados utiliza páginas web o aplicaciones móviles para este propósito. Estos resultados indican que, aunque los medios tradicionales como las llamadas telefónicas

### Tiempo de respuesta

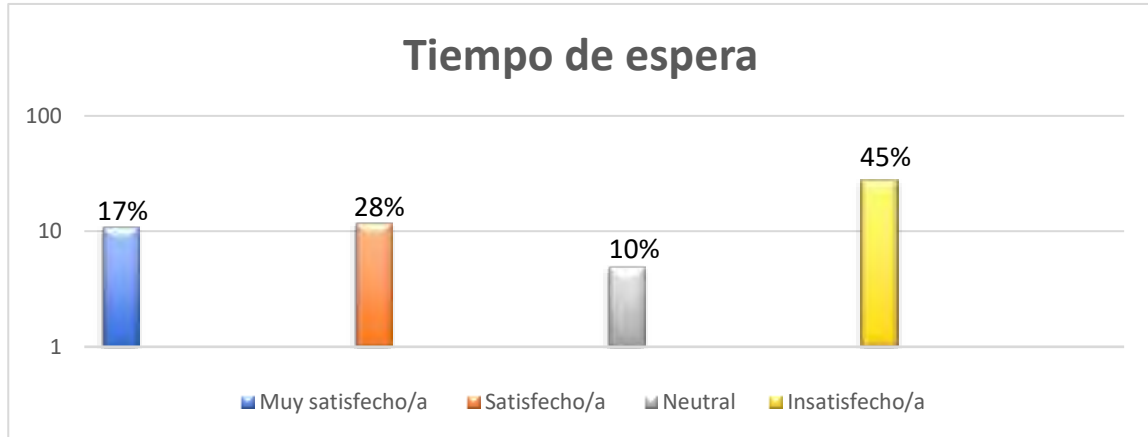
- ¿Cómo califica el tiempo de respuesta a la solicitud de una cita odontológica?



La mayoría de los encuestados (57%) considera que el tiempo de respuesta a la solicitud de una cita odontológica es normal, mientras que un 31% lo califica como rápido. Un menor porcentaje, 12%, considera que el tiempo de respuesta es demorado. Estos resultados indican que la mayoría de los pacientes tiene una percepción positiva o neutra sobre la eficiencia del proceso de agendamiento, con una pequeña proporción de pacientes experimentando demoras en la respuesta.

### Satisfacción con la puntualidad y tiempo de espera

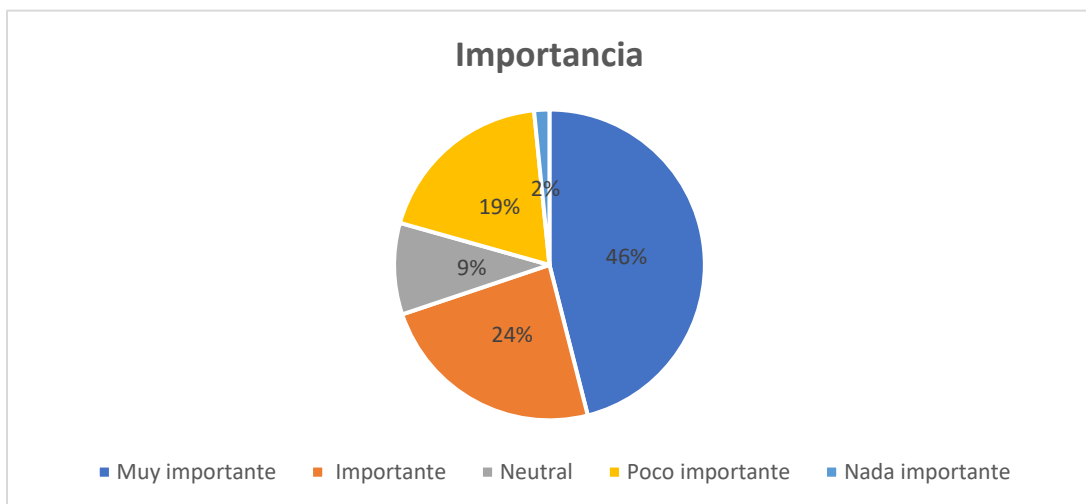
- ¿Cuál es su nivel de satisfacción con la puntualidad y tiempo de espera para ser atendido por el personal?



El 17% de los pacientes se siente muy satisfecho/a, reflejando un grupo que está completamente contento con el servicio. Un 28% está satisfecho/a, sugiriendo que hay una actitud positiva general pero también oportunidades para mejorar. El 10% se encuentra en una posición neutral, sin una opinión fuerte sobre el servicio. Sin embargo, un 45% de los pacientes se siente insatisfecho/a, indicando preocupaciones significativas y experiencias negativas relacionadas con la calidad

### Importancia del uso de tecnologías

- ¿Qué tan importante es para usted el uso de tecnologías más eficaces en la administración de un Centro Odontológico?



La mayoría de los encuestados considera que el uso de tecnologías más eficaces en la administración de un centro odontológico es **muy importante** (46%), con un 24% que lo considera **importante**. Solo un 9% se encuentra en una posición **neutral** sobre este tema, lo que sugiere una opinión indiferente o no bien formada. Un 19% de los encuestados piensa que es **poco importante**, y solo un 2% lo considera **nada importante**. Estos resultados muestran que una gran mayoría valora la implementación de tecnologías eficaces, reflejando una clara inclinación hacia la modernización y mejora de la gestión del centro odontológico. La importancia atribuida a estas tecnologías subraya la necesidad de considerar su integración para optimizar la administración y potencialmente mejorar la eficiencia y calidad del servicio.

### Aprobación del uso de un sistema automatizado

- ¿Le parece apropiado que el Centro Odontológico Velastegui haga uso de un sistema automatizado para gestionar los procesos de agendamientos de citas y manejo de historiales clínicos?

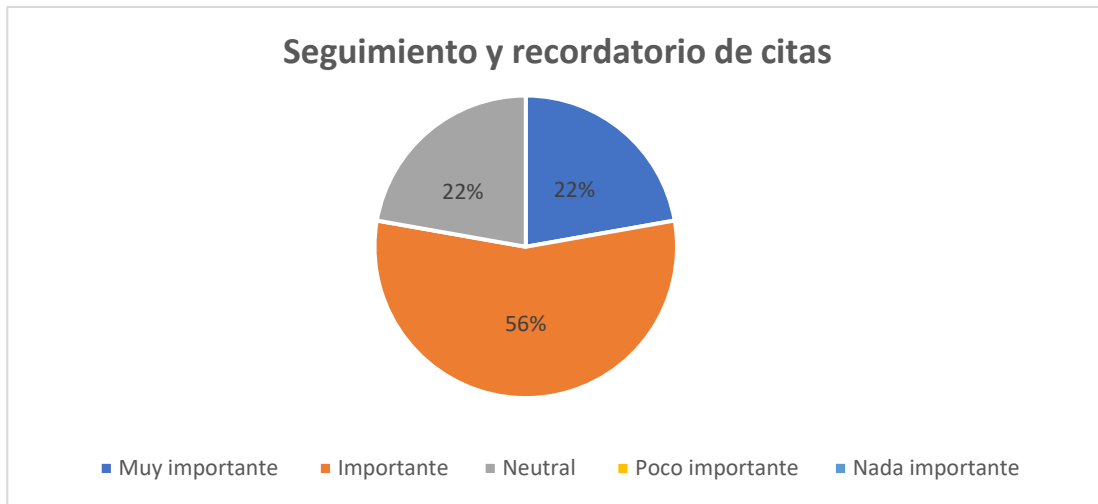


La gran mayoría de los encuestados (98%) considera que es apropiado que el Centro Odontológico Velastegui implemente un sistema automatizado para gestionar los procesos de agendamientos de citas y manejo de historiales clínicos, mientras que solo un 2% no está seguro. La abrumadora preferencia por el uso de un sistema automatizado sugiere que los pacientes valoran la eficiencia y la organización que puede proporcionar una solución tecnológica en estos aspectos. Esta opinión generalizada a favor de la

automatización destaca una fuerte demanda por mejoras en la gestión del centro, que podrían resultar en una mayor satisfacción del paciente y una operación más fluida y eficiente.

### Importancia del seguimiento y recordatorio de citas

- ¿Qué tan importante es para usted que este Centro Odontológico ofrezca un servicio de seguimiento y recordatorio de citas mediante correo electrónico y/o WhatsApp?



Un 22.22% de los encuestados considera que es **muy importante** que el Centro Odontológico ofrezca un servicio de seguimiento y recordatorio de citas mediante correo electrónico y/o WhatsApp, mientras que un 55.56% lo considera **importante**. Un 22.22% se encuentra en una posición **neutral**, sin una opinión fuerte sobre el tema. Estos resultados reflejan una significativa preferencia por la implementación de sistemas de recordatorio digital, indicando que una gran mayoría

# **ANEXO N°6**

---

## **ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE (SRS)**

## 1. Introducción

### 1.1 Propósito

El propósito de este documento es detallar los requisitos del sistema web para gestionar citas e historias clínicas en el Centro Odontológico Velasteguí.

### 1.2 Alcance

El sistema permitirá a los administradores, doctores, secretarias y pacientes gestionar las citas y acceder a las historias clínicas de los pacientes.

### 1.3 Definiciones, Acrónimos y Abreviaturas

- Administrador: Usuario con permisos completos en el sistema.
- Doctor: Usuario que puede acceder a historias clínicas y gestionar citas.
- Secretaria: Usuario que puede registrar pacientes, doctores y citas.
- Paciente: Usuario que puede gestionar sus propias citas y ver información relevante.

## 2. Requerimientos Funcionales

### Requerimiento Funcional RF-01

ID	Descripción	Usuarios
RF-01	Iniciar Sesión	Administrador, doctor, secretaria, paciente

### Requerimiento Funcional RF-02

ID	Descripción	Usuarios
RF-02	Registro de Pacientes	Administrador, secretaria

### Requerimiento Funcional RF-03

ID	Descripción	Usuarios
RF-03	Listado de Pacientes	Administrador, doctor, secretaria

### Requerimiento Funcional RF-04

ID	Descripción	Usuarios
RF-04	Registro de Doctores	Administrador, secretaria

### Requerimiento Funcional RF-05

ID	Descripción	Usuarios
RF-05	Listado de Doctores	Administrador, secretaria, paciente

### Requerimiento Funcional RF-06

ID	Descripción	Usuarios
RF-06	Registro de Horarios de los Doctores	Administrador, secretaria

**Requerimiento Funcional RF-07**

ID	Descripción	Usuarios
RF-07	Registro de Especialidades	Administrador, secretaria

**Requerimiento Funcional RF-08**

ID	Descripción	Usuarios
RF-08	Listado de Especialidades	Administrador, doctor, secretaria

**Requerimiento Funcional RF-09**

ID	Descripción	Usuarios
RF-09	Registro de Citas Médicas	Administrador, secretaria, paciente

**Requerimiento Funcional RF-10**

ID	Descripción	Usuarios
RF-10	Listado de Citas Médicas	Administrador, doctor, secretaria, paciente

**Requerimiento Funcional RF-11**

ID	Descripción	Usuarios
RF-11	Cancelación de Citas Médicas	Administrador, doctor, secretaria, paciente

**Requerimiento Funcional RF-12**

ID	Descripción	Usuarios
RF-12	Reagendamiento de Citas Médicas	Administrador, doctor, secretaria, paciente

**Requerimiento Funcional RF-13**

ID	Descripción	Usuarios
RF-13	Notificación de Citas Médicas	Administrador, doctor, secretaria, paciente

**Requerimiento Funcional RF-14**

ID	Descripción	Usuarios
RF-14	Manejo de Historias Clínicas	Administrador, doctor, secretaria

### 3. Requerimientos No Funcionales

#### Requerimiento No Funcional RNF-01

ID	Descripción	Detalles
RNF-01	Rendimiento	El sistema SWCH-OV debe ofrecer tiempos de respuesta rápidos para todas las operaciones, incluyendo la gestión de citas, el acceso a historias clínicas y la actualización de datos. Se requiere que el sistema pueda manejar múltiples solicitudes simultáneas sin afectar su rendimiento.

#### Requerimiento No Funcional RNF-02

ID	Descripción	Detalles
RNF-02	Escalabilidad	El sistema debe ser escalable para adaptarse al crecimiento del Centro Odontológico Velasteguí. Capacidad de manejar un aumento en el número de usuarios, pacientes y registros sin comprometer la funcionalidad.

#### Requerimiento No Funcional RNF-03

ID	Descripción	Detalles
RNF-03	Seguridad	El sistema debe implementar medidas de seguridad robustas para proteger la información sensible de los pacientes y los datos del centro. Uso de cifrado para la transmisión de datos y autenticación segura de usuarios.

#### Requerimiento No Funcional RNF-04

ID	Descripción	Detalles
RNF-04	Disponibilidad	El sistema debe estar disponible en todo momento durante las horas de operación del centro. Mecanismos para garantizar la continuidad del servicio y minimizar el tiempo de inactividad, incluyendo la realización de copias de seguridad periódicas.

#### Requerimiento No Funcional RNF-05

ID	Descripción	Detalles
RNF-05	Usabilidad	El sistema debe ser fácil de usar para todos los perfiles de usuario, con una interfaz intuitiva y accesible que permita realizar las tareas de manera eficiente

#### Requerimiento No Funcional RNF-06

ID	Descripción	Detalles
RNF-06	Mantenimiento	El sistema debe permitir un mantenimiento sencillo y efectivo, con la capacidad de realizar actualizaciones y correcciones sin interrumpir el servicio.

### 4. Requerimientos de Interfaz

#### Requerimiento de Interfaz RI-01

ID	Descripción	Detalles
RI-01	Interfaz de Usuario Web	El sistema debe ofrecer una interfaz web que se adapte a diferentes dispositivos y navegadores.

#### Requerimiento de Interfaz RI-02

ID	Descripción	Detalles
RI-02	Interfaz de Acceso	Interfaz de inicio de sesión que permita a los usuarios autenticar sus credenciales de manera segura y eficiente. Opciones para recuperar

		contraseñas y gestionar credenciales de forma sencilla.
--	--	---

### Requerimiento de Interfaz RI-03

ID	Descripción	Detalles
RI-03	Interfaz de Gestión de Datos	La interfaz para el registro y manejo de datos (pacientes, doctores, citas) debe permitir la entrada, modificación y visualización de información de manera efectiva.

## 5. Historias de usuario

*Tabla 1 Historia de usuario Iniciar Sesión.*

<b>Número:</b>	1
<b>Usuario:</b>	Administrador, doctor, secretaria, paciente
<b>Nombre historia:</b>	Iniciar Sesión
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Baja
<b>Puntos estimados:</b>	1
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Luis Herrera
<b>Descripción:</b>	El usuario puede iniciar sesión en el sistema proporcionando sus credenciales.
<b>Validación:</b>	Ninguna

*Tabla 2 Historia de usuario Registro de Pacientes.*

<b>Número:</b>	2
<b>Usuario:</b>	Administrador, secretaria
<b>Nombre historia:</b>	Registro de Pacientes
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Media
<b>Puntos estimados:</b>	2
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Jorge Cardenas
<b>Descripción:</b>	El sistema permite registrar nuevos pacientes en la base de datos.

<b>Validación:</b>	El paciente es registrado correctamente.
--------------------	--

**Tabla 3 Historia de usuario Listado de Pacientes.**

<b>Número:</b>	3
<b>Usuario:</b>	Administrador, doctor, secretaria
<b>Nombre historia:</b>	Listado de Pacientes
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Baja
<b>Puntos estimados:</b>	1
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Luis Herrera
<b>Descripción:</b>	El sistema muestra una lista de todos los pacientes registrados.
<b>Validación:</b>	La lista se muestra correctamente.

**Tabla 4 Historia de usuario Registro de Doctores.**

<b>Número:</b>	4
<b>Usuario:</b>	Administrador, secretaria
<b>Nombre historia:</b>	Registro de Doctores
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Media
<b>Puntos estimados:</b>	2
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Jorge Cardenas
<b>Descripción:</b>	El sistema permite registrar nuevos doctores en la base de datos.
<b>Validación:</b>	El doctor es registrado correctamente.

**Tabla 5 Historia de usuario Listado de Doctores.**

<b>Número:</b>	5
<b>Usuario:</b>	Administrador, secretaria, paciente
<b>Nombre historia:</b>	Listado de Doctores
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Baja
<b>Puntos estimados:</b>	1
<b>Iteración asignada:</b>	1

<b>Programador responsable:</b>	Luis Herrera
<b>Descripción:</b>	El sistema muestra una lista de todos los doctores registrados.
<b>Validación:</b>	La lista se muestra correctamente.

*Tabla 6 Historia de usuario Registro de Horarios de los Doctores.*

<b>Número:</b>	6
<b>Usuario:</b>	Administrador, secretaria
<b>Nombre historia:</b>	Registro de Horarios de los Doctores
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Media
<b>Puntos estimados:</b>	2
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Jorge Cardenas
<b>Descripción:</b>	El sistema permite registrar los horarios de los doctores.
<b>Validación:</b>	El horario es registrado correctamente.

*Tabla 7 Historia de usuario Registro de Especialidades.*

<b>Número:</b>	7
<b>Usuario:</b>	Administrador, secretaria
<b>Nombre historia:</b>	Registro de Especialidades
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Media
<b>Puntos estimados:</b>	2
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Luis Herrera
<b>Descripción:</b>	El sistema permite registrar nuevas especialidades en la base de datos.
<b>Validación:</b>	La especialidad es registrada correctamente.

*Tabla 8 Historia de usuario Listado de Especialidades.*

<b>Número:</b>	8
<b>Usuario:</b>	Administrador, doctor, secretaria
<b>Nombre historia:</b>	Listado de Especialidades
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Baja

<b>Puntos estimados:</b>	1
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Jorge Cardenas
<b>Descripción:</b>	El sistema muestra una lista de todas las especialidades registradas.
<b>Validación:</b>	La lista se muestra correctamente.

**Tabla 9 Historia de usuario Registro de Citas Médicas.**

<b>Número:</b>	9
<b>Usuario:</b>	Administrador, secretaria, paciente
<b>Nombre historia:</b>	Registro de Citas Médicas
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Media
<b>Puntos estimados:</b>	2
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Luis Herrera
<b>Descripción:</b>	El sistema permite registrar nuevas citas médicas.
<b>Validación:</b>	La cita es registrada correctamente.

**Tabla 10 Historia de usuario Listado de Citas Médicas.**

<b>Número:</b>	10
<b>Usuario:</b>	Administrador, doctor, secretaria, paciente
<b>Nombre historia:</b>	Listado de Citas Médicas
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Baja
<b>Puntos estimados:</b>	1
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Jorge Cardenas
<b>Descripción:</b>	El sistema muestra una lista de todas las citas médicas registradas.
<b>Validación:</b>	La lista se muestra correctamente.

**Tabla 11 Historia de usuario Cancelación de Citas Médicas.**

<b>Número:</b>	11
<b>Usuario:</b>	Administrador, doctor, secretaria, paciente
<b>Nombre historia:</b>	Cancelación de Citas Médicas

<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Media
<b>Puntos estimados:</b>	2
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Luis Herrera
<b>Descripción:</b>	El sistema permite cancelar citas médicas.
<b>Validación:</b>	La cita es cancelada correctamente.

*Tabla 12 Historia de usuario Reagendamiento de Citas Médicas.*

<b>Número:</b>	12
<b>Usuario:</b>	Administrador, doctor, secretaria, paciente
<b>Nombre historia:</b>	Reagendamiento de Citas Médicas
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Media
<b>Puntos estimados:</b>	2
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Jorge Cardenas
<b>Descripción:</b>	El sistema permite reagendar citas médicas.
<b>Validación:</b>	La cita es reagendada correctamente.

*Tabla 13 Historia de usuario Notificación de Citas Médicas.*

<b>Número:</b>	13
<b>Usuario:</b>	Administrador, doctor, secretaria, paciente
<b>Nombre historia:</b>	Notificación de Citas Médicas
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Baja
<b>Puntos estimados:</b>	1
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Luis Herrera
<b>Descripción:</b>	El sistema notifica a los usuarios sobre sus citas médicas.
<b>Validación:</b>	La notificación es enviada correctamente.

**Tabla 14** *Historia de usuario Manejo de Historias Clínicas.*

<b>Número:</b>	14
<b>Usuario:</b>	Administrador, doctor, secretaria
<b>Nombre historia:</b>	Manejo de Historias Clínicas
<b>Prioridad en la institución:</b>	Alta
<b>Riesgo en el desarrollo:</b>	Media
<b>Puntos estimados:</b>	2
<b>Iteración asignada:</b>	1
<b>Programador responsable:</b>	Jorge Cárdenas
<b>Descripción:</b>	El sistema permite gestionar las historias clínicas de los pacientes.
<b>Validación:</b>	La historia clínica es gestionada correctamente.

---

**ANEXO N°7 - MANUAL DE  
USUARIO: SISTEMA DE GESTIÓN  
DE CITAS E HISTORIAS CLÍNICAS  
DEL CENTRO ODONTOLÓGICO  
VELASTEGUÍ**

**“SIWCH-OV” V 1.0**

---



**CENTRO ODONTOLÓGICO VELASTEGUI**

*Elaborado Por: Jorge Alexander Cárdenas Coloma*

*Luis Ángel Herrera Cedeño*

# Contenido

1.	IMPLEMENTACIÓN DEL SISTEMA .....	109
1.1.	Doctores .....	109
1.2.	Secretaria .....	109
1.3.	Paciente: .....	109
2.	PANTALLA PRINCIPAL DEL SISTEMA .....	110
3.	INGRESO AL SISTEMA .....	110
4.	ACCESO.....	110
5.	ALERTAS DE SEGURIDAD .....	111
6.	MÓDULOS QUE INTEGRAN EL SISTEMA .....	115
6.1.	Roles y Permisos .....	115
6.2.	Personal .....	117
6.3.	Doctores .....	119
6.4.	Especialidades .....	122
6.5.	Pacientes .....	124
6.6.	Odontograma.....	126
6.7.	Citas Médicas .....	128

## 1. IMPLEMENTACIÓN DEL SISTEMA

### 1.1.Doctores

**Conocimientos Básicos de Informática:** Los doctores tengan un nivel básico de habilidades en informáticas.

**Navegación:** Los doctores estén familiarizados con la navegación básica en aplicaciones y sitios web.

### 1.2.Secretaria

**Conocimientos Básicos de Informática:** La secretaria debe poseer un nivel básico conocimientos informáticos.

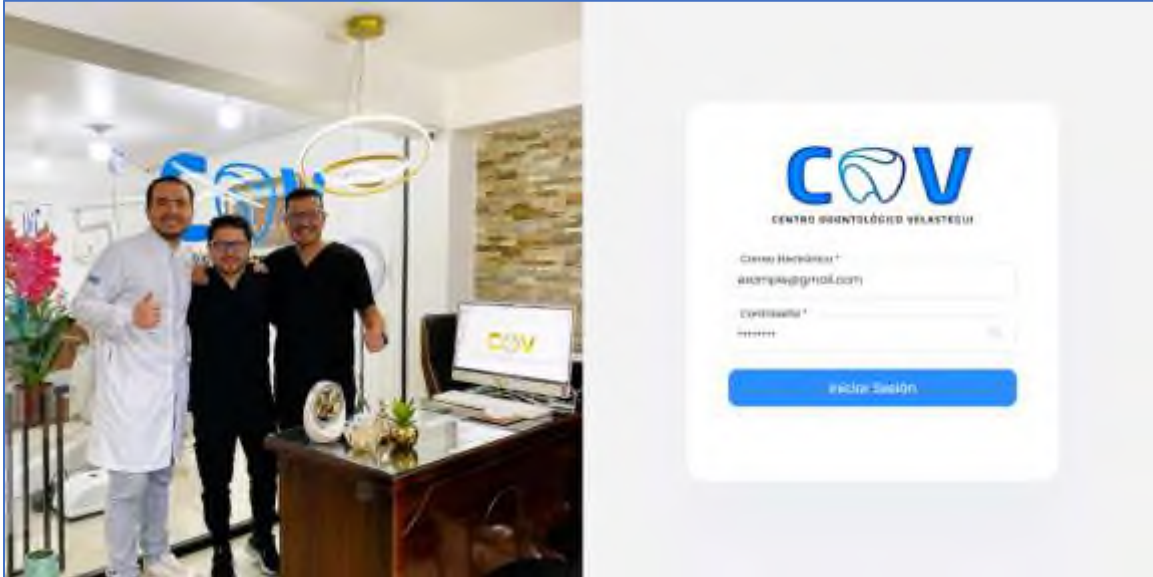
**Navegación:** La secretaria tiene experiencia en la navegación de aplicaciones y plataformas en línea.

### 1.3.Paciente:

**Conocimientos Básicos de Informática:** Los pacientes tengan un nivel básico de habilidades en informáticas.

**Navegación:** Los pacientes estén familiarizados con la navegación básica en aplicaciones y sitios web.

## 2. PANTALLA PRINCIPAL DEL SISTEMA



## 3. INGRESO AL SISTEMA

Para acceder a la aplicación, el usuario debe de hacer uso de sus credenciales de acceso (usuario y contraseña).

## 4. ACCESO

Una vez introducido el usuario y la contraseña, pulsar este botón **“Iniciar Sesión”** a los diferentes módulos que integran el sistema.

## 5. ALERTAS DE SEGURIDAD

Correo Electrónico \*



Ingrese el correo electrónico.

Se mostrará cuando el usuario intente continuar sin haber ingresado un correo electrónico.

Correo Electrónico \*



Por favor, ingrese una dirección válida.


Se mostrará cuando el usuario ingrese un correo electrónico en un formato incorrecto.

Contraseña \*



Ingrese la contraseña.

Se mostrará cuando el usuario intente continuar sin haber proporcionado una contraseña.

**Error!** EL usuario y contraseña son incorrectos. 

Se mostrará cuando el usuario intente iniciar sesión con credenciales incorrectas.

**Advertencia!** EXISTEN CAMPOS NECESARIOS.



Este tipo de advertencias aparecen cuando el usuario no ha completado correctamente uno o más campos obligatorios dentro de los formularios del sistema. Estas notificaciones sirven para guiar al usuario, indicando qué información falta o es incorrecta, y sugiriendo las acciones necesarias para corregir los errores

**Advertencia!** LAS CONTRASEÑA DEBEN SER IGUALES.



Se mostrará cuando el usuario intente confirmar una contraseña sin que ambas contraseñas coincidan.

**Error!** ES NECESARIO DARLE UN NOMBRE AL ROL Y SELECCIONAR UN PERMISO.



Se mostrará cuando el usuario intente crear o editar un rol sin haber asignado un nombre o permiso sin seleccionar al menos un permiso.



**¿Está seguro de eliminar al Doctor José Luis?**

NO

SI



**¿Está seguro de eliminar este usuario?**

NO

SI



**¿Está seguro de cancelar esta cita?**

NO

SI



**¿Está seguro de eliminar al paciente Alex Andrés?**

NO

SI

Estos tipos de alertas se mostrarán cuando el usuario intente eliminar o cancelar un registro, como un paciente, cita o cualquier otro elemento importante. El mensaje de la alerta será claro y directo. Cabe mencionar que existen restricciones que no permitirán que se puedan realizar la eliminación en algunos casos

! ES NECESARIO DARLE UN NOMBRE A LA ESPECIALIDAD.



Esta alerta aparece si se intenta crear una nueva especialidad sin ingresar un nombre. El sistema requiere que se proporcione un nombre para cada especialidad, ya que es un campo obligatorio para completar el registro.

**Advertencia!** El campo de C.L es obligatorio..

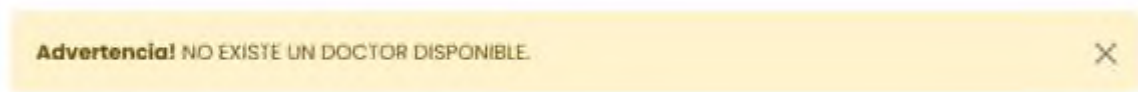


Se mostrará cuando el usuario intente continuar sin haber ingresado una cédula en un campo obligatorio.

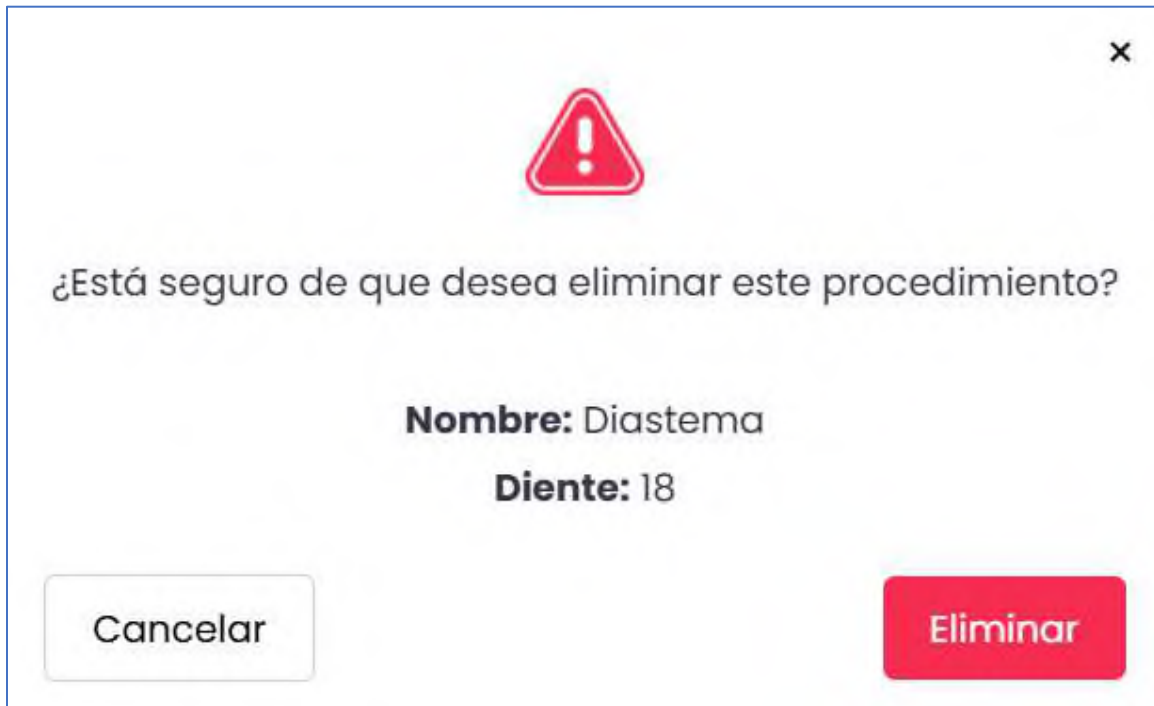
**Advertencia!** El paciente no existe en la base de datos. Por favor, registre al paciente antes de agendar la cita..



Se mostrará cuando el usuario intente agendar una cita insertando un campo de C.I incorrecto, entonces el sistema no reconocerá este valor y generará esta advertencia.



Se mostrará cuando el usuario intente agendar una cita que requiera la disponibilidad de un médico, pero no haya doctores disponibles en esa fecha, hora o especialidad.



Se mostrará cuando el usuario intente eliminar un registro de la lista de procedimientos. Al realizar esta acción, dicho procedimiento dejará de estar visible tanto en la lista de procedimientos como en el odontograma del paciente.

## 6. MÓDULOS QUE INTEGRAN EL SISTEMA

### 6.1. Roles y Permisos

#### 6.1.1. Agregar Roles y Permisos

Módulos	Permisos
Panel	<input type="checkbox"/> Administrador
Roles y Permisos	<input type="checkbox"/> Listado de Roles
	<input type="checkbox"/> Agregar Rol
	<input type="checkbox"/> Editar Rol
	<input type="checkbox"/> Eliminar Rol

Este sistema de citas e historiales clínicos se basa en el manejo de roles, mismo que son asignados a cada usuario existente, el mismo que contará con diversos permisos sobre el sistema, es decir cada usuario tendrá asignadas distintas funcionalidades dentro del sistema, pero así mismo, estará restringido para algunas.

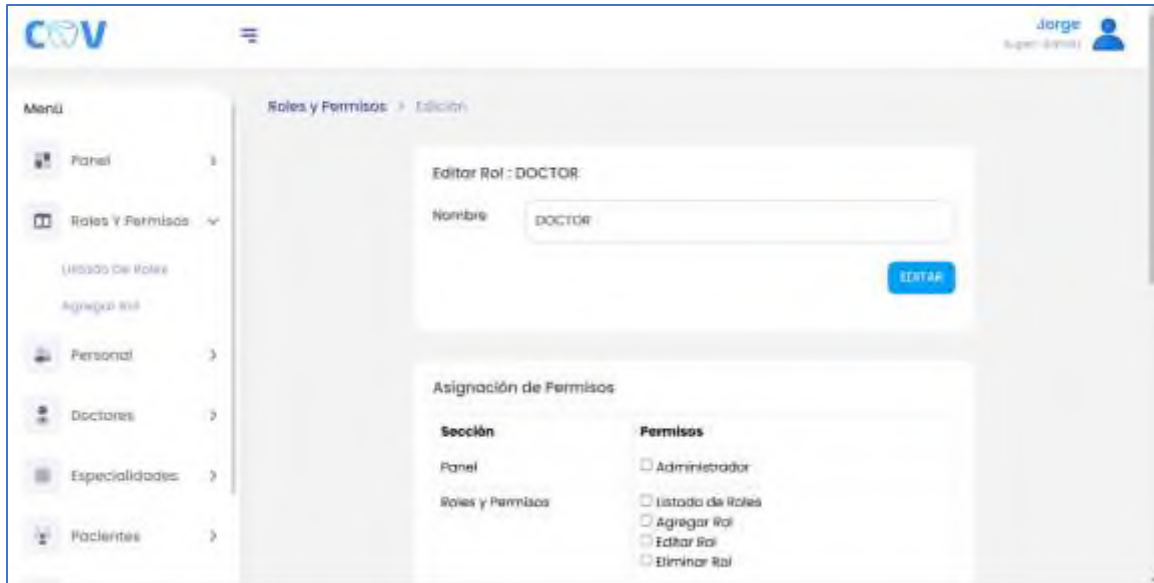
En esta página se hace el debido registro del nombre del rol y los permisos que se le desee agregar para su posterior aplicación.

#### 6.1.2. Listado de Roles y Permisos

Nombre	Permisos
SECRETARIA	register_doctor,list_doctor,edit_doctor,delete_doctor,profile_doctor,register_patient,list_patient,ed
PACIENTE	list_doctor,profile_doctor,register_appointment,list_appointment,list_specialty,calendar
DOCTOR	list_doctor,profile_doctor,list_patient,edit_patient,profile_patient,list_appointment,list_specialty,co
Super-Admin	register_rol,list_rol,edit_rol,delete_rol,register_doctor,list_doctor,edit_doctor,delete_doctor,profile_

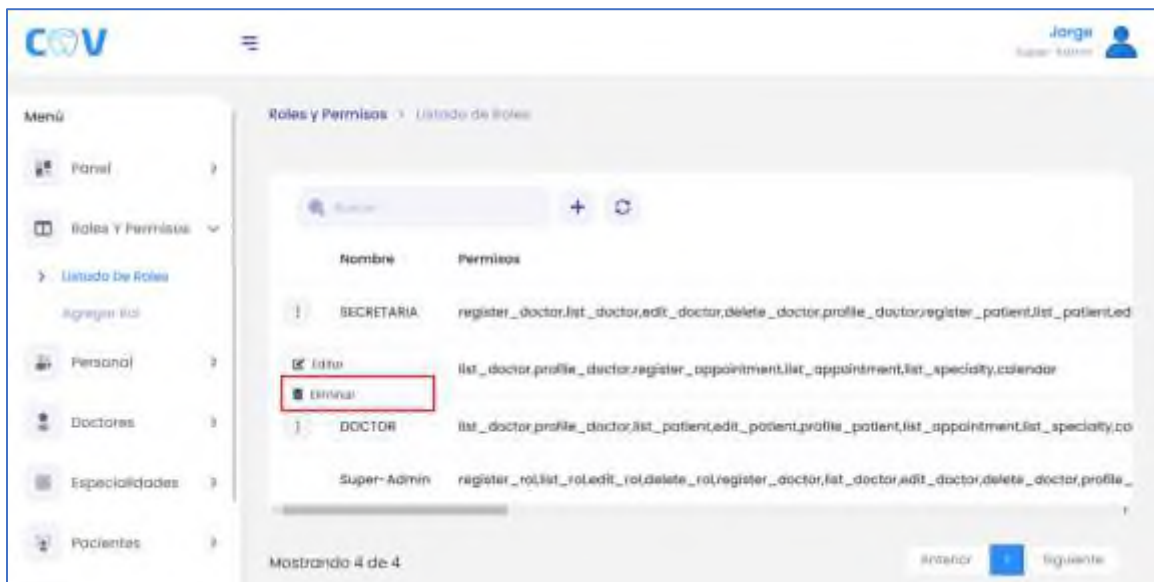
Esta interfaz nos muestra detalladamente cada rol existente en el sistema con sus respectivos permisos asignados, los cuales son gestionados estáticamente por el administrador, es decir, no cualquier usuario puede crearlos o modificarlos debido a que son funcionalidades propias del sistema.

### 6.1.3. Editar Roles y Permisos



En este formulario de edición, se pueden añadir o quitar permisos al rol existente, o a su vez, cambiar el nombre del rol por uno nuevo.

### 6.1.4. Eliminar Roles y Permisos



Esta funcionalidad sirve para eliminar definitivamente el rol existente, de la mano de todos sus permisos definidos, esta opción no siempre estará disponible debido a que esta tabla tiene una relación de tipo RESTRICT con otras tablas del sistema y no se pueden eliminar.

## 6.2. Personal

### 6.2.1. Agregar Personal

The screenshot shows the 'Agregar Personal' form in the COV system. The form is titled 'Agregar Personal' and is located in the 'Personal' section. The form includes the following fields:

- Nombre \*
- Apellidos \*
- Teléfono
- Correo Electrónico \*
- Contraseña \*
- Confirmar Contraseña \*
- Fecha de nacimiento
- Sexo: Masculino / Femenino
- Nivel de Educación
- Asignación
- Rol: Selección de rol

Este formulario está dirigido a recopilar los datos de un nuevo usuario de tipo secretaria o administrador, para almacenarlo en nuestra base de datos con sus respectivos permisos.

### 6.2.2. Listado del Personal

The screenshot shows the 'Listado del Personal' table in the COV system. The table displays a list of users with the following columns:

Nombre y Apellidos	Correo Electrónico	Rol	Teléfono
Ana Lucia Velastigui Vaca	anifa@gmail.com	SECRETARIA	0981273456
Jorge Cordinas	jorge@gmail.com	Super-Admin	0981833000

Mostrando 2 de 2

Anterior **1** Siguiente

Aquí se lista al personal registrado con sus debidos datos principales como el correo electrónico del usuario, su rol y su teléfono.

### 6.2.3. Editar Personal

The screenshot shows the 'Editar Usuario' form for Ana Lucia Vilastegui Vaca. The form includes fields for Name, Surname, Telephone, Email, Password, Confirm Password, Date of Birth, Gender, and Role. The role is currently set to 'SECRETARIA'.

Nombre *	Apellido *	Teléfono
Ana Lucia	Vilastegui Vaca	0981273456

Correo Electrónico *	Contraseña *
ana@gmail.com	

Confirmar Contraseña *	Fecha de Nacimiento
	10/31/2024

género	Nivel de Educación
<input checked="" type="radio"/> Masculino <input type="radio"/> Femenino	

Agrupación	Rol *
	SECRETARIA

Con este formulario se realiza una edición de los datos del usuario registrado, para poder actualizar su información y hacer uso de ella.

### 6.2.4. Eliminar Personal

The screenshot shows the 'Lista de Personal' table with columns for Names and Surnames, Email, Role, and Telephone. The user 'Ana Lucia Vilastegui Vaca' is listed with the role 'SECRETARIA'. A red box highlights the 'Eliminar' button for the user 'Benas'.

Nombre y Apellido	Correo Electrónico	Rol	Teléfono
Ana Lucia Vilastegui Vaca	ana@gmail.com	SECRETARIA	0981273456
Benas	jorge@gmail.com	Super-Admin	0981833000

Mostrando 2 de 2

Anterior **1** Siguiente

Esta opción va dirigida a retirar definitivamente el usuario, pero igual existen restricciones para su manejo.

## 6.3. Doctores

### 6.3.1. Agregar Doctores

The screenshot shows the 'Agregar Doctor' form in the COV system. The form is titled 'Agregar Doctor' and is located in the 'Doctores' section of the 'Agregar Doctor' menu. The form includes the following fields:

- Nombre \*
- Apellidos \*
- Teléfono
- Correo Electrónico \*
- Contraseña \*
- Confirmar Contraseña \*
- Fecha de nacimiento
- Género: Masculino / Femenino
- Dirección \*
- Roles: Seleccionar el rol
- Especialidades: Seleccionar Especialidad

La opción Agregar Doctor permite registrar un nuevo doctor en el sistema, pero requiere que se completen todos los campos obligatorios para finalizar el proceso. Entre estos campos, la selección de un horario es obligatoria. El sistema no permitirá que el doctor sea agregado si no se ha asignado un horario.

### 6.3.2. Agregar Horarios de los Doctores

The screenshot shows the 'Disponibilidad de horario' form in the COV system. The form is titled 'Disponibilidad de horario' and is located in the 'Doctores' section of the 'Agregar Doctor' menu. The form includes a grid of days and hours with checkboxes for selecting availability.

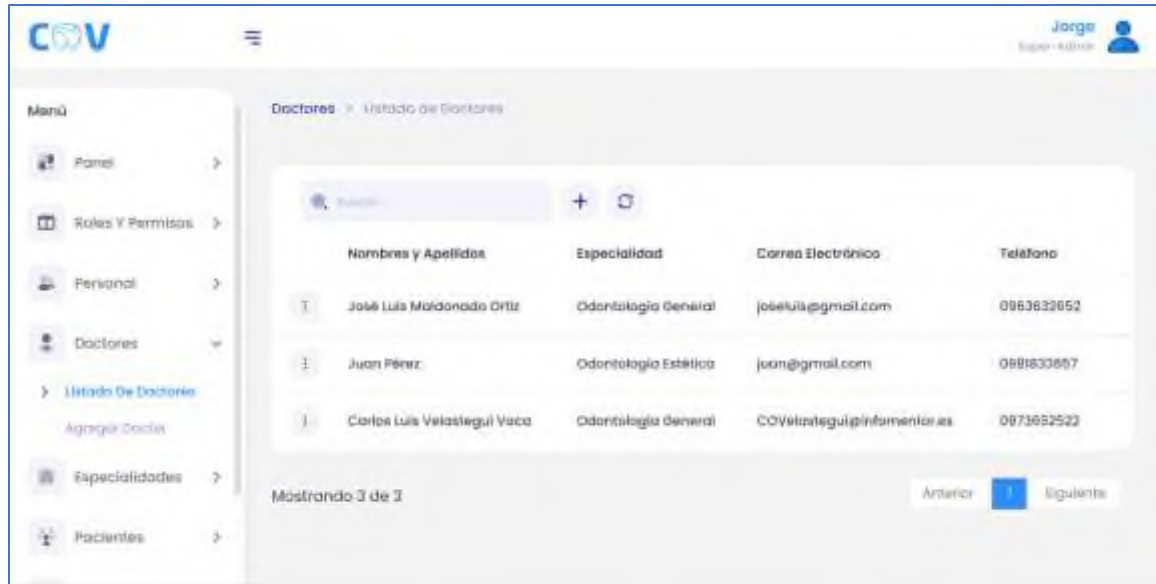
DIAS/HORAS	Lunes	Martes	Miércoles	Jueves	Viernes
<input type="checkbox"/> 08:00 AM	<input type="checkbox"/> 08:00 AM - TODOS	<input type="checkbox"/> 08:00 AM - TODOS	<input type="checkbox"/> 08:00 AM - TODOS	<input type="checkbox"/> 08:00 AM - TODOS	<input type="checkbox"/> 08:00 AM - TODOS
<input type="checkbox"/> 08:30 AM	<input type="checkbox"/> 08:00 AM - 08:30 AM <input type="checkbox"/> 08:30 AM - 09:00 AM	<input type="checkbox"/> 08:00 AM - 08:30 AM <input type="checkbox"/> 08:30 AM - 09:00 AM	<input type="checkbox"/> 08:00 AM - 08:30 AM <input type="checkbox"/> 08:30 AM - 09:00 AM	<input type="checkbox"/> 08:00 AM - 08:30 AM <input type="checkbox"/> 08:30 AM - 09:00 AM	<input type="checkbox"/> 08:00 AM - 08:30 AM <input type="checkbox"/> 08:30 AM - 09:00 AM
<input type="checkbox"/> 09:00 AM	<input type="checkbox"/> 09:00 AM - TODOS	<input type="checkbox"/> 09:00 AM - TODOS	<input type="checkbox"/> 09:00 AM - TODOS	<input type="checkbox"/> 09:00 AM - TODOS	<input type="checkbox"/> 09:00 AM - TODOS
<input type="checkbox"/> 09:30 AM	<input type="checkbox"/> 09:00 AM - 09:30 AM <input type="checkbox"/> 09:30 AM - 10:00 AM	<input type="checkbox"/> 09:00 AM - 09:30 AM <input type="checkbox"/> 09:30 AM - 10:00 AM	<input type="checkbox"/> 09:00 AM - 09:30 AM <input type="checkbox"/> 09:30 AM - 10:00 AM	<input type="checkbox"/> 09:00 AM - 09:30 AM <input type="checkbox"/> 09:30 AM - 10:00 AM	<input type="checkbox"/> 09:00 AM - 09:30 AM <input type="checkbox"/> 09:30 AM - 10:00 AM
<input type="checkbox"/> 10:00 AM	<input type="checkbox"/> 10:00 AM - TODOS	<input type="checkbox"/> 10:00 AM - TODOS	<input type="checkbox"/> 10:00 AM - TODOS	<input type="checkbox"/> 10:00 AM - TODOS	<input type="checkbox"/> 10:00 AM - TODOS
<input type="checkbox"/> 10:30 AM	<input type="checkbox"/> 10:00 AM - 10:30 AM <input type="checkbox"/> 10:30 AM - 11:00 AM	<input type="checkbox"/> 10:00 AM - 10:30 AM <input type="checkbox"/> 10:30 AM - 11:00 AM	<input type="checkbox"/> 10:00 AM - 10:30 AM <input type="checkbox"/> 10:30 AM - 11:00 AM	<input type="checkbox"/> 10:00 AM - 10:30 AM <input type="checkbox"/> 10:30 AM - 11:00 AM	<input type="checkbox"/> 10:00 AM - 10:30 AM <input type="checkbox"/> 10:30 AM - 11:00 AM

#### **Nota Importante:** Selección de Horario

Al agregar un nuevo doctor al sistema, es esencial seleccionar un horario para que el registro sea completado correctamente. El sistema requiere que se asigne un horario específico para cada doctor para asegurar una correcta organización y disponibilidad.

Si no se selecciona un horario, el sistema no permitirá que el doctor sea registrado, mostrando una alerta que indicará la necesidad de completar este campo antes de continuar.

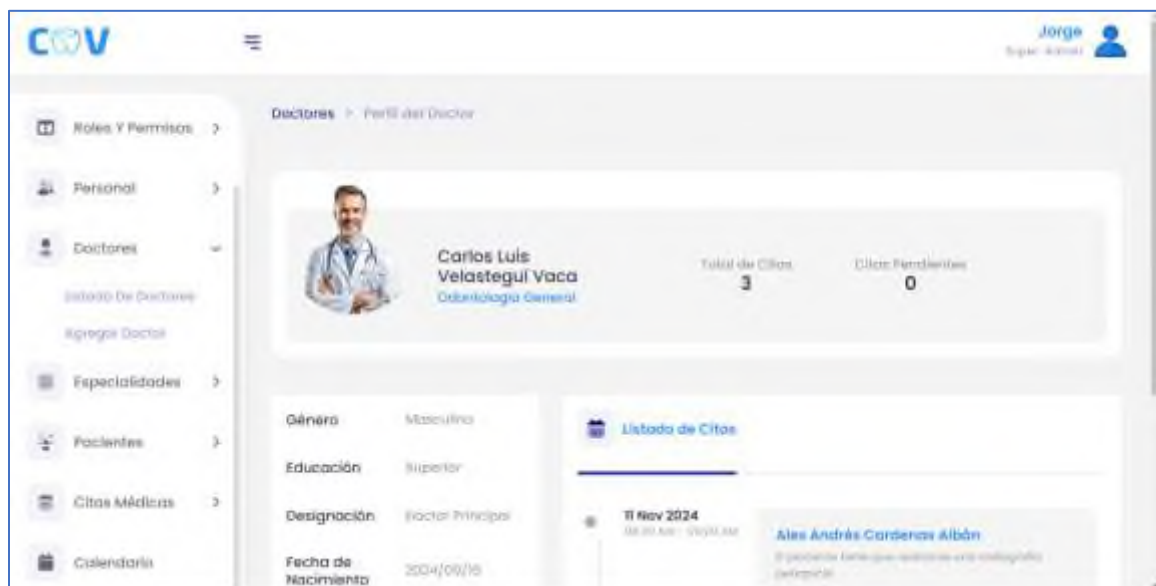
### 6.3.3. Listado de Doctores



Nombres y Apellidos	Especialidad	Correo Electrónico	Teléfono
José Luis Maldonado Ortiz	Odontología General	joselula@gmail.com	0883832652
Juan Pérez	Odontología Estética	juan@gmail.com	0883832657
Carlos Luis Velastegui Vaca	Odontología General	COVelastegui@gmail.com	0873652522

El Listado de Doctores muestra todos los doctores registrados en el sistema. En esta sección, se puede visualizar información esencial sobre cada doctor, como su nombre, especialidad, y datos de contacto. Además, en esta interfaz se tiene el acceso al perfil de este doctor.

### 6.3.4. Perfil del Doctor



**Carlos Luis Velastegui Vaca**  
Odontólogo General

Total de Citas: **3**      Citas Pendientes: **0**

Género: Masculino  
Educación: Superior  
Designación: Doctor Principal  
Fecha de Nacimiento: 2004/09/16

**11 Nov 2024**  
08:30 AM - 09:00 AM  
**Alex Andrés Cordero Albán**  
El paciente tiene que ser atendido por odontología general.

En esta pantalla se visualizará todos los datos del doctor en cuestión, así como su disponibilidad de días y listado de citas.

### 6.3.5. Editar Doctor





The screenshot shows the 'Editar Doctor' form in the COV system. The form is titled 'Editar Doctor : Carlos Luis' and contains several input fields for personal and professional information. The fields are organized into a grid-like structure. The left sidebar contains a menu with options like 'Roles Y Permisos', 'Personal', 'Doctores', 'Unidad De Doctores', 'Especialidades', 'Pacientes', 'Citas Médicas', and 'Calendario'. The main content area shows the following fields:

- Nombre \* (Carlos Luis)
- Apellido \* (Velastegui Vaca)
- Teléfono (0973652522)
- Correo Electrónico \* (COVelastegui@intomental.es)
- Continuar Correo \* (empty)
- Fecha de Nacimiento (9/12/2024)
- Género (Masculino)
- Nivel de Educación (Superior)
- Asignación (Doctor Principal)
- ROL \* (DOCTOR)
- Especialidad \* (Odontología General)


La interfaz permite modificar la información de un médico de forma sencilla, mostrando un formulario prellenado con datos como nombre completo, especialidad, entre otros. Los campos son intuitivos y editables, organizados de manera clara para facilitar su actualización.

### 6.3.6. Eliminar Doctor

The screenshot shows the 'Unidad de Doctores' page in the COV system. The page displays a table of doctors with columns for 'Nombres y Apellidos', 'Especialidad', 'Correo Electrónico', and 'Teléfono'. The 'Eliminar' button for the second doctor is highlighted with a red box.

	Nombres y Apellidos	Especialidad	Correo Electrónico	Teléfono
				
	José Luis Maldonado Ortiz	Odontología General	jose Luis@gmail.com	0983632652
	Juan	Odontología Estéticas	Juan@gmail.com	098833657
PERFIL DE DOCTOR				
	Carlos Luis Velastegui Vaca	Odontología General	COVelastegui@intomental.es	0973652522

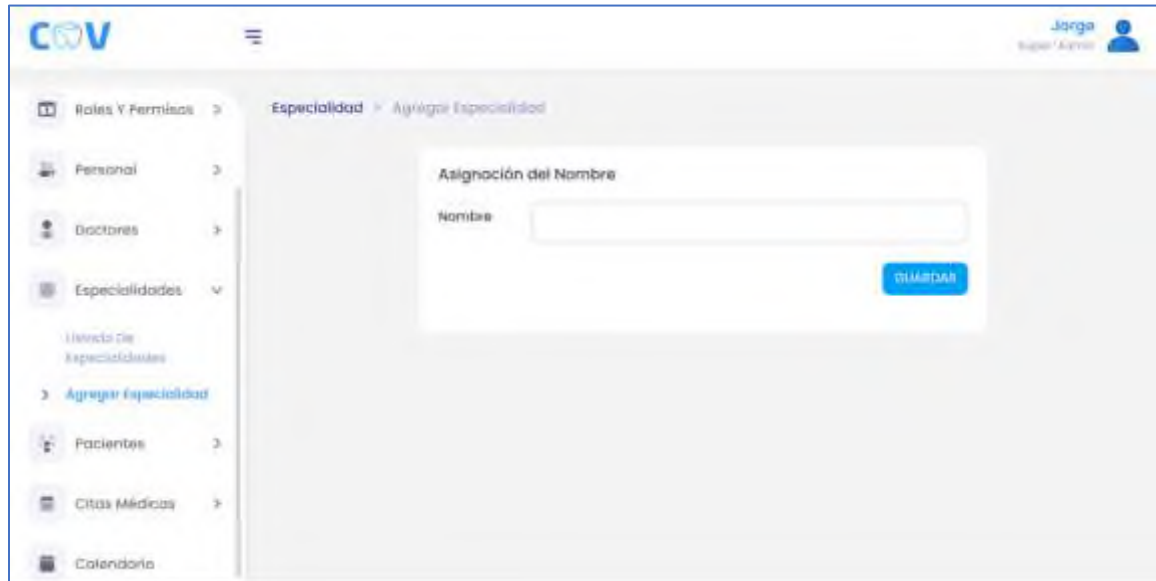
Mostrando 3 de 3

Anterior  Siguiente

Permite al usuario borrar el registro de un médico de forma rápida y segura, mostrando un mensaje de confirmación destacando los datos clave del médico a ser eliminado.

## 6.4.Especialidades

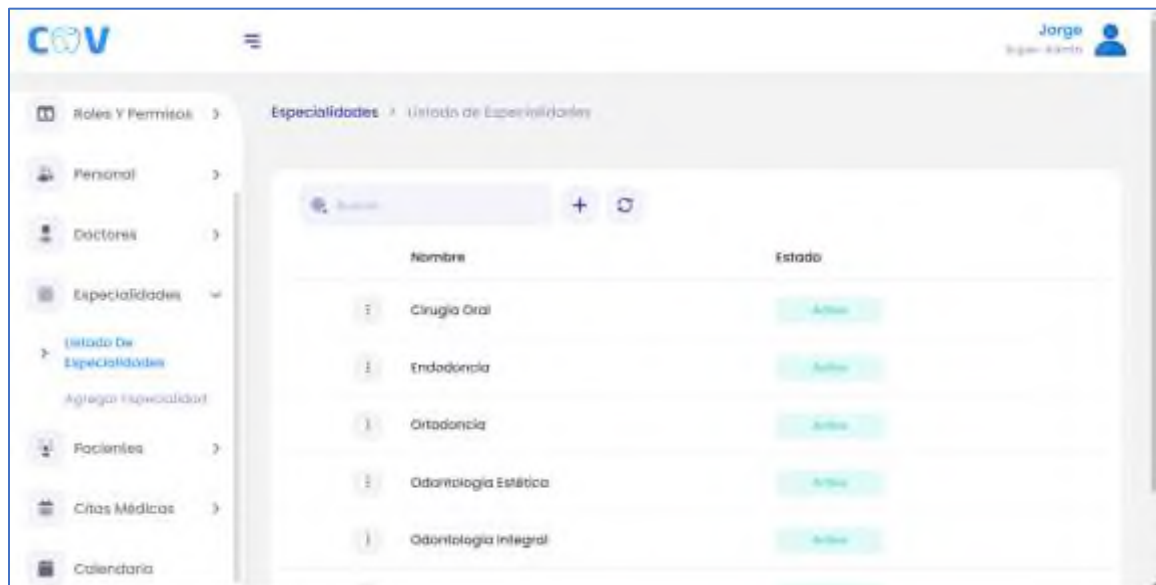
### 6.4.1. Agregar Especialidades



The screenshot shows the 'Agregar Especialidad' (Add Speciality) form in the COV system. The form is titled 'Asignación del Nombre' (Name Assignment) and contains a single text input field labeled 'Nombre' (Name). A blue 'GUARDAR' (Save) button is located at the bottom right of the form. The left sidebar shows the navigation menu with 'Especialidades' selected, and the breadcrumb trail indicates 'Especialidad > Agregar Especialidad'.

La opción de agregar nuevas especialidades permite introducir una nueva especialidad en el sistema. Para completar este proceso, solo se requiere ingresar el nombre de la especialidad. Una vez registrado, el nuevo nombre de especialidad estará disponible para ser asignado a los doctores en el sistema.

### 6.4.2. Listado de Especialidades

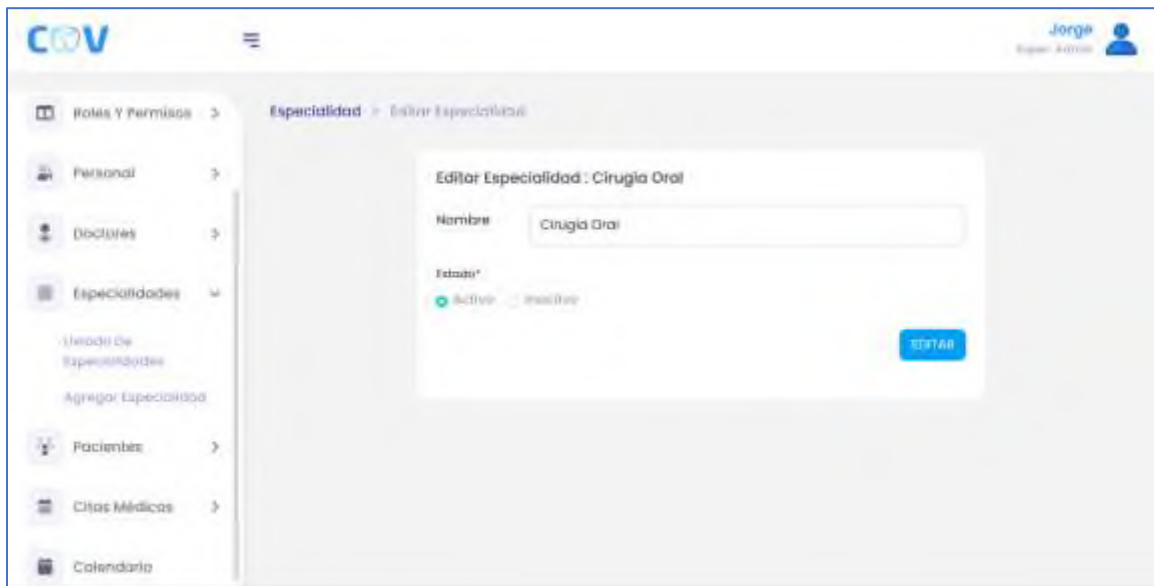


The screenshot shows the 'Listado de Especialidades' (Specialities List) in the COV system. The table displays a list of registered specialities with their names and states. The table has two columns: 'Nombre' (Name) and 'Estado' (State). The 'Estado' column contains green buttons labeled 'Activo' (Active). The left sidebar shows the navigation menu with 'Especialidades' selected, and the breadcrumb trail indicates 'Especialidades > Listado de Especialidades'.

Nombre	Estado
Cirugía Oral	Activo
Endodoncia	Activo
Ortodoncia	Activo
Odonatología Estética	Activo
Odonatología Integral	Activo

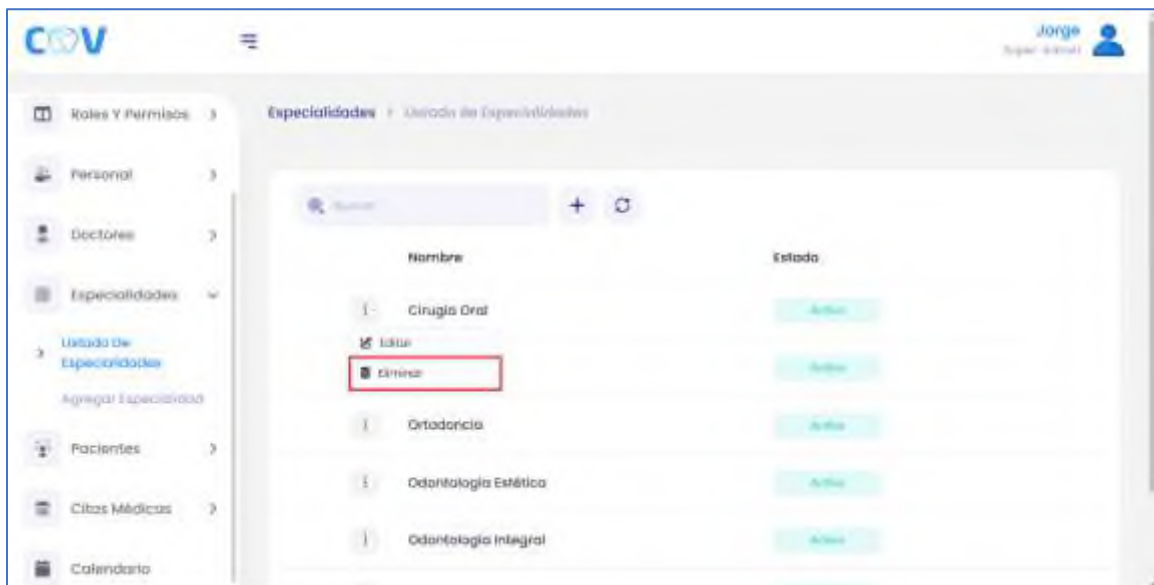
El Listado de Especialidades muestra todas las especialidades registradas en el sistema, incluyendo el nombre de cada especialidad y su estado (activo o inactivo).

### 6.4.3. Editar Especialidades



Aquí se puede editar la información de la especialidad, como el estado y su nombre.

### 6.4.4. Eliminar Especialidades



Permite al usuario eliminar una especialidad médica del sistema de manera sencilla y segura. Esto incluye un mensaje de advertencia que destaque el nombre de la especialidad seleccionada. Cabe recalcar que, si existe una relación de esta especialidad con un doctor, dicha especialidad no podrá ser eliminada, ya que un doctor está haciendo uso de ella.

## 6.5. Pacientes

### 6.5.1. Agregar Pacientes

The screenshot shows the 'Agregar Paciente' form in the COV system. The form is titled 'Agregar Paciente' and is located in the 'Pacientes' section. The form includes the following fields:

- Nombre \*
- Apellido \*
- Teléfono
- Correo Electrónico \*
- Contraseña \*
- Confirmar Contraseña \*
- Fecha de Nacimiento
- Sexo:  Masculino  Femenino
- Nivel de Educación
- C.I. \*
- Dirección

En este formulario, es necesario completar todos los campos correspondientes para registrar correctamente a un nuevo paciente en el sistema. Los campos obligatorios incluyen información personal, como nombre, apellido, teléfono, correo electrónico, contraseña, cédula o número de identificación, fecha de nacimiento, datos de contacto. El sistema no permitirá el registro si alguno de los campos requeridos queda vacío, mostrando una alerta de validación que indicará al usuario qué información falta. Es importante asegurarse de que todos los datos ingresados sean precisos.

### 6.5.2. Listado de Pacientes

The screenshot shows the 'Listado de Pacientes' in the COV system. The table displays the following data:

Nombre y Apellidos	Correo Electrónico	C.I.	Teléfono
Alex Andrés Corderos Albán	alexalban@gmail.com	0203472648	0845722515
Elm Estefanía Coloma Vázquez	elm23@gmail.com	0201672636	0972566117
Julia César Arías Meléndez	jc@gmail.com	5885232544	0843671345

Mostrando 3 de 3

Anterior [1](#) Siguiente

El listado de pacientes registrados muestra todos los pacientes que han sido añadidos al sistema tras completar correctamente su registro. En esta sección, se despliega información esencial de cada paciente. Además, en esta interfaz se encuentra el acceso a la historia clínica de cada paciente.

### 6.5.3. Editar Paciente

The screenshot shows the 'Editar Paciente' form with the following data:

Nombre *	Apellido *	Teléfono
Kim Estefanía	Coloma Vázquez	097250017

Correo Electrónico *	Contraseña *
kim125@gmail.com	

Confirmar Contraseña *	Fecha de Nacimiento
	11/8/2024

Sexo	Nivel de Educación	C.I. *
<input type="radio"/> Masculino <input type="radio"/> Femenino	Secundaria	0201672636

Dirección: Av. Magisterio y Ángel Palavicino Chávez

Permite al usuario actualizar la información de un paciente de forma rápida y organizada. Aquí se muestra un formulario prellenado con campos como nombre completo, número de identificación, fecha de nacimiento, datos de contacto y antecedentes médicos relevantes, mismos que pueden ser editados sin ningún inconveniente.

### 6.5.4. Eliminar Paciente

The screenshot shows the 'Lista de Pacientes' view with the following data:

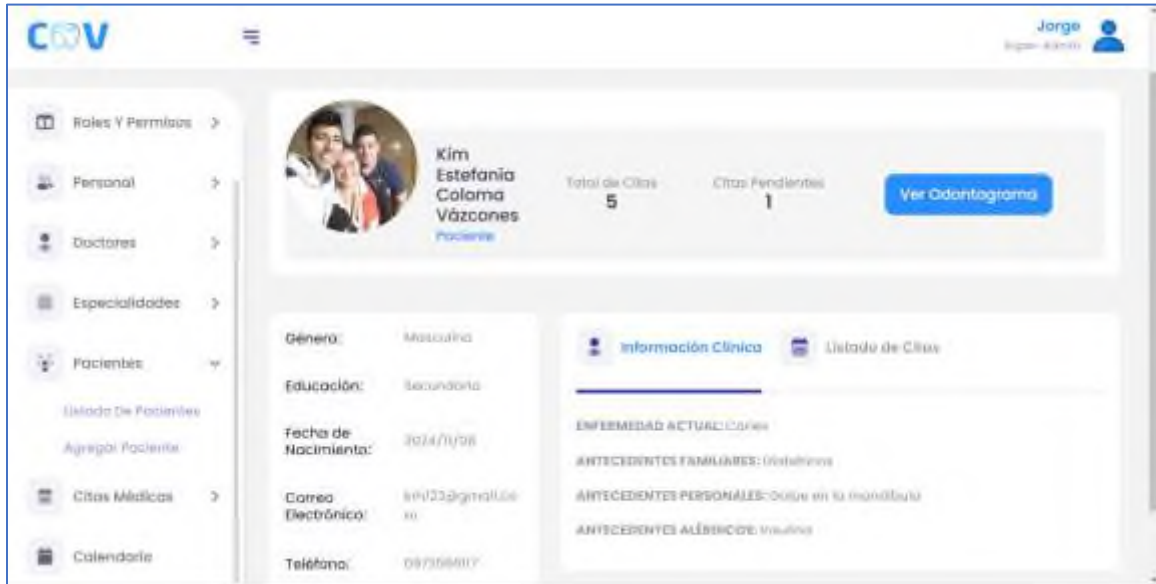
Nombre y Apellido	Correo Electrónico	C.I.	Teléfono
<b>Estefanía Coloma Vázquez</b>	alexalbon@gmail.com	0203472648	094672515
Kim Estefanía Coloma Vázquez	kim125@gmail.com	0201672636	097250017
Julia César Ariza Meléndez	jcg@gmail.com	5688232544	0943671245

Mostrando 3 de 3

Anterior 1 Siguiente

Aquí se puede eliminar de manera segura el registro de un paciente. Mostrará un mensaje de confirmación con detalles clave como el nombre completo del paciente.

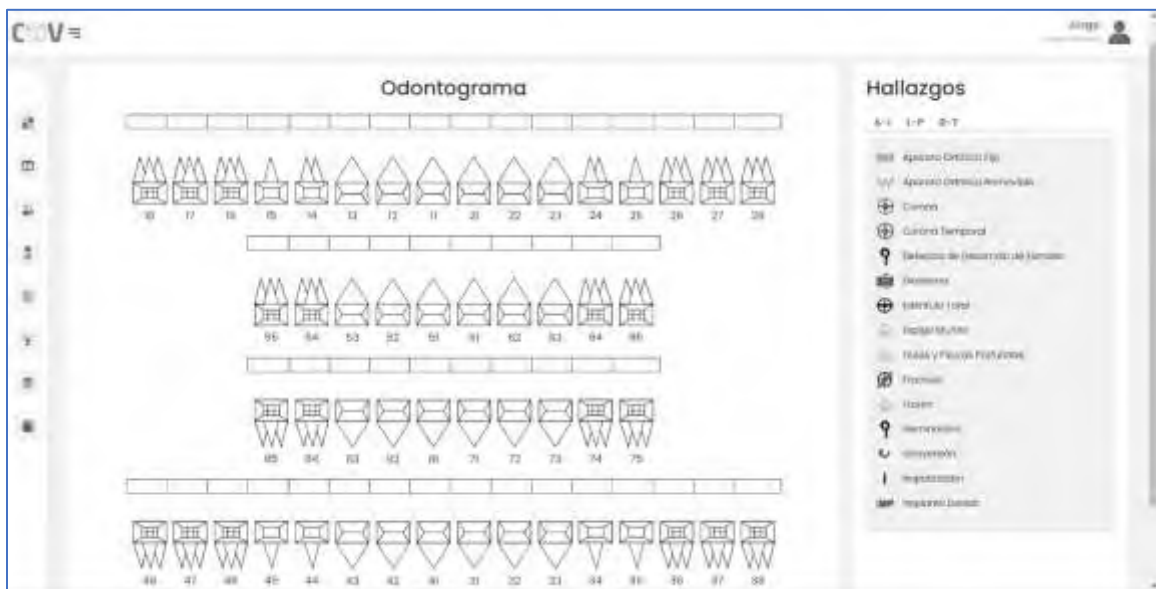
### 6.5.5. Historia Clínica



La **Historia Clínica** recopila y almacena toda la información médica relevante de un paciente, incluyendo diagnósticos, tratamientos, antecedentes médicos, notas de consulta y el odontograma.

## 6.6. Odontograma

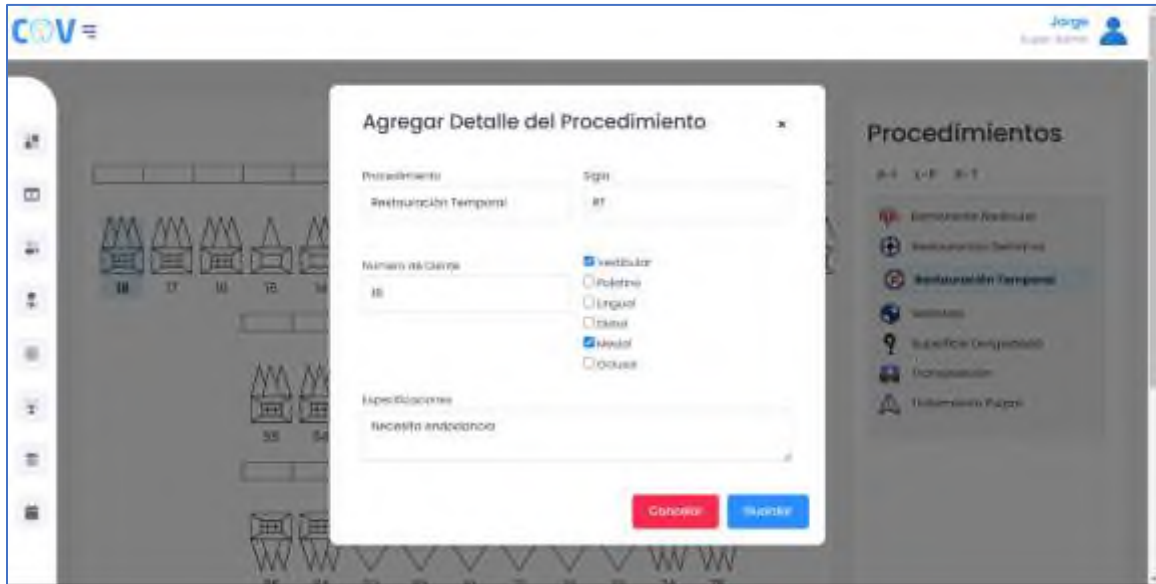
### 6.6.1. Visualizar Odontograma



Esta interfaz principal muestra el gráfico geométrico del odontograma de un paciente, de la mano de los procedimientos que pueden ser aplicados a uno o varios dientes del

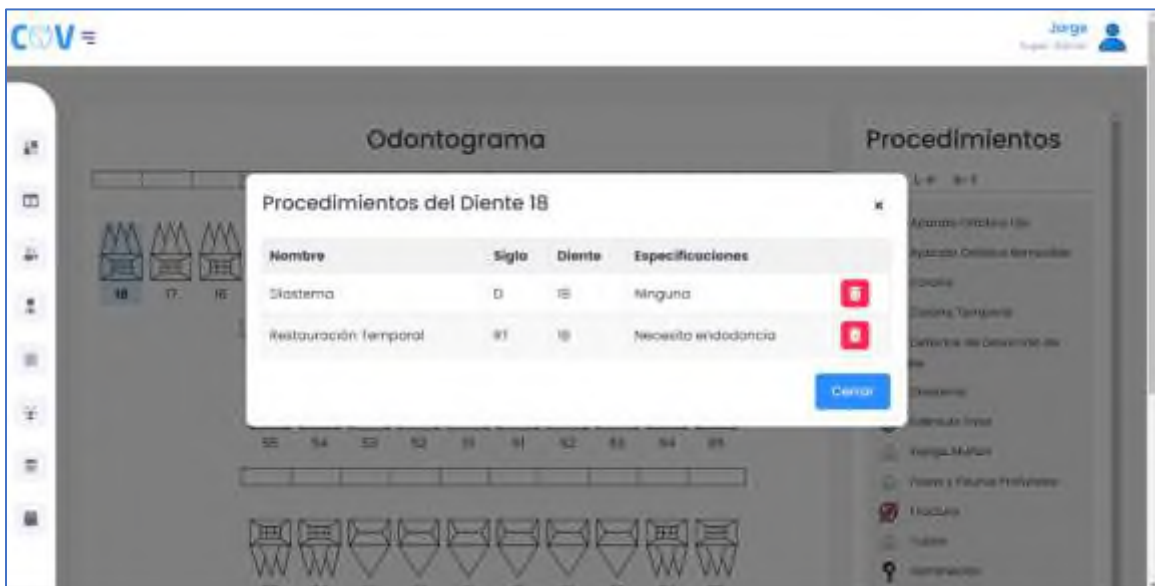
odontograma. Estos procedimientos se encuentran divididos en 3 distintas columnas que se pueden acceder mediante el panel de selección.

### 6.6.2. Agregar un procedimiento



Este modal permite registrar un procedimiento médico realizado a un paciente de manera sencilla. Incluirá un formulario con campos para ingresar información clave dependiendo el tipo de procedimiento a realizar.

### 6.6.3. Listado de procedimientos



Este modal muestra un resumen de todos los procedimientos registrados para un paciente o en general, organizados de manera clara y accesible. Cada procedimiento estará listado con información relevante correspondiente al mismo.

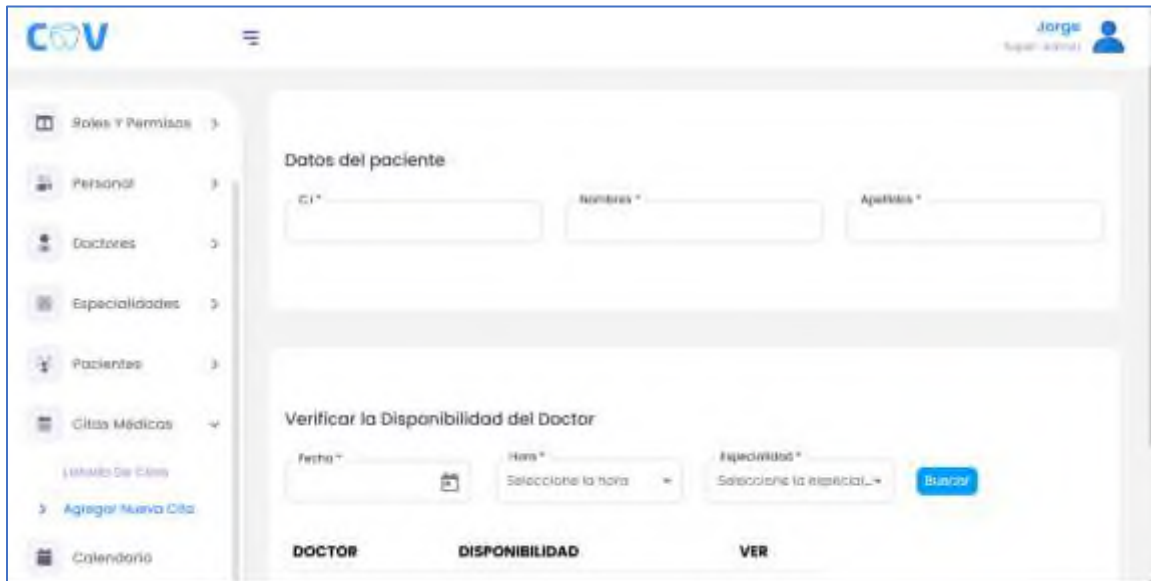
#### 6.6.4. Eliminar procedimiento



Permite eliminar un procedimiento registrado de forma segura, incluyendo un mensaje de confirmación que destaque los detalles clave del procedimiento a ser eliminado.

### 6.7. Citas Médicas

#### 6.7.1. Agregar Citas



La opción para agregar una cita permite programar una nueva cita en el sistema, requiriendo la entrada de datos esenciales como la fecha, la hora, la especialidad médica, el doctor asignado y la disponibilidad del doctor en el horario seleccionado. Es necesario completar todos estos campos para que la cita sea registrada correctamente y garantizar que todos los detalles estén correctamente coordinados.

### 6.7.2. Listado de Citas

Doctor	Paciente	Fecha	Hora	Especialidad	Estado
José Luis Maldonado Ortiz	Kim Estefanía Coloma Vázquez	2024-11-26	10:00 AM - 10:30 AM	Odonatología General	ATENDIDO
José Luis Maldonado Ortiz	Kim Estefanía Coloma Vázquez	2024-11-26	12:00 PM - 12:30 PM	Odonatología General	PENDIENTE
Carlos Luis Velastegui Vaca	Kim Estefanía Coloma Vázquez	2024-11-21	08:00 AM - 08:30 AM	Odonatología General	PENDIENTE
Carlos Luis Velastegui Vaca	Alex Andrés Cordero Albán	2024-11-11	09:00 AM - 09:30 AM	Odonatología General	PENDIENTE

El listado de citas muestra todas las citas programadas en el sistema, proporcionando información detallada sobre cada una. En esta sección, se visualizan el nombre del doctor, el paciente, la fecha y hora de la cita, la especialidad médica, el estado de la cita (pendiente o atendido) y la fecha en que la cita fue registrada. Este listado permite a los usuarios gestionar y revisar las citas de manera eficiente.

### 6.7.3. Reagendar Citas

Editar Cita : #98

Fecha \* 11/20/2024 Hora \* 12:00PM Especialidad \* Odontología General

**DOCTOR** **HORARIO REGISTRADO**  
 José Luis Maldonado Ortiz 12:00 PM a 12:30 PM

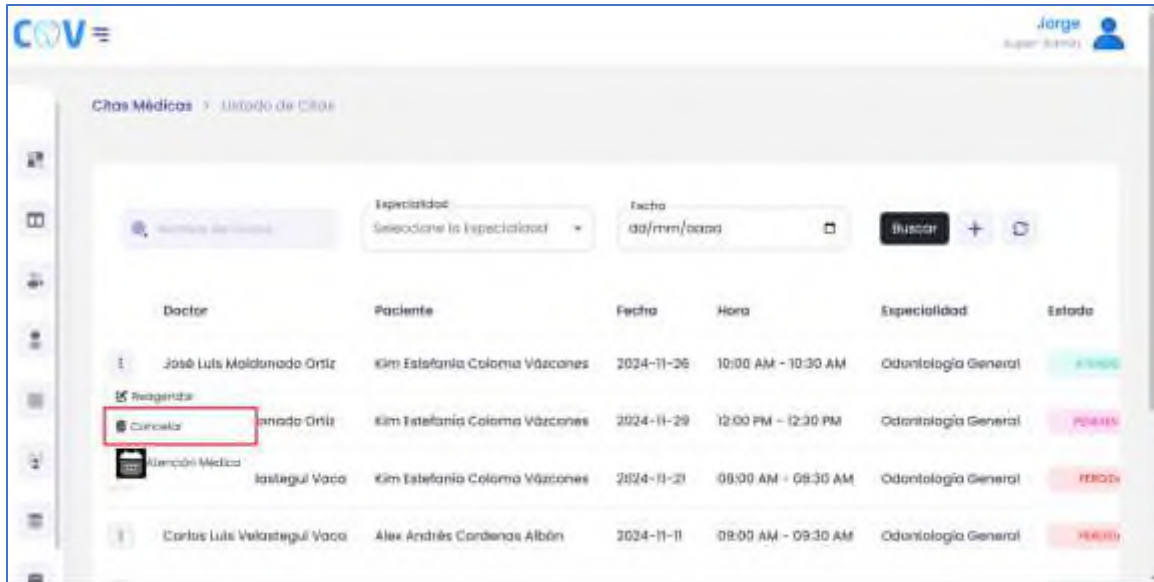
**DOCTOR** **DISPONIBILIDAD** **VER**

Datos del paciente:  
 N° Documento \* 0201672636 Nombre \* Kim Estefanía Apellido \* Coloma Vázquez

Editar

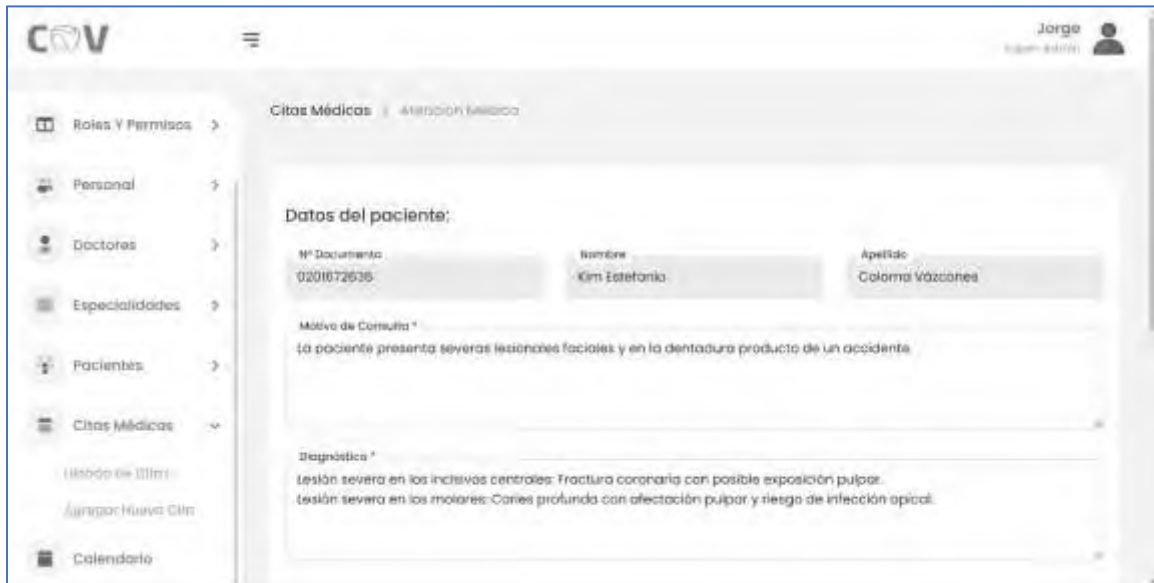
La opción reagendar una cita permite modificar la fecha y hora de una cita previamente programada, el usuario solo debe seleccionar la nueva fecha y hora, sin necesidad de volver a ingresar toda la información de la cita.

#### 6.7.4. Cancelar Citas



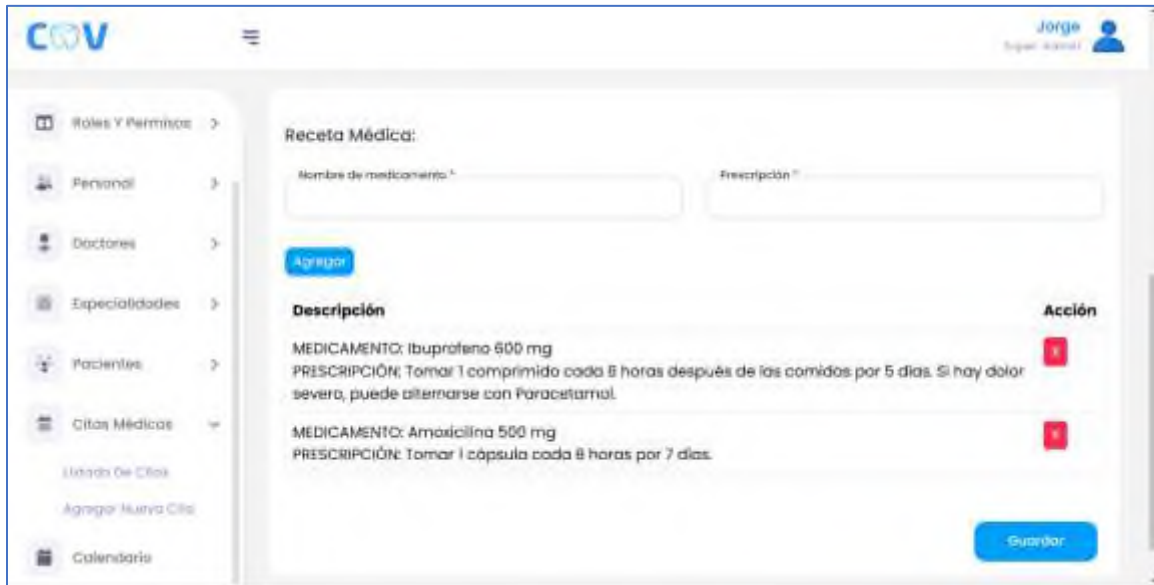
La opción para cancelar una cita permite eliminar una cita programada en el sistema de manera sencilla. Para cancelar una cita, el usuario solo necesita seleccionar la opción de cancelar, y el registro de su cita se eliminará.

#### 6.7.5. Atención de las Citas



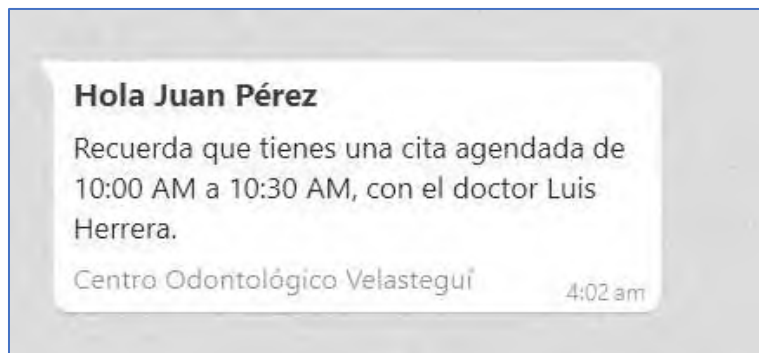
Aquí se puede realizar el debido registro de atención médica al paciente, donde se genera un motivo de consulta, diagnóstico y receta médica.

### 6.7.6. Receta medica



Permite a los usuarios generar y visualizar recetas médicas de manera estructurada. Incluye campos para ingresar los detalles del paciente, la fecha, el diagnóstico y una lista de medicamentos prescritos, con espacio para las dosis, la frecuencia y las instrucciones específicas para cada uno.

### 6.7.7. Notificación de Recordatorio



Esta funcionalidad permite al usuario enviar alertas al paciente sobre eventos importantes, en este caso son las citas médicas, con la respectiva información pertinente para que el paciente tenga conocimiento de manera previa y pueda asistir a la hora y día indicado.

---

**ANEXO N°8 - MANUAL TÉCNICO:  
SISTEMA DE GESTIÓN DE CITAS E  
HISTORIAS CLÍNICAS DEL  
CENTRO ODONTOLÓGICO  
VELASTEGUÍ**

**“SIWCH-OV” V 1.0**

---



**CENTRO ODONTOLÓGICO VELASTEGUI**

*Elaborado Por: Jorge Alexander Cárdenas Coloma*

*Luis Ángel Herrera Cedeño*

# CONTENIDO

1. Capa de Datos: MySQL.....	134
1.1 Sistema de Base de Datos.....	134
1.2 Estructura de la Base de Datos.....	134
1.3 Optimización.....	134
2. Capa de Lógica de Negocios: Laravel.....	135
2.1 Visión General del Framework.....	135
2.2 Componentes Clave.....	135
2.3 Seguridad.....	135
3. Capa de Presentación: Angular.....	135
3.1 Visión General del Framework.....	135
3.2 Componentes.....	136
4. Integración y Comunicación API.....	136
4.1 Estructura de API REST.....	136
4.2 Seguridad y Roles.....	136
5. Gestión de Usuarios.....	136
5.1 Gestión de Citas.....	137
6. Conclusión.....	137

## **Descripción General**

El sistema de gestión de citas e historias clínicas del "Centro Odontológico Velasteguí" está estructurado con una arquitectura de tres capas: Capa de Datos, Capa Lógica de Negocio y Capa de Presentación.

Este sistema gestiona diversas funcionalidades como citas, horarios de doctores, registros de pacientes e historias clínicas, además del manejo digital del odontograma de cada paciente. La arquitectura del sistema garantiza escalabilidad, integridad de datos y una interfaz de usuario fluida e intuitiva para administradores, doctores, secretarías y pacientes.

### **1. Capa de Datos: MySQL**

#### **1.1 Sistema de Base de Datos**

La capa de datos utiliza MySQL como sistema de gestión de bases de datos. Se almacena toda la información crítica del sistema, incluidos los detalles de los pacientes, horarios de doctores, citas e historias clínicas y demás información pertinente.

#### **1.2 Estructura de la Base de Datos**

- **Tablas:** El sistema incluye varias tablas clave para cada entidad.
- **Pacientes:** Almacena la información de contacto y los identificadores únicos de los pacientes, así como su información médica, entre otros.
- **Doctores:** Almacena los perfiles de los doctores, especialidades y disponibilidad.
- **Citas:** Gestiona los registros de citas programadas, cancelaciones y reprogramaciones.
- **Historias:** Contiene los registros de historias clínicas de los pacientes incluido el odontograma.
- **Relaciones:** La base de datos utiliza llaves primarias y foráneas para enlazar las tablas, asegurando la integridad referencial.

#### **1.3 Optimización**

El rendimiento de la base de datos se mejora mediante:

- Indexación de campos frecuentemente consultados como Id de pacientes, fechas de citas, etc.

- Procedimientos almacenados para manejar consultas complejas de manera eficiente.

## **2. Capa de Lógica de Negocios: Laravel**

### **2.1 Visión General del Framework**

El núcleo de la lógica de negocio está construido utilizando Laravel, un framework PHP conocido por su patrón de desarrollo arquitectónico MVC y su soporte nativo para APIs. La API del backend maneja las solicitudes y asegura una comunicación fluida entre el frontend (Angular) y la base de datos MySQL.

### **2.2 Componentes Clave**

**Rutas y Controladores:** Laravel utiliza `web.php` y `api.php` para el enrutamiento. Estos archivos definen cómo se asignan las diferentes URLs a funciones específicas en el sistema, ya sea sirviendo la interfaz web o procesando llamadas API.

**Modelos Eloquent:** El ORM Eloquent de Laravel permite la manipulación orientada a objetos de los registros de la base de datos. Las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) se gestionan fácilmente mediante Eloquent, que abstrae los comandos SQL.

**Servicios:** Servicios específicos manejan:

- Horarios de doctores.
- Asignación de citas.
- Gestión de historias clínicas, desacoplando la lógica compleja de los controladores.

### **2.3 Seguridad**

**Autenticación:** La API implementa autenticación mediante JWT (JSON Web Token) para garantizar un acceso seguro basado en roles de usuario, como administrador, doctor, secretaria o paciente. Los tokens JWT validan la identidad y controlan los privilegios.

## **3. Capa de Presentación: Angular**

### **3.1 Visión General del Framework**

El frontend está construido usando Angular CLI, un framework popular para desarrollar aplicaciones web dinámicas. Permite actualizaciones en tiempo real y una experiencia de usuario fluida para diferentes roles de usuario.

### 3.2 Componentes

**Interfaz Responsiva:** La interfaz del sistema está optimizada para diferentes dispositivos, asegurando su usabilidad tanto en computadoras de escritorio como en móviles, con interfaces específicas para administradores, doctores y secretarias.

## 4. Integración y Comunicación API

### 4.1 Estructura de API REST

El sistema expone una API RESTful, lo que permite una fácil comunicación entre el frontend y el backend. Esta API maneja todas las operaciones principales, incluyendo operaciones CRUD para citas, pacientes, doctores entre otros.

### 4.2 Seguridad y Roles

La API está asegurada utilizando tokens JWT para restringir el acceso en función del rol del usuario. Los administradores pueden ver y modificar todos los registros, mientras que los doctores y secretarias tienen privilegios limitados dependiendo de sus tareas.

## 5. Gestión de Usuarios

**Administrador:** Tiene control total sobre el sistema, incluyendo la gestión de usuarios (doctores, secretarias), la asignación de roles, la supervisión y manejo de citas e historias clínicas de los pacientes.

**Doctor:** Accede a la lista de pacientes, horarios de citas e historias clínicas. Puede actualizar historias clínicas y aprobar/cancelar citas. Además, el doctor será el principal encargado de manejar los procedimientos dentales en el odontograma de cada paciente, interactuando con la información pertinente de cada diente y su evolución.

**Secretaria:** Es responsable de programar, cancelar y reprogramar citas. También gestiona los horarios de los doctores y el registro de pacientes.

**Paciente:** Puede programar, cancelar y reprogramar citas. Además, puede visualizar las especialidades, doctores y días de trabajo de los doctores disponibles.

## **5.1 Gestión de Citas**

**Crear/Actualizar/Eliminar Citas:** Los administradores, doctores y secretarias tienen permisos para gestionar las citas de los pacientes. Los pacientes solo pueden interactuar con sus propias citas.

**Notificaciones:** El sistema envía notificaciones a los pacientes para citas próximas o cambios en las citas agendadas por motivos varios acordes a la disposición administrativa del Centro Odontológico.

## **6. Conclusión**

Este manual describe la estructura técnica y funcionalidad del sistema de gestión de citas e historias clínicas del "Centro Odontológico Velasteguí". El sistema aprovecha Laravel para la lógica de negocio, MySQL para el almacenamiento de datos y Angular para la capa de presentación, proporcionando una solución robusta, escalable y segura para gestionar citas clínicas e historiales médicos de los pacientes, incluyendo la capacidad de manejar sus odontogramas de manera digital y efectiva para el correcto seguimiento a los tratamientos dentales de los pacientes, y de una manera u otra optimizando su atención clínica.

# **ANEXO N°9**

---

**CERTIFICADO DE CONFORMIDAD POR  
PARTE DEL CENTRO ODONTOLÓGICO  
VELASTEGUÍ**



CENTRO ODONTOLÓGICO VELASTEGUI  
DR. CARLOS VELASTEGUI

Echeandía, 06 de mayo del 2024

Doctor  
Carlos Velastegui  
Propietario

A PETICIÓN DE LOS INTERESADOS, CERTIFICO:

De acuerdo con lo solicitado por los interesados, certifico que los señores Jorge Alexander Cárdenas Coloma y Luis Ángel Herrera Cedeño, egresados de la UNIVERSIDAD ESTATAL DE BOLIVAR, de la carrera de SOFTWARE, realizaron la entrega y capacitación del SISTEMA WEB BASADO EN UNA METODOLOGÍA ÁGIL PARA GESTIONAR CITAS E HISTORIAS CLÍNICAS EN EL CENTRO ODONTOLÓGICO VELASTEGUÍ, CANTÓN ECHEANDÍA, correspondiente al proyecto de integración curricular, en el cual mi persona como propietario del centro odontológico beneficiado estoy de acuerdo con lo presentado por los responsables.

Es todo cuanto puedo mencionar en honor a la verdad, pudiendo los interesados hacer uso del presente como a bien lo tuviera.

Agradezco la atención al presente.

Saludos Cordiales

Atentamente,

Dr. Carlos Velastegui V.  
ODONTOLOGO  
REGISTRO SENESCYI  
1945-2017-1600260

Dr. Carlos Velastegui  
C. I: 0202244539  
Propietario del Centro Odontológico Velastegui

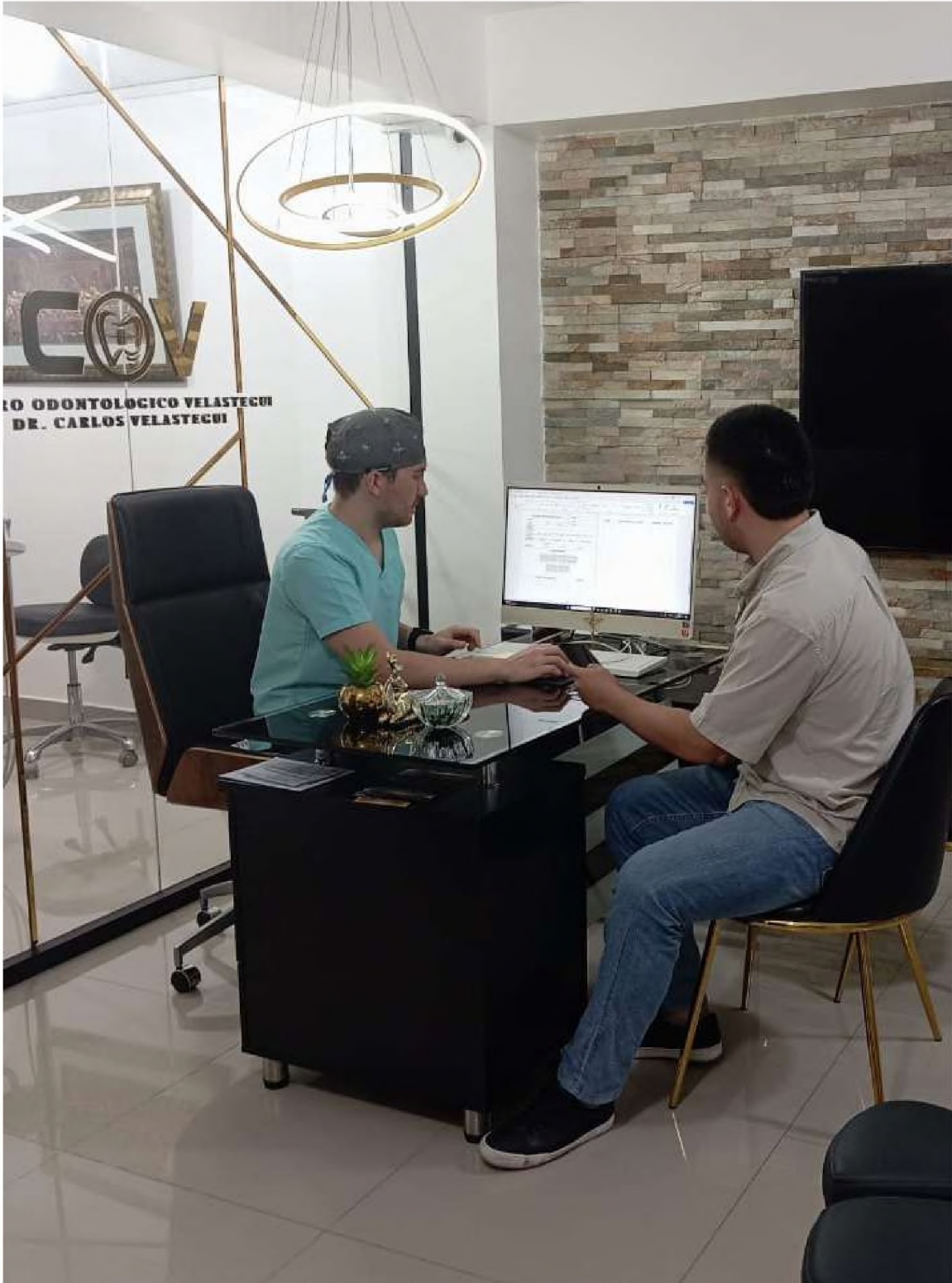
# **ANEXO N°10**

---

**EVIDENCIA DE LAS ENCUESTAS,  
ENTREVISTAS Y CAPACITACIÓN CON LOS  
MIEMBROS DEL CENTRO ODONTOLÓGICO  
VELASTEGUÍ**







# **ANEXO N°11**

---

**CERTIFICADO ANTIPLAGIO**

**ING. GALUTH IRENE GARCÍA CAMACHO EN CALIDAD DE  
DIRECTORA DEL TRABAJO DE INTEGRACIÓN CURRICULAR,**

### **CERTIFICA**

Que el trabajo de integración curricular denominado **"SISTEMA WEB BASADO EN UNA METODOLOGÍA ÁGIL PARA GESTIONAR CITAS E HISTORIAS CLÍNICAS EN EL CENTRO ODONTOLÓGICO VELASTEGUÍ, CANTÓN ECHEANDÍA"**, presentado por Luis Ángel Herrera Cedeño y Jorge Alexander Cárdenas Coloma estudiantes de la carrera de **Software** pasó el análisis de coincidencia no accidental en la herramienta TURNITIN, reflejando un porcentaje de similitud del 10%, como se puede evidenciar en el documento adjunto.

Guaranda, 05 de diciembre del 2024

Atentamente,  
  
Ing. Galuth García C.  
Directora

**Jorge Cárdenas Luis Herrera**

# **SISTEMA WEB BASADO EN UNA METODOLOGÍA ÁGIL PARA GESTIONAR CITAS E HISTORIAS CLÍNICAS.pdf**

 My Files

 My Files

 Universidad Estatal de Bolívar

## **Detalles del documento**

Identificador de la entrega

turnitin:3117412878610

**121 Páginas**

Fecha de entrega

5 dic 2024, 7:35 a.m. GMT-5

**19,388 Palabras**

Fecha de descarga

5 dic 2024, 8:12 p.m. GMT-5

**123,177 Caracteres**

Nombre de archivo

TESIS FINAL CARDENAS HERRERA.pdf

Tamaño de archivo

**2.4 MB**

## 10% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes sujeta a puntuación, para el documento.

### Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado
- Coincidencias menores (menos de 12 palabras)
- Trabajos entregados
- Base de datos de contenido publicado de Crossref

### Exclusiones

- N.º de fuentes excluidas
- N.º de coincidencias excluidas

### Fuentes principales

- 10% Fuentes de Internet
- 2% Publicaciones
- 0% Trabajos entregados (trabajos del estudiante)

### Marcas de integridad

N.º de alertas de integridad para revisión

- Texto oculto**  
11 caracteres sospechosos en N.º de página  
El texto es añadido para mezclarse con el fondo blanco del documento.

Las alertas de nuestro sistema se añaden los documentos en profundidad para buscar inconsistencias que permitan distinguir de una entrega normal. Si detectamos algo extraño, le hacemos una marca alerta para que pueda revisar.

Una marca de alerta de inconsistencia es un indicador de precaución. En un momento, recomendamos que revise el contenido que puede ser sospechoso y lo revise.