



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Tema:

**APLICACIÓN WEB EMPLEANDO TECNOLOGÍA RESPONSIVE WEB
DESIGN PARA EL AGENDAMIENTO DE CITAS EN EL CENTRO
ODONTOLÓGICO “DENTALSIS”.**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniera en Tecnologías de la
Información

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de software

AUTOR: Valeria Michelle Logroño Padilla

TUTOR: Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

Ambato - Ecuador

agosto – 2023

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: APLICACIÓN WEB EMPLEANDO TECNOLOGÍA RESPONSIVE WEB DESIGN PARA EL AGENDAMIENTO DE CITAS EN EL CENTRO ODONTOLÓGICO “DENTALSIS”, desarrollado bajo la modalidad Proyecto de Investigación por la señorita Valeria Michelle Logroño Padilla, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorada durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, agosto 2023.

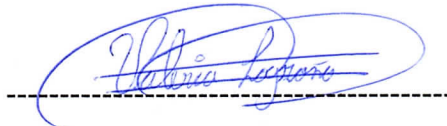
Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación titulado: APLICACIÓN WEB EMPLEANDO TECNOLOGÍA RESPONSIVE WEB DESIGN PARA EL AGENDAMIENTO DE CITAS EN EL CENTRO ODONTOLÓGICO “DENTALSIS” es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2023.



Valeria Michelle Logroño Padilla

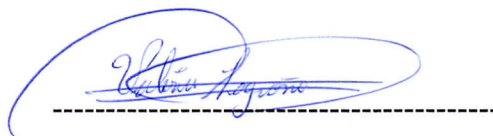
C.C. 180375057-7

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, agosto 2023.

A handwritten signature in blue ink, appearing to read 'Valeria Logroño Padilla', is written over a horizontal dashed line.

Valeria Michelle Logroño Padilla

C.C. 180375057-7

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por la señorita Valeria Michelle Logroño Padilla, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **APLICACIÓN WEB EMPLEANDO TECNOLOGÍA RESPONSIVE WEB DESIGN PARA EL AGENDAMIENTO DE CITAS EN EL CENTRO ODONTOLÓGICO “DENTALSIS”**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, agosto 2023.

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Hernán Fabricio Naranjo Ávalos, Mg.
PROFESOR CALIFICADOR

Ing. Oscar Fernando Ibarra Torres, Mg.
PROFESOR CALIFICADOR

DEDICATORIA

A mis padres Edmundo y Mirian quienes con infinita paciencia, amor y dedicación me han permitido cumplir uno de mis grandes sueños, gracias por impartir en mí el ejemplo de esfuerzo, valentía y coraje, de no temer a las dificultades debido a la constante compañía de Dios en mi vida.

A mis angelitos, mis abuelitos Lucinda y Sergio que desde pequeña me enseñaron a ser una mujer de bien, perseverante y luchadora. A Jesús que siempre estuvo conmigo en las buenas y malas enseñándome a afrontar las dificultades de la vida con una sonrisa, sé que me cuidan y guían desde el cielo.

A mis hermanas Leonor y Jessica por su amor y cuidado, ya que a través de su afecto me han enseñado valiosas lecciones sobre cómo enfrentar desafíos y seguir avanzando.

Finalmente, a toda mi familia y amigos quienes siempre me han ofrecido consejos, respaldo incondicional y afecto a lo largo de este recorrido, muchas gracias.

Valeria Michelle Logroño Padilla

AGRADECIMIENTO

Deseo expresar mi más sincero agradecimiento, en primer lugar, a Dios por orientarme en mi trayecto y permitirme culminar una nueva etapa de formación académica.

A mis padres y familia por su confianza inquebrantable en mí, respaldando cada elección y cada paso que he dado en mi vida.

A mi tutor, Ing. Franklin Mayorga por asesorarme en el desarrollo del presente proyecto, compartir su amplio conocimiento y por ser un modelo ejemplar en el ámbito profesional.

A la odontóloga, Dra. Pamela Toapanta propietaria del Centro Odontológico “DENTALSIS” por haberme proporcionado todas las facilidades necesarias durante la realización de la investigación.

A mi mejor amigo Jonathan por su apoyo y amistad en todos estos años de estudio y de esfuerzo.

Por último, deseo expresar gratitud hacia mi persona, por dedicar tanto esfuerzo en este arduo trabajo, por la capacidad para superar obstáculos con valentía y empeñarme para ser la mejor versión de mí misma.

Valeria Michelle Logroño Padilla

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR.....	iv
APROBACIÓN DEL TRIBUNAL DE GRADO.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO.....	vii
ÍNDICE GENERAL.....	viii
ÍNDICE DE GRÁFICOS.....	xi
ÍNDICE DE TABLAS.....	xv
RESUMEN EJECUTIVO.....	xviii
ABSTRACT.....	xix
Capítulo I.- MARCO TEÓRICO.....	1
1.1. Tema de investigación.....	1
1.1.1. Planteamiento del Problema.....	1
1.2. Antecedentes Investigativos.....	2
1.3. Fundamentación Teórica.....	5
1.4. Objetivos.....	25
1.4.1. Objetivo General.....	25
1.4.2. Objetivos Específicos.....	25
Capítulo II.- METODOLOGÍA.....	26
2.1. Materiales.....	26
2.2. Métodos.....	30
2.2.1. Modalidad de la investigación.....	30
2.2.2. Población y Muestra.....	30
2.2.3. Recolección de la información.....	31

2.2.4.	Procesamiento y análisis de datos	49
Capítulo III.- RESULTADOS Y DISCUSIÓN		50
3.1.	Análisis y discusión de los resultados	50
3.1.1.	Proceso actual del agendamiento de citas	50
3.1.2.	Proceso de Agendamiento de citas optimizado	51
3.1.3.	Análisis de la Tecnología Responsive Web Design (RWD).....	53
3.1.4.	Tecnologías Front-end.....	55
3.1.5.	Frameworks de Diseño.....	56
3.1.6.	Frameworks Backend.....	58
3.1.7.	Gestores de base de datos.....	61
3.1.8.	Metodologías de Desarrollo Web.....	62
3.2.	Desarrollo de la propuesta	65
3.2.1.	Fase I: Planificación.....	65
3.2.1.1.	Levantamiento de la información.....	65
3.2.1.2.	Requisitos Funcionales del Sistema	65
3.2.1.3.	Roles Asignados en el proyecto.....	66
3.2.1.4.	Arquitectura de la aplicación.....	67
3.2.1.5.	Requerimientos de Software.....	69
3.2.1.6.	Historias de Usuario	69
3.2.1.7.	Valoración / Estimación de Historias de Usuario	76
3.2.1.8.	Plan de entrega	76
3.2.2.	Fase II: Diseño	77
3.2.2.1.	Iteración 1	77
3.2.2.2.	Iteración 2.....	78
3.2.2.3.	Iteración 3	80
3.2.2.4.	Iteración 4.....	81
3.2.2.5.	Tarjetas CRC	82

3.2.2.6.	Diseño de modelo de datos.....	86
3.2.2.7.	Diseño de Iteraciones	87
3.2.3.	Fase III: Codificación.....	96
3.2.3.1.	Creación de la Solución en Visual Studio 2019	96
3.2.3.2.	Conexión de base de datos	96
3.2.3.3.	Desarrollo Back - End	97
3.2.3.4.	Desarrollo Front-end	121
3.2.4.	Fase IV: Pruebas.....	124
3.2.4.1.	Pruebas de aceptación.....	124
3.2.4.2.	Pruebas de usabilidad	129
3.2.5.	Fase V: Implantación.....	131
Capítulo IV.- CONCLUSIONES Y RECOMENDACIONES		137
4.1.	Conclusiones.....	137
4.2.	Recomendaciones	138
BIBLIOGRAFÍA.....		139
ANEXOS.....		145

ÍNDICE DE GRÁFICOS

Gráfico 2.1. Tiempo que acuden los pacientes al centro odontológico.....	33
Gráfico 2.2. Frecuencia de los pacientes a realizarse una consulta.....	34
Gráfico 2.3. Nivel de satisfacción referente a la atención.....	35
Gráfico 2.4. ¿Cree usted que el consultorio debe seguir llevando el control de la documentación de forma física?.....	36
Gráfico 2.5. Dificultad para reservar una cita	37
Gráfico 2.6. ¿Cree usted que se está realizando de forma ordenada y eficaz las reservas de citas?.....	38
Gráfico 2.7. Formas que se realiza la reserva de una cita en el consultorio	39
Gráfico 2.8. ¿Cree usted qué es necesario un sistema informático para registrar la información dentro del consultorio para evitar perdida de datos?	40
Gráfico 2.9. Toma del tiempo se considera excesivo	41
Gráfico 2.10. Tiempo de espera para ser atendido en la cita.....	42
Gráfico 2.11. ¿Considera usted que se podría agilizar el proceso de asignar turnos del centro odontológico?	43
Gráfico 2.12. ¿Considera positivo que un sistema informático mejore el proceso de agendamiento de citas?	44
Gráfico 3.1. Proceso actual del agendamiento de citas	50
Gráfico 3.2. Proceso de Agendamiento de citas optimizado.....	52
Gráfico 3.3. Arquitectura del Sistema	68
Gráfico 3.4. Modelo Entidad – Relación de la base de datos.....	87
Gráfico 3.5. Inicio de Sesión.....	88
Gráfico 3.6. Recuperar Contraseña	88
Gráfico 3.7. Página Principal	89
Gráfico 3.8. Registro y listado de Pacientes.....	90
Gráfico 3.9. Modal de Actualizar Paciente	90
Gráfico 3.10. Gestión de horarios de odontólogos.....	91
Gráfico 3.11. Modal de Agregar Horario Atención.....	91
Gráfico 3.12. Modal de Editar Horario Atención.....	92
Gráfico 3.13. Gestión de reserva de citas.....	92
Gráfico 3.14. Gestión Atención Pacientes.....	93

Gráfico 3.15. Registrar Diagnostico.....	93
Gráfico 3.16. Registro y listado de Odontólogos	94
Gráfico 3.17. Modal de Actualizar Odontólogo	94
Gráfico 3.18. Interfaz del comprobante	95
Gráfico 3.19. Formato del comprobante	95
Gráfico 3.20. Finalizar Sesión.....	96
Gráfico 3.21. Capas de la solución.....	96
Gráfico 3.22. Conector MySQL para ASP.NET Web Applications	96
Gráfico 3.23. Clase Conexión	97
Gráfico 3.24. Entidades del proyecto	97
Gráfico 3.25. Entidad Cita.....	98
Gráfico 3.26. Entidad Diagnostico	98
Gráfico 3.27. Entidad Empleado	99
Gráfico 3.28. Entidad Especialidad.....	99
Gráfico 3.29. Entidad HistoriaClinica.....	100
Gráfico 3.30. Entidad Hora	100
Gráfico 3.31. Entidad HorarioAtencion	100
Gráfico 3.32. Entidad IngresoSistema.....	101
Gráfico 3.33. Entidad Odontólogo	101
Gráfico 3.34. Entidad Paciente.....	102
Gráfico 3.35. Entidad TipoEmpledo	103
Gráfico 3.36. Estructura del Patrón Singleton.....	103
Gráfico 3.37. Método registrar cita	104
Gráfico 3.38. Método listar citas	104
Gráfico 3.39. Método actualizar cita	105
Gráfico 3.40. Método registrar diagnostico	105
Gráfico 3.41. Método listar especialidades	106
Gráfico 3.42. Método registrar horario atención.....	106
Gráfico 3.43. Método actualizar horario de atención.....	107
Gráfico 3.44. Método eliminar horario de atención.	107
Gráfico 3.45. Método listar horario de atención por id del odontólogo.....	108
Gráfico 3.46. Método listar horario de atención por fecha	108
Gráfico 3.47. Método para acceder al sistema.	109

Gráfico 3.48. Método para obtener el nombre del odontólogo	109
Gráfico 3.49. Método para buscar odontólogo por cedula de identidad.	110
Gráfico 3.50. Método para registrar odontólogo.....	111
Gráfico 3.51. Método para listar odontólogos.....	111
Gráfico 3.52. Método para eliminar odontólogo.....	112
Gráfico 3.53. Método para actualizar odontólogo.....	112
Gráfico 3.54. Método para registrar paciente.....	113
Gráfico 3.55. Método para eliminar paciente por cedula de identidad	113
Gráfico 3.56. Método para listar pacientes activos	114
Gráfico 3.57. Método para actualizar paciente	114
Gráfico 3.58. Método para buscar paciente por cedula de identidad	115
Gráfico 3.59. Método para buscar paciente por ide de la cita.....	115
Gráfico 3.60. Atributos de la clase RecuperarPasswordDAO.....	116
Gráfico 3.61. Método para inicializar el cliente SMTP	116
Gráfico 3.62. Método de la estructura del correo.....	116
Gráfico 3.63. Método para recuperar la contraseña por correo.....	117
Gráfico 3.64. Envío de los atributos heredados de RecuperarPasswordDAO.	117
Gráfico 3.65. Clase de lógica de negocio de Cita	118
Gráfico 3.66. Clase de lógica de negocio de Diagnostico.....	118
Gráfico 3.67. Clase de lógica de negocio de Especialidad.....	118
Gráfico 3.68. Clase de lógica de negocio de HorarioAtencion.....	119
Gráfico 3.69. Clase de lógica de negocio de Ingreso	119
Gráfico 3.70. Clase de lógica de negocio de Odontólogo.....	120
Gráfico 3.71. Clase de lógica de negocio de Paciente	120
Gráfico 3.72. Clase de lógica de negocio de recoverPassword.....	121
Gráfico 3.73. Validación de solo ingreso de números.....	121
Gráfico 3.74. Validación de solo ingreso de letras mayúsculas	121
Gráfico 3.75. Archivo Javascript de HorarioOdontologo	122
Gráfico 3.76. Archivo Javascript de Paciente	122
Gráfico 3.77. Archivo Javascript de Reserva	123
Gráfico 3.78. Archivo Javascript de Odontólogo.....	123
Gráfico 3.79. Archivo Javascript de Logout	123
Gráfico 3.80. Características del servidor de Smarterasp.net	131

Gráfico 3.81. Creación de la cuenta en Smarterasp.net	132
Gráfico 3.82. Importación de la Base de Datos MySQL	132
Gráfico 3.83. Credenciales de la Base de Datos MySQL	132
Gráfico 3.84. Credenciales del dominio.....	133
Gráfico 3.85. Cadena de conexión	133
Gráfico 3.86. Configuraciones de la Conexión del Web Deploy	134
Gráfico 3.87. Publicación de la Aplicación web con servidor Smarterasp.net	134
Gráfico 3.88. Archivos y carpetas publicados en el servidor Smarterasp.net	135
Gráfico 3.89. Aplicación implementada en el sitio web	135

ÍNDICE DE TABLAS

Tabla 2.1. Guía de entrevista aplicada al odontólogo principal del consultorio.	27
Tabla 2.2. Población.....	30
Tabla 2.3. Varianza de los resultados obtenidos en la encuesta aplicada	31
Tabla 2.4. Tabulación de resultados de la pregunta 1	32
Tabla 2.5. Tabulación de resultados de la pregunta 2.....	33
Tabla 2.6. Tabulación de resultados de la pregunta 3.....	34
Tabla 2.7. Tabulación de resultados de la pregunta 4.....	35
Tabla 2.8. Tabulación de resultados de la pregunta 5.....	36
Tabla 2.9. Tabulación de resultados de la pregunta 6.....	37
Tabla 2.10. Tabulación de resultados de la pregunta 7.....	38
Tabla 2.11. Tabulación de resultados de la pregunta 8.....	39
Tabla 2.12. Tabulación de resultados de la pregunta 9.....	40
Tabla 2.13. Tabulación de resultados de la pregunta 10.....	41
Tabla 2.14. Tabulación de resultados de la pregunta 11	42
Tabla 2.15. Tabulación de resultados de la pregunta 12.....	43
Tabla 2.16. Resultados de la entrevista	48
Tabla 3.1. Tabla comparativa entre Responsive Web Design (RWD) y Adaptive Web Design (AWD).....	54
Tabla 3.2. Tabla comparativa entre las tecnologías utilizadas en el desarrollo de aplicaciones RWD.....	56
Tabla 3.3. Tabla comparativa entre los frameworks utilizados en el desarrollo de aplicaciones RWD.....	58
Tabla 3.4. Tabla comparativa entre Frameworks Backend para el desarrollo de aplicaciones Responsive Web Design (RWD)	60
Tabla 3.5. Tabla comparativa entre los gestores de bases de datos	61
Tabla 3.6. Tabla comparativa entre las Metodologías de Desarrollo Web	64
Tabla 3.7. Requisitos Funcionales del Sistema	65
Tabla 3.8. Definición de Roles XP	67
Tabla 3.9. Modelo de historia de usuario	69
Tabla 3.10. Historia de usuario para el ingreso al sistema	70
Tabla 3.11. Historia de usuario para la recuperación de contraseña.....	71

Tabla 3.12. Historia de usuario para la creación de la página principal.....	71
Tabla 3.13. Historia de usuario para la gestión pacientes	72
Tabla 3.14. Historia de usuario para la gestión de horarios de atención	72
Tabla 3.15. Historia de usuario para la reserva de citas	73
Tabla 3.16. Historia de usuario para la gestión de atención de pacientes	73
Tabla 3.17. Historia de usuario para la gestión de odontólogos.....	74
Tabla 3.18. Historia de usuario para la creación del Comprobante.....	74
Tabla 3.19. Historia de usuario para generar el pdf del comprobante.....	75
Tabla 3.20. Historia de usuario para finalizar sesión	75
Tabla 3.21. Estimación de historias de usuarios.....	76
Tabla 3.22. Plan de entrega	77
Tabla 3.23. Historias de usuario de la iteración 1	77
Tabla 3.24. Actividades de la historia de usuario 1	78
Tabla 3.25. Actividades de la historia de usuario 2	78
Tabla 3.26. Actividades de la historia de usuario 3	78
Tabla 3.27. Historias de usuario de la iteración 2	79
Tabla 3.28. Actividades de la historia de usuario 4	79
Tabla 3.29. Actividades de la historia de usuario 5	79
Tabla 3.30. Actividades de la historia de usuario 6	80
Tabla 3.31. Historias de usuario de la iteración 3	80
Tabla 3.32. Actividades de la historia de usuario 7	80
Tabla 3.33. Actividades de la historia de usuario 8	81
Tabla 3.34. Actividades de la historia de usuario 9	81
Tabla 3.35. Historias de usuario de la iteración 4	81
Tabla 3.36. Actividades de la historia de usuario 10	82
Tabla 3.37. Actividades de la historia de usuario 11	82
Tabla 3.38. Tarjeta CRC – Ingreso al sistema	83
Tabla 3.39. Tarjeta CRC – Recuperar Contraseña.....	83
Tabla 3.40. Tarjeta CRC – Página Principal.....	83
Tabla 3.41. Tarjeta CRC – Gestión Pacientes.....	84
Tabla 3.42. Tarjeta CRC – Gestión Horarios de Atención	84
Tabla 3.43. Tarjeta CRC – Reserva de Citas.	84
Tabla 3.44. Tarjeta CRC – Gestión Atención Pacientes	85

Tabla 3.45. Tarjeta CRC – Gestión Odontólogos	85
Tabla 3.46. Tarjeta CRC – Comprobante	85
Tabla 3.47. Tarjeta CRC – Generar Pdf del Comprobante	86
Tabla 3.48. Tarjeta CRC – Finalizar Sesión	86
Tabla 3.49. Prueba de Aceptación 1 - Ingreso al sistema	124
Tabla 3.50. Prueba de Aceptación 2 - Ingreso al sistema	124
Tabla 3.51. Prueba de Aceptación 3 - Recuperar contraseña	125
Tabla 3.52. Prueba de Aceptación 4 - Página Principal.....	125
Tabla 3.53. Prueba de Aceptación 5 - Gestión Pacientes	126
Tabla 3.54. Prueba de Aceptación 6 - Gestión Horarios de Atención	126
Tabla 3.55. Prueba de Aceptación 7 - Reserva de Citas	127
Tabla 3.56. Prueba de Aceptación 8 - Gestión Atención Pacientes.....	127
Tabla 3.57. Prueba de Aceptación 9 - Gestión Odontólogos.....	128
Tabla 3.58. Prueba de Aceptación 10 - Comprobante	128
Tabla 3.59. Prueba de Aceptación 11 - Generar Pdf del Comprobante.....	129
Tabla 3.60. Prueba de Aceptación 12 - Finalizar Sesión.....	129
Tabla 3.61. Prueba de Usabilidad.....	130
Tabla 3.62. Cronograma de capacitación referente a la aplicación web	136

RESUMEN EJECUTIVO

Actualmente, las aplicaciones web han adquirido un papel fundamental y esencial en las actividades cotidianas de las personas. Estas herramientas mejoran los procesos en términos de eficiencia y rapidez, proporcionando un mayor nivel de apoyo y asistencia al usuario final.

El objetivo de este proyecto es el desarrollo de una aplicación web empleando la tecnología responsive web design para el agendamiento de citas en el Centro Odontológico “DENTALSIS” con el fin de automatizar este proceso que se realizaba de forma manual con el riesgo de pérdida de información.

La elaboración de la aplicación web se realizó siguiendo los requisitos definidos por la institución. Se implementó la metodología Extreme Programming (XP) para la planificación, diseño, codificación y pruebas del producto, lo que facilitó una supervisión eficiente de los entregables de cada proceso solicitado.

El framework que se utilizó para el back-end fue ASP.NET debido a las ventajas que dispone, para el desarrollo front-end se implementó Bootstrap y la plantilla AdminLTE que proporcionan un conjunto de componentes predefinidos, el gestor de base de datos usado fue MySQL porque ofrece fiabilidad, rendimiento, escalabilidad, facilidad de uso y compatibilidad con diferentes sistemas operativos.

Palabras clave: Aplicación web, responsive web design, Agendamiento, ASP.NET, MySQL, Metodología XP.

ABSTRACT

Currently, web applications have acquired a fundamental and essential role in people's daily activities. These tools improve processes in terms of efficiency and speed, by providing a higher level of support and assistance to the end user.

The objective of this project is the development of a web application using the technology responsive web design for scheduling appointments at the Odontological Center "DENTALSIS" in order to automate this process that was done of manual form with the risk of loss of information.

The development of the web application was conducted in accordance with the requirements defined by the institution. The Extreme Programming (XP) methodology was used for planning, designing, coding and testing of the product, which allowed efficient monitoring of the products of each requested process.

The framework that was used for the back-end was ASP.NET because of the advantages it has, for front-end development was implemented Bootstrap and the AdminLTE template that provide a set of predefined components, the database manager used was MySQL because it offers reliability, performance, scalability, ease of use and compatibility with different operating systems.

Keywords: Web application, Responsive web design, Scheduling, ASP.NET, MySQL, XP Methodology.

Capítulo I.- MARCO TEÓRICO

1.1. Tema de investigación

APLICACIÓN WEB EMPLEANDO TECNOLOGÍA RESPONSIVE WEB DESIGN PARA EL AGENDAMIENTO DE CITAS EN EL CENTRO ODONTOLÓGICO “DENTALSIS”.

1.1.1. Planteamiento del Problema

En el transcurso de los años 80, surgió un creciente interés en la tecnología con el propósito de asegurar el éxito empresarial a nivel global. Esto se debió a su influencia en la determinación de la calidad y los costos de los productos, lo que a su vez definía la competitividad tanto en el presente como en el futuro. Esta tendencia tenía como objetivo lograr resultados financieros favorables y optimizar la toma de decisiones. En la actualidad, muchas empresas gestionan la tecnología en los niveles más altos de su estructura, ya que la consideran la clave del éxito y una herramienta poderosa para alcanzar el triunfo y mantener una ventaja competitiva [1]. En concreto, cada empresa busca la automatización de procesos como una estrategia para optimizar sus recursos. Esto les permite ofrecer un alto nivel de excelencia a sus clientes, con el fin de aumentar las ganancias de la organización. La automatización no solo impulsa la eficiencia interna, sino que también contribuye a perfeccionar la experiencia que se proporciona a los clientes.

En lo que respecta a la satisfacción de los usuarios en el ámbito de la salud, se subraya la importancia de ofrecer calidad, lo cual implica tener en cuenta las expectativas de los usuarios. En este contexto, proporcionar una atención deficiente conlleva consigo una serie de problemas y descontento entre los pacientes, lo que puede tener efectos perjudiciales tanto en la economía como en la imagen de las instituciones de salud [2]. La prestación de un servicio médico apropiado tiene como objetivo abordar las necesidades individuales de los pacientes, al tiempo que contribuye a incrementar el nivel de confianza y a mejorar la reputación del sector en cuestión. En última instancia, brindar una atención médica de calidad no solo promueve el bienestar de los pacientes,

sino que también refuerza la relación de confianza entre los profesionales de la salud y aquellos a quienes atienden.

El uso masivo de smartphones, tablets y smart TV han forzado a que los sistemas web se adapten a un único canal cuya estructura sea flexible, esto es conocido como Responsive Web Design (RWD). En el Ecuador, el Instituto Nacional de Estadística y Censos (INEC), en su reporte del año 2020, indica que el 25.3% de ecuatorianos poseen una computadora de escritorio, el 31.3% computadora portátil y el 51.5% smartphone o teléfono inteligente, relativo al acceso de internet a nivel nacional el porcentaje es de 70.7% [3]. En conclusión, la mayoría de la población del Ecuador tiene un dispositivo para acceder a internet lo cual nos indica que las páginas web deben implementar la tecnología RWD para proveer un ambiente cómodo y amigable al usuario.

Los centros odontológicos de la provincia de Tungurahua permanecen en constante crecimiento, por lo cual es necesario la aplicación de nuevas tecnologías para la optimización de los procesos referente a la gestión de datos y agendamiento de citas. Con lo mencionado anteriormente se mejora el procedimiento de obtención de turnos y el almacenamiento de información de cada paciente.

Con respecto al agendamiento de citas en una aplicación web empleando Tecnología Responsive Web Design se vuelve un punto esencial dado que el diseño de la interfaz es capaz de adaptarse a cualquier tipo de dispositivo y así permita una mejor fluidez de su contenido [4]. Mediante la utilización de esta tecnología se intenta reducir los costos y tiempos en desarrollo dado que solo es necesario un tipo de diseño el cual se ajuste a los tamaños de pantalla del equipo del usuario.

1.2. Antecedentes Investigativos

Analizando las investigaciones bibliográficas en revistas científicas en Internet y repositorios digitales de diferentes universidades internacionales y nacionales se identificó trabajos que sustentaran al presente trabajo de investigación:

En la tesis de Lizbet Rodríguez y Catherin Santillán [5] exponen el propósito de desarrollar un sistema informático web de control de citas médicas para el departamento de rehabilitación física de la Clínica San Juan de Dios. Este estudio se enmarca en la categoría de investigación aplicada y tiene como objetivo evaluar el estado actual del control de citas médicas en la clínica. Además, describen la metodología utilizada para analizar, diseñar y desarrollar el software, la cual se basa en el Proceso Unificado de Racional (RUP). Como resultado, se logró optimizar la gestión de citas médicas, lo que se tradujo en la reducción de tiempos, costos, recursos materiales y humanos. La metodología empleada en la tesis es particularmente interesante, ya que se caracteriza por ser iterativa e incremental. Se centra en la arquitectura del sistema y se guía por casos de uso específicos. Esta aproximación permitió satisfacer las necesidades de los usuarios finales dentro de un marco de tiempo y costos previamente establecidos.

En el trabajo investigativo de Neyra Liber [6], destaca que el propósito central de la tesis es llevar a cabo la implementación de un Sistema Web utilizando el enfoque hipotético-deductivo. Además, para recopilar información se recurrió a cuestionarios cuantitativos que incluían escalas de Likert. En conclusión, la implantación del Sistema Web resultó viable y beneficiosa. Este sistema mejoró la calidad del servicio, a la vez que garantizó eficiencia en la gestión de las citas médicas. Adicionalmente señala que este sistema también beneficia al personal de la institución al proporcionar información precisa y actualizada sobre los pacientes, lo que a su vez contribuye a un desempeño más efectivo en sus funciones.

Según Riccardi Yanitza, Vega Rosa y Miyares Ernesto [7] en el artículo se menciona que el objetivo principal consiste en crear e implementar un nuevo diseño del sitio, aplicando la filosofía de Responsive Web Design. El propósito de esta implementación es resolver el problema de visibilidad en diversos dispositivos móviles, así como incrementar la cantidad de visitas al sitio y sentar las bases para la integración de nuevos servicios. El resultado del proyecto fue la creación de un sitio web institucional basado en el modelo

Vista-Controlador. Este enfoque asegura un mantenimiento y actualización sencillos, lo que permite ahorrar tiempo estimado en estas tareas.

En el trabajo investigativo de Calle Paulina [8] se destaca que las herramientas basadas en la tecnología Responsive Web Design ofrecen la capacidad de que un sitio web se adapte de manera inteligente a la resolución de pantalla de diversos dispositivos, como celulares, portátiles, tablets, computadoras de escritorio u otros dispositivos de navegación en internet. Además, en este estudio se implementa la metodología ágil de desarrollo SCRUM, la cual se estructura en cinco fases:

1. Planificación: En esta etapa, el equipo define las tareas y objetivos del proyecto. Con el fin de crear un plan detallado para el trabajo a realizar en el período siguiente.
2. Creación: Aquí, el equipo trabaja en la implementación de las tareas planificadas, siguiendo ciclos cortos de desarrollo llamados "sprints".
3. Revisión: Al finalizar cada sprint, se lleva a cabo una revisión donde se muestra el trabajo realizado a los Stakeholders.
4. Retrospectiva: Luego de la revisión, el equipo analiza su desempeño durante el sprint, identificando lo que funcionó bien y áreas que podrían mejorar.
5. Avance: Cada sprint permite avanzar en el proyecto, ofreciendo mejoras funcionales y adaptándose a los cambios según la retroalimentación recibida y los ajustes realizados.

En el trabajo de titulación de Jurado Jeanneth [9] indica que: el presente proyecto de investigación pretende mejorar la relación entre médico - paciente, al proporcionar un servicio en línea para la asignación y cancelación de citas médicas. Este servicio está diseñado para facilitar la atención de los pacientes, especialmente en situaciones de emergencia, ya sea en sus hogares o en los consultorios. Además, en el proceso de recolección de información, se emplearon herramientas y técnicas para lograr una comprensión más clara sobre cómo se almacena la información de los pacientes en el Consultorio Médico Gutiérrez. Esto permitió obtener una comprensión precisa de las necesidades y, en consecuencia, brindar una solución confiable durante el

desarrollo del proyecto. Como resultado de este esfuerzo, se creó una aplicación que cumple con los requerimientos tanto de los médicos como de los pacientes.

1.3. Fundamentación Teórica

Gestión de Documentación Clínica

El propósito de la gestión de documentación clínica es planificar y administrar toda la información clínica generada a lo largo de los continuos procesos asistenciales del paciente. Se resume en tres aspectos:

- Gestionar la historia clínica

Para la gestión de historias clínicas se involucran algunas actividades como:

- a) Identificar: elaboración y actualización del fichero de pacientes.
- b) Diseñar y mantener actualizado el formato: estandarización de la documentación clínica.
- c) Clasificar, integrar y coordinar toda la información clínico-asistencial.
- d) Controlar la calidad: Valoración del contenido de los documentos utilizados en la asistencia.
- e) Garantizar la accesibilidad: Asegurar la disponibilidad de información acerca de pacientes y asegurar la confidencialidad.

- Gestionar y organizar los archivos de documentación e historias clínicas

Para proteger la información personal de los pacientes se debe:

- a) Custodiar, prestar y recepcionar las historias clínicas.
- b) Efectuar un seguimiento de la documentación prestada.
- c) Identificar, mantener y cuidar la documentación clínica
- d) Evaluar la actividad y control de calidad del Archivo de historias clínicas.

- Organizar y gestionar operativamente la codificación clínica

Para una óptima organización y manejo de la información de los pacientes es necesario cumplir con lo siguiente:

- a) Definir las fuentes de datos del sistema de información.
- b) Tratar la información clínica extraída de las historias.
- c) Controlar la calidad del sistema de información [10].

Historias Clínicas

La historia clínica es un documento donde se detalla la información clínica de un paciente, además se menciona los procesos médicos a los cuales ha sido sometido. Generalmente la historia clínica procede de la práctica clínica de una paciente y forma una parte importante en la atención integral y adecuada de una paciente, además representa el aval de la actuación del personal médico.

Características:

- Su práctica es obligatoria.
- Insustituible.
- Privada y pertenece al paciente.
- Provee confidencialidad [11].

Agendamiento de citas

Es el proceso por el cual un paciente reserva una cita médica, es decir una persona agenda una atención con un médico en donde el especialista se compromete a brindar un servicio de calidad y cumpliendo con todas las necesidades del usuario [12].

Web

Es un “Sistema lógico de acceso y búsqueda de la información disponible en Internet, cuyas unidades informativas son las páginas web”. Además, la Web introduce un concepto fundamental: la posibilidad de lectura universal, que

consiste en que una vez que la información esté disponible, se pueda acceder a ella desde cualquier ordenador, desde cualquier país, por cualquier persona autorizada, usando un único y simple programa. Para que esto fuese posible, se utilizan una serie de conceptos, el más conocido es el hipertexto [13].

Sistema de Información

Se refiere a un sistema conformado por elementos interconectados que tienen la responsabilidad de recopilar, almacenar, procesar y distribuir datos con el propósito de respaldar la toma de decisiones, el control y la coordinación dentro de una entidad organizacional. Estos sistemas operan con el objetivo de administrar y manipular información de manera precisa y eficiente [14].

Tipos de Sistemas de Información:

- De Procesamiento de Datos (TPS – Transactional Processing System).

La función de un TPS es la administración y el manejo de datos producidos en una transacción, una transacción es una interacción entre dos o más partes donde ocurre un intercambio de bienes o elementos de valor.

- Sistemas de Información para la Administración o Gerenciales (MIS- Management Information Systems).

Son una colección de sistemas de información que interactúan entre sí y que proporcionan información tanto para las necesidades de las operaciones como de la administración en la empresa. Realizan básicamente dos acciones: Resumir las transacciones almacenadas a través de los sistemas de procesamiento de transacciones y proporcionar dicha información resumida a gerentes de nivel medio, de forma periódica.

- Sistemas de Soporte a la Toma de Decisiones (DSS – Decision Support Systems).

Son aquellos sistemas orientados a ayudar la toma de decisiones para aquellos que se enfrentan a esta labor, los DSS son automáticos y computarizados y además suelen ser “interactivos, flexibles y adaptables” [15].

- Sistemas de información para ejecutivos (EIS – Executive Information Systems).

Es un sistema de información para ejecutivos que tiene aplicación a cualquier parte de la empresa, permite que la información que es recogida en la figura de informes se encuentre online a todo momento para que sea aprovechada por todas las partes de la organización [16].

- Sistemas Expertos o sistemas basados en el conocimiento (WKS – Knowledge Working Systems).

Promueven la creación de nuevo conocimiento y permiten que dicho conocimiento, así como la experiencia adquirida en su creación, se integre en la empresa. Son utilizados por trabajadores del conocimiento, por lo que están más relacionados con los productos y servicios que con la gestión de la empresa [17].

Aplicación Web

Las aplicaciones web, también conocidas como "sistemas web", son aquellas que se desarrollan e implementan en plataformas o sistemas operativos (como Windows, Linux o Mac). Sin embargo, en lugar de instalarse directamente en un dispositivo, estas aplicaciones se alojan en servidores en Internet o en redes locales (intranets). Aunque su apariencia es similar a la de las páginas web convencionales que solemos ver, en realidad, los sistemas web poseen capacidades altamente poderosas que ofrecen soluciones específicas para casos particulares.

Los sistemas web son compatibles con diversos navegadores web (como Chrome, Firefox, Internet Explorer y Opera) independientemente del sistema operativo utilizado. Estas aplicaciones web no requieren instalación individual

en cada computadora, ya que los usuarios se conectan al servidor donde está alojado el sistema [18].

Arquitectura

- Tres niveles

Divide la funcionalidad para optimizar el uso de recursos. Se consiguen soluciones mucho más flexibles y escalables. Los tres niveles son:

- **Cliente:** Contiene los componentes de usuario que son únicos para cada uno de ellos. Esto es la lógica de aplicación específica del usuario y la interfaz.
- **Aplicación:** Constituye un entorno multiusuario y mantiene las partes compartidas de la aplicación. Las operaciones con un uso intensivo de datos deben ejecutarse en este nivel.
- **Almacenamiento:** Es nivel de la base de datos. Se especializa en dar un servicio de persistencia a los datos de la aplicación y permite manejar grandes volúmenes de ellos.

- Cuatro niveles

La arquitectura en cuatro niveles nació como una evolución de la de tres para aplicaciones con interfaces de usuario complejas. Para ello integra el patrón de diseño MVC (Modelo-VistaControlador). Los cuatro niveles son:

- **Vista:** contiene los elementos de interfaz que presentan la información. Se asocian a las vistas del patrón MVC, aunque este nivel también puede incluir elementos de tipo controlador sencillos.
- **Controlador:** se trata de los componentes que intermedian entre los elementos de la interfaz y los del modelo del dominio. Son responsables del flujo de aplicación y controlan la navegación entre vistas.

- Dominio: son los objetos que modelan el dominio de la aplicación.
- Infraestructura: incluye el almacenamiento en bases de datos.
- Hexagonal

En esta arquitectura no se define una parte frontal hacia el usuario (front-end) y una parte trasera (back-end) como en los patrones arquitectónicos anteriores. Por el contrario, se considera que la aplicación se encuentra situada en el centro y que interacciona con el resto a través de una serie de transformadores que adaptan los mensajes de la aplicación al elemento con el cual se interacciona en cada caso [19].

- N-Capas

La arquitectura N-Capas es un enfoque común utilizado en el desarrollo de software para separar y organizar el código en capas lógicas y físicas. Cada capa tiene una responsabilidad específica y se comunica con las capas adyacentes a través de interfaces bien definidas.

- Estratificación Estricta: Cada capa solo puede comunicarse con las capas adyacentes inmediatas, esto quiere decir que una capa solo puede acceder a la capa que está directamente encima o debajo de ella en la jerarquía de capas.
- Estratificación Flexible: Permite que las capas se comuniquen con cualquier otra capa dentro del sistema, independientemente de su proximidad en la jerarquía de capas. Esto implica que una capa puede acceder directamente a cualquier otra capa, lo que facilita la flexibilidad en la implementación y la interacción entre componentes [20].

Responsive Web Design

Es una metodología de diseño y desarrollo web que busca crear sitios web que se adapten de manera automática y fluida a diferentes dispositivos y tamaños de pantalla. El objetivo principal es proporcionar una experiencia de usuario

óptima sin importar si el usuario accede al sitio web desde un ordenador de escritorio, una tablet, un teléfono móvil u otro dispositivo.

Responsive Web Design utiliza técnicas como el uso de rejillas fluidas, imágenes flexibles y consultas de medios CSS para ajustar el diseño y el contenido del sitio web según las características del dispositivo y su pantalla.

Las características clave del Responsive Web Design incluyen:

- Rejillas fluidas: Permiten que los elementos del diseño se redimensionen proporcionalmente en función del tamaño de la pantalla.
- Imágenes flexibles: Se ajustan automáticamente según el tamaño de la pantalla, evitando la descarga de imágenes innecesariamente grandes en dispositivos con pantallas pequeñas.
- Consultas de medios CSS: Se utilizan para aplicar estilos específicos a diferentes dispositivos y tamaños de pantalla. Esto permite adaptar el diseño y la disposición del contenido según las necesidades de cada dispositivo.
- Contenido adaptable: Se adapta el contenido del sitio web para que sea legible y accesible en diferentes tamaños de pantalla, evitando la necesidad de hacer zoom o desplazarse horizontalmente [21].

Adaptive Web Design

Es una metodología de diseño y desarrollo web que se enfoca en adaptar el diseño, el contenido y la funcionalidad de un sitio web a diferentes dispositivos y contextos de uso. El Adaptive Web Design utiliza múltiples versiones del sitio web conocidas como "diseños adaptativos" o "puntos de interrupción", cada una diseñada específicamente para adaptarse a un conjunto particular de dispositivos o características del usuario.

Algunas características clave del Adaptive Web Design son:

- Diseños adaptativos: Es la creación de múltiples versiones adaptadas a dispositivos específicos o grupos de dispositivos, cada versión se entregará al usuario en función de las características del dispositivo o del contexto.
- Detección de características del dispositivo: Para determinar qué versión del sitio web se debe entregar al usuario, se utilizan técnicas de detección de características del dispositivo, como la resolución de pantalla, el tamaño, la capacidad de procesamiento y el ancho de banda.
- Personalización del contenido y diseño: Cada versión del sitio web puede tener diseños y contenido personalizados para optimizar la experiencia del usuario en dispositivos específicos.
- Redireccionamiento del servidor: Para entregar la versión adecuada del sitio web al dispositivo del usuario, se utiliza la técnica de redireccionamiento del servidor.
- Enfoque en la experiencia del usuario: Proporciona una experiencia óptima al usuario en diferentes dispositivos, adaptando el diseño, el contenido y la funcionalidad del sitio web para cada uno de ellos [22].

Tecnologías Front-end

Son un conjunto de herramientas, lenguajes de programación y frameworks utilizados en el desarrollo de la capa de interfaz de usuario de aplicaciones web y móviles. Estas tecnologías se enfocan en la presentación visual, la interacción y la experiencia del usuario.

Dentro del área de front-end se trabaja con lenguajes como:

- HTML (lenguaje de marcado de hipertexto, se utilizan etiquetas que estructuran y organizan el contenido de la web).

- CSS (se encarga del formato y diseño visual de las páginas web escritas en html) para darle estructura y estilo al sitio.
- Javascript (un lenguaje de programación, rápido y seguro para programar centros de datos, consolas, teléfonos móviles o Internet) para complementar los anteriores y darles dinamismo a los sitios web.

HTML y CSS son lenguajes de marcado y estilo. Javascript es un lenguaje de programación [23].

Framework de diseño

También conocido como framework de UI (User Interface), es una estructura predefinida que proporciona una base sólida y consistente para el diseño y desarrollo de interfaces, además ofrece un conjunto de componentes, patrones de diseño, estilos, reglas y funcionalidades listas para usar, lo que permite a los desarrolladores agilizar y estandarizar el proceso de creación de interfaces.

Algunos de los beneficios que ofrece son:

- Ahorro de tiempo: Proporcionan componentes y estilos predefinidos, lo que permite a los desarrolladores ahorrar tiempo.
- Consistencia visual: Al momento de utilizarlos se asegura una apariencia y experiencia de usuario coherentes en toda la aplicación.
- Responsividad y adaptabilidad: Generalmente integran sistemas de rejilla y estilos responsivos, lo que simplifica la creación de interfaces que se ajustan automáticamente a diferentes tamaños de pantalla y dispositivos.
- Mejoras en la productividad: Los desarrolladores pueden concentrarse más en la lógica y funcionalidad de la aplicación en lugar de perder tiempo en aspectos visuales y de diseño.

- Comunidad y soporte activo: Poseen una gran comunidad de desarrolladores que comparten recursos, ejemplos, tutoriales y soluciones a problemas comunes.
- Actualizaciones y mejoras constantes: Los miembros de la comunidad trabajan para corregir errores, agregar nuevas funcionalidades y adaptarse a las últimas tendencias de diseño, lo que asegura que el framework esté actualizado y alineado con las mejores prácticas [24].

Tecnología Backend

Se refiere a las tecnologías y herramientas utilizadas en el desarrollo de la lógica del servidor y la gestión de bases de datos en una aplicación web. Estas tecnologías están enfocadas en el procesamiento de datos, la implementación de la lógica empresarial y la interacción con las bases de datos y otros sistemas [25].

Framework Backend

Es una herramienta o conjunto de bibliotecas y componentes que proporciona a los desarrolladores un conjunto predefinido de funcionalidades y estructuras para facilitar el desarrollo de la lógica del servidor y la gestión de bases de datos en una aplicación web.

Los frameworks Backend ayudan a los desarrolladores a construir aplicaciones web robustas y escalables, al proporcionar una estructura organizada y un conjunto de características comunes necesarias en el desarrollo de servidores [26].

Gestor De Base De Datos (DBMS)

Es un software diseñado para administrar y organizar grandes volúmenes de información de manera eficiente. Su función principal es permitir la creación, modificación y consulta de bases de datos, así como controlar el acceso y la seguridad de los datos almacenados [27].

Ventajas:

- Permiten organizar y estructurar los datos de manera coherente mediante tablas, relaciones y restricciones, lo que facilita el almacenamiento y la recuperación de la información.
- Gestionan el acceso concurrente a la base de datos, permitiendo que varios usuarios accedan y manipulen los datos al mismo tiempo de manera segura y consistente.
- Ofrecen mecanismos de control de acceso y seguridad, lo que significa que se pueden establecer permisos y roles de usuario para restringir el acceso a la información y protegerla contra accesos no autorizados.
- Garantizan la integridad y consistencia de los datos mediante la aplicación de restricciones y reglas definidas. Esto ayuda a evitar la corrupción de datos y asegura que los datos sean precisos y confiables.

Desventajas:

- Varios DBMS ser costosos tanto en términos de licencias como de mantenimiento.
- Algunos DBMS pueden tener limitaciones en cuanto a escalabilidad, especialmente cuando se trata de grandes volúmenes de datos o altas cargas de trabajo
- Ciertos DBMS están diseñados para un modelo de datos específico, lo que puede limitar la flexibilidad si tus necesidades cambian o si deseas utilizar diferentes modelos de datos en tu aplicación [28].

Servidor Local

Permite a los desarrolladores crear, probar y depurar aplicaciones web de forma segura y sin necesidad de una conexión a Internet. Proporciona un entorno aislado donde se pueden instalar y configurar todos los componentes necesarios para el desarrollo web, como el servidor web, el servidor de base de datos y el lenguaje de programación [29].

Metodologías de desarrollo de software

Son enfoques y prácticas sistemáticas que se utilizan para planificar, organizar, ejecutar y controlar el proceso de desarrollo de software. Estas metodologías proporcionan una estructura y guía para el equipo, permitiendo una forma más eficiente y efectiva de construir el software [30].

Existen varios tipos de metodologías de desarrollo de software, cada una con enfoques y características diferentes. A continuación, se presentan algunos de los tipos más comunes:

- Metodologías tradicionales: Se caracterizan por seguir un enfoque secuencial y lineal, con fases bien definidas y un énfasis en la planificación y documentación exhaustivas.
- Metodologías ágiles: Son enfoques de desarrollo de software que se centran en la entrega iterativa, la adaptabilidad a los cambios y la colaboración entre los miembros del equipo y los clientes. Estas metodologías valoran la flexibilidad, la comunicación y la respuesta rápida a los requisitos cambiantes [31].
- Metodologías en espiral: Es un enfoque de desarrollo de software que combina elementos de las metodologías tradicionales y las metodologías iterativas, se basa en la gestión de riesgos a lo largo del ciclo de desarrollo. La metodología en espiral se compone de varias fases que se recorren en forma de espiral, con cada vuelta representando una iteración del proceso [32].
- Metodologías iterativas: Son enfoques de desarrollo de software que se basan en la repetición de ciclos de trabajo o iteraciones para construir y mejorar gradualmente un producto. A medida que avanza el proyecto, se obtienen retroalimentaciones y se aplican los aprendizajes en cada iteración [33].
- Metodologías de desarrollo rápido de aplicaciones: Son enfoques de desarrollo de software que se centran en la entrega rápida de soluciones

funcionales. Estas metodologías se basan en la simplificación del proceso de desarrollo y la reducción de las actividades y etapas que no agregan valor directo al producto final [34].

- Metodologías de desarrollo basadas en modelos: Son enfoques que se centran en la utilización de modelos y representaciones abstractas del sistema para guiar el proceso de desarrollo. Estos modelos ayudan a comprender, diseñar, analizar y documentar el sistema antes de la implementación [35].
- Metodologías de desarrollo dirigidas por el dominio: Son enfoques de desarrollo de software que se centran en comprender y modelar el dominio del problema que el software debe resolver. Se basan en la colaboración estrecha entre los expertos en el dominio y los desarrolladores para capturar y representar correctamente el conocimiento y las reglas del negocio en el diseño y la implementación del software [36].
- Metodologías Lean: Son enfoques de gestión y mejora continua que se originaron en la industria manufacturera y que luego se aplicaron al desarrollo de software y otros campos. Se enfocan en eliminar el desperdicio, optimizar los procesos y maximizar el valor para el cliente [37].

Algunas de las metodologías de desarrollo de software más comunes son:

- Relationship Management Methodology (RMM)

La metodología de administración de relaciones, se refiere a un enfoque estructurado y sistemático para gestionar las relaciones con los clientes, proveedores, socios comerciales y otras partes interesadas. Se centra en establecer, desarrollar y mantener relaciones sólidas y beneficiosas a lo largo del tiempo.

Algunos aspectos clave de la metodología de administración de relaciones son:

- Análisis y comprensión de las partes interesadas.
- Establecimiento de relaciones sólidas.
- Desarrollo y mantenimiento de relaciones a largo tiempo.
- Medición y evaluación del rendimiento de las relaciones.

La metodología de administración de relaciones es especialmente relevante en entornos empresariales donde las relaciones a largo plazo son fundamentales para el éxito. Al gestionar adecuadamente las relaciones con las partes interesadas, las organizaciones pueden construir alianzas sólidas, obtener el apoyo necesario y lograr una mayor satisfacción del cliente [38].

- Object-Oriented Hypermedia Design Methodology (OOHDM)

La Metodología de Diseño Orientado a Objetos e Hipermedia (Object-Oriented Hypermedia Design Methodology, OOHDM) es un enfoque para el diseño de aplicaciones hipermedia que combina principios de diseño orientado a objetos y conceptos de hipermedia. Fue propuesta por los investigadores de la Universidad de Roma en la década de 1990 [39].

La metodología OOHDM se utiliza para diseñar aplicaciones hipermedia interactivas que permiten a los usuarios navegar y acceder a diferentes tipos de información de manera no lineal. Entre las características y principios fundamentales de esta metodología se encuentran:

- Modelo de objetos: Representa las entidades y relaciones clave en el dominio de la aplicación, se utiliza para estructurar y organizar la información hipermedia.

- Hipertexto e hipermedia: Permite a los usuarios navegar de forma no secuencial a través de diferentes elementos de información, como documentos, imágenes, videos, etc.
- Separación de la estructura y la presentación: Esto permite una mayor flexibilidad y reutilización al cambiar la apariencia de la aplicación sin afectar su estructura subyacente.
- Ciclo de vida iterativo: Involucra etapas de análisis, diseño, implementación y evaluación. Cada iteración permite refinar y mejorar el diseño de la aplicación [40].
- Extreme Programming (XP)

Es una metodología ágil de desarrollo de software que se enfoca en la entrega rápida y continua de software de alta calidad. Fue creado por Kent Beck a finales de la década de 1990 y se basa en una serie de principios y prácticas que promueven la colaboración, la retroalimentación continua y la adaptabilidad [41].

Algunos de los aspectos relevantes de la metodología XP son:

- Comunicación constante: Esto ayuda a mantener una comprensión compartida de los requisitos, resolver problemas rápidamente y mantener un enfoque centrado en el cliente.
- Desarrollo iterativo e incremental: El desarrollo se realiza en ciclos cortos y repetitivos, conocidos como iteraciones. Cada iteración se enfoca en entregar una funcionalidad completa y lista para su uso. Esto permite una entrega rápida y la posibilidad de adaptarse a medida que se obtiene retroalimentación y se realizan ajustes.

- Pruebas continuas: Se enfatiza la escritura de pruebas automatizadas desde el inicio del proyecto. Las pruebas se ejecutan de forma continua para asegurar que el software cumpla con los requisitos y funcione correctamente. Esto permite detectar errores tempranos y facilita los cambios y mejoras continuas.
- Diseño simple: Se busca mantener un diseño simple y limpio. Se favorecen las soluciones sencillas y se permite la refactorización del código para mejorar su calidad y mantenibilidad.
- Integración continua: Esto implica combinar y probar los cambios de forma regular. Esto ayuda a detectar y solucionar problemas de integración de manera temprana y garantiza una base de código estable.
- Valorar la retroalimentación: Se busca obtener retroalimentación constante de los clientes y usuarios finales para mejorar continuamente el producto. Los requisitos pueden cambiar a medida que se obtiene más conocimiento y experiencia, y se priorizan las necesidades del cliente [42].
- Rational Unified Process (RUP)

Es una metodología de desarrollo de software que se centra en la gestión del ciclo de vida del software. Fue desarrollada por Rational Software Corporation (adquirida posteriormente por IBM) y se basa en principios de ingeniería de software, prácticas de gestión de proyectos y enfoques orientados a objetos [43].

La metodología RUP se distingue por los siguientes factores:

- Enfoque iterativo e incremental: Se fundamenta en la idea de dividir el desarrollo de software en fases iterativas y

entregables incrementales. Cada iteración se enfoca en el desarrollo y entrega de una parte del sistema funcional.

- **Arquitectura centrada en casos de uso:** Se basa en la identificación y análisis de los casos de uso, que representan las interacciones entre los actores y el sistema. La arquitectura del sistema se define en función de estos casos de uso.
- **Enfoque basado en componentes reutilizables:** Promueve la reutilización de componentes de software a través del desarrollo de componentes genéricos y la adopción de estándares industriales. Esto permite una mayor eficiencia y calidad en el desarrollo.
- **Orientación a la gestión de riesgos:** Presta especial atención a la gestión de riesgos a lo largo del ciclo de vida del proyecto. Se realizan evaluaciones de riesgos y se toman medidas para mitigarlos de manera temprana y efectiva.
- **Enfoque colaborativo y multidisciplinario:** Fomenta la colaboración entre los diferentes roles y disciplinas en un equipo de desarrollo, como analistas, diseñadores, desarrolladores y probadores. Se promueve la comunicación efectiva y la participación activa de todos los miembros del equipo.
- **Personalización y adaptabilidad:** Es una metodología flexible que se puede adaptar a las necesidades y características específicas de cada proyecto. Permite la personalización y selección de las mejores prácticas y técnicas adecuadas para el contexto del proyecto [44].

Página Maestra

Una página maestra o página principal es una estructura base que se utiliza para diseñar y mantener la estabilidad visual y estructural en un sitio web o una aplicación, se establecen elementos como el encabezado, el pie de página, la barra de navegación y otros componentes que se repiten en todas las páginas. También se definen la estructura de columnas, los estilos de tipografía, los colores y otros aspectos visuales que se mantendrán constantes en todo el sitio. Algunos de los beneficios que proporciona son:

- **Consistencia visual:** Al definir los elementos comunes y el diseño general en la página maestra, se asegura que todas las páginas o pantallas tengan una apariencia consistente, lo que brinda una experiencia visual más agradable y profesional para los usuarios.
- **Eficiencia en el diseño:** Los elementos y la estructura de diseño se definen una vez en la página maestra y se aplican automáticamente a todas las páginas vinculadas a ella. Esto agiliza el proceso de diseño y permite realizar cambios de manera más eficiente.
- **Facilidad de actualización:** Facilita la actualización y el mantenimiento del sitio, ya que no es necesario editar cada página por separado.
- **Mejor experiencia del usuario:** Los usuarios pueden navegar y encontrar información de manera más intuitiva, ya que los elementos comunes están ubicados en las mismas áreas en todas las páginas.
- **Flexibilidad y personalización:** La página maestra proporciona una guía y un marco, pero se pueden realizar modificaciones específicas en cada página según las necesidades y los objetivos específicos [45].

Mockflow

Es una herramienta online utilizada en la creación de prototipos de interfaces de usuario, diagramas de flujo y wireframes interactivos. Mockflow ofrece una interfaz sencilla, ya que permite al usuario construir velozmente

representaciones visuales de las interfaces de usuario, además posee gran variedad de componentes y plantillas personalizables [46].

Patrón Singleton

Es un patrón de diseño del tipo creacional que se utiliza para garantizar que una clase tenga una única instancia y proporcionar un punto de acceso global a dicha instancia.

Posee un constructor privado que evita que se creen instancias directamente desde fuera de la clase, una instancia estática que almacena la única instancia de la clase y un método estático de acceso que permite acceder a la instancia única de la clase. Este método se encarga de crear la instancia si aún no existe y la devuelve en todas las llamadas subsiguientes [47].

El uso del patrón Singleton en el diseño de software puede proporcionar varios beneficios, que incluyen:

- **Acceso global:** Proporciona un punto de acceso global a la única instancia de la clase, lo que facilita la interacción con la misma desde cualquier parte del código.
- **Garantía de una única instancia:** Evita la creación accidental de múltiples instancias y asegura que todos los usuarios accedan a la misma instancia.
- **Ahorro de recursos:** Se evita el consumo innecesario de recursos, como memoria y tiempo de procesamiento, que se requerirían para crear múltiples instancias de la misma clase.
- **Coherencia y consistencia:** Se garantiza la coherencia y consistencia en el estado y el comportamiento de la clase en toda la aplicación, ya que todos los usuarios comparten la misma instancia.
- **Facilidad de implementación:** El patrón Singleton es relativamente fácil de implementar y entender. Con solo unas pocas líneas de código,

se puede asegurar que una clase tenga una única instancia y proporcionar un acceso global a ella.

- Flexibilidad y extensibilidad: El patrón Singleton permite que la instancia única se acceda y se utilice en diferentes partes del código, lo que proporciona flexibilidad y extensibilidad al sistema. Además, si en el futuro se requiere cambiar la implementación del Singleton, se puede hacer sin afectar el resto del código que lo utiliza [48].

Métodos AJAX

Permite enviar y recibir datos de forma asíncrona con el servidor sin necesidad de recargar toda la página. Utiliza una combinación de tecnologías, incluyendo JavaScript, XML (aunque también se pueden utilizar otros formatos de datos como JSON) y el objeto XMLHttpRequest [49].

Los principales beneficios de AJAX son:

- Permite actualizar partes específicas de una página web sin tener que recargarla por completo, lo que proporciona una experiencia más fluida e interactiva para el usuario.
- Al realizar solicitudes asíncronas al servidor, se reduce la cantidad de datos que se envían y se minimiza el tiempo de espera, lo que mejora la velocidad y el rendimiento general de la aplicación web.
- Permite cargar y mostrar contenido dinámicamente, responder a acciones del usuario de forma inmediata y proporcionar retroalimentación en tiempo real, lo que mejora la experiencia de usuario.
- Permite separar la lógica del cliente y del servidor, ya que las solicitudes y respuestas pueden manejarse de manera independiente, lo que facilita el desarrollo y el mantenimiento del código.

- Permite reutilizar componentes y servicios del servidor, ya que las solicitudes AJAX pueden comunicarse con APIs y servicios web para obtener o enviar datos.
- Al minimizar la cantidad de datos que se transmiten y reducir la carga en el servidor, AJAX contribuye a una mejor escalabilidad de las aplicaciones web, permitiendo un mayor número de usuarios simultáneos [50].

1.4. Objetivos

1.4.1. Objetivo General

Implantar una Aplicación Web empleando Tecnología Responsive Web Design para el agendamiento de citas en el Centro Odontológico “DENTALSIS”.

1.4.2. Objetivos Específicos

- Identificar el proceso actual del agendamiento de citas para determinar los requisitos del sistema.
- Analizar la Tecnología Responsive Web Design en la creación y desarrollo de aplicaciones web.
- Desarrollar una Aplicación Web empleando Tecnología Responsive Web Design para el agendamiento de citas en el Centro Odontológico “DENTALSIS”.

Capítulo II.- METODOLOGÍA

2.1. Materiales

De acuerdo a la naturaleza del proyecto, las técnicas que se emplearon para la recolección de información fue una entrevista al odontólogo principal con la finalidad de conocer los requerimientos e inconvenientes sugeridos para el desarrollo de la aplicación. Del mismo modo se realizó una encuesta destinada a los pacientes del centro odontológico, con el objetivo de medir la aceptación e impacto que podría generar una aplicación web de agendamiento de citas odontológicas.

Guía de entrevista aplicada al odontólogo principal del consultorio.

DATOS PERSONALES			
Apellidos y Nombres:			
Institución:			
Cargo:			
Objetivo: Recopilar información acerca del proceso de agendamiento de citas en el Centro Odontológico “DENTALSIS”.			
Nº	Pregunta	Respuesta	Observación
1	¿Qué sector de la población por lo general acude a reservar una cita?		
2	¿Qué información del paciente es necesaria para su registro?		
3	¿Cómo es el proceso actual de agendamiento de citas?		
4	¿Cuánto tiempo aproximadamente se demora en el agendamiento de citas?		
5	¿Cuáles son los inconvenientes que se han presentado al momento del agendamiento de citas?		
6	Utiliza alguna aplicación para realizar el agendamiento de citas		

7	¿Qué problema ocasiona este tipo de proceso manual?		
8	¿Qué persona se encarga del agendamiento y registro de pacientes?		
9	¿Quiénes tienen acceso a la información?		
10	¿Cuál es su opinión acerca de implementar una solución informática que ayude en la gestión de las actividades que se realizan en el centro odontológico?		
Conclusión:			

Tabla 2.1. Guía de entrevista aplicada al odontólogo principal del consultorio.

Elaborado por: El investigador

Encuesta aplicada a pacientes del Centro Odontológico “DENTALSIS”

Objetivo: Recopilar información acerca del proceso de agendamiento de citas en el Centro Odontológico “DENTALSIS”.

CUESTIONARIO:

1. ¿Durante cuánto tiempo acude al centro odontológico?

- Menos de 2 meses
- 2 a 6 meses
- 6 meses a un año
- 1 a 2 años
- Más de 2 años

2. ¿Con qué frecuencia acude al centro odontológico a realizarse una consulta?
- 1 vez a la semana
 - 1 vez al mes
 - 1 vez al año
 - Cuando presenta molestias
3. ¿Se siente satisfecho con la atención brindada por el centro odontológico?
- Muy satisfecho
 - Algo satisfecho
 - Ni satisfecho/Ni insatisfecho
 - Algo insatisfecho
 - Muy insatisfecho
4. ¿Cree usted que el consultorio debe seguir llevando el control de la documentación de forma física?
- Si
 - No
5. ¿Tiene dificultad para reservar una cita?
- Siempre
 - Casi siempre
 - En ocasiones
 - Casi nunca
 - Nunca
6. ¿Cree usted que se está realizando de forma ordenada y eficaz las reservas de citas?
- Si
 - No

7. ¿De qué modo realiza la reserva de una cita en el consultorio?
- Vía online
 - Vía telefónica
 - Consultorio odontológico
8. ¿Cree usted que es necesario un sistema informático para registrar la información dentro del consultorio para evitar pérdida de datos?
- Si
 - No
9. ¿Considera que el tiempo de toma de información personal es excesivo?
- Si
 - No
10. ¿Cuánto tiempo usualmente tiene que esperar para ser atendido en su cita?
- Menos de 10 minutos
 - 10 a 20 minutos
 - 20 a 30 minutos
 - Más de una hora
11. ¿Considera usted que se podría agilizar el proceso de asignar turnos del centro odontológico?
- Si
 - No
12. ¿Considera positivo que un sistema informático mejore el proceso de agendamiento de citas?
- Totalmente de acuerdo
 - De acuerdo
 - Ni de acuerdo/Ni en desacuerdo
 - En desacuerdo
 - Totalmente en desacuerdo

2.2. Métodos

2.2.1. Modalidad de la investigación

El presente trabajo se contextualizó en la modalidad de investigación de campo y documental-bibliográfica.

- Investigación de campo

La modalidad fue de campo a causa que se trató de obtener información correspondiente a los procedimientos y servicios que proveen al cliente directamente con el personal implicado.

- Investigación documental-bibliográfica

La modalidad fue documental-bibliográfica debido a que se sustentó en fuentes como libros, documentos técnicos, tesis del área de sistemas, revistas, artículos científicos y fichas empleadas en el entorno médico para la recolección de información.

2.2.2. Población y Muestra

En el marco de esta investigación, la población considerada abarca a todos los miembros del centro odontológico “DENTALSIS”. En cuanto a la cantidad de pacientes, se focaliza en aquellos que asisten con mayor regularidad o que están en proceso de tratamiento.

Población	Número	Porcentaje
Odontólogo	1	3.23%
Pacientes	30	96.77%
Total	31	100%

Tabla 2.2. Población

Elaborado por: El investigador

Considerando que el número de población a ser investigada no sobrepasa de cien elementos se ocupó la totalidad del universo sin que sea fundamental la utilización de muestras representativas.

2.2.3. Recolección de la información

Para la aplicación del cuestionario a los pacientes del centro odontológico se utilizó un formulario de Google, ya que la misma plataforma nos brinda la tabulación de los datos.

Validación del instrumento

Con el objetivo de que la información obtenida sea lo más cercana a la realidad se utiliza el coeficiente Alfa de Cronbach para validar la encuesta.

Pregunta	Varianza del Ítem	Varianza del Instrumento
1	1,4933	53,84
2	0,7289	
3	0,5733	
4	0,1956	
5	1,6767	
6	0,1156	
7	0,3822	
8	0,0322	
9	0,2322	
10	1,2489	
11	0,1600	
12	0,5289	
Sumatoria	7,3678	

Tabla 2.3. Varianza de los resultados obtenidos en la encuesta aplicada

Elaborado por: El investigador

Cálculo del Coeficiente de Alfa de Cronbach

$$\alpha = \frac{K}{K - 1} \left[1 - \frac{\sum s^2}{s^2} \right]$$

K: Número de Ítems del instrumento

$\sum s^2$: Sumatoria de las varianzas de los ítems

s^2 : Varianza total del instrumento

Donde:

K: 12

$$\alpha = \frac{12}{11} \left[1 - \frac{7.3678}{53.84} \right]$$

$\sum s^2 : 7.3678$

$$\alpha = 0.94$$

$s^2 : 53.84$

El resultado del coeficiente de Alfa de Cronbach en este caso tiene un valor de 0.94, lo que nos indica que el instrumento empleado tiene un alto grado de confiabilidad.

Resultados de la encuesta

Los resultados de la encuesta aplicada a 30 pacientes del Centro Odontológico “DENTALSIS” son los siguientes:

Pregunta 1: ¿Durante cuánto tiempo acude al centro odontológico?

INDICADORES	FRECUENCIA	PORCENTAJE
Menos de 2 meses	4	13,3%
2 a 6 meses	3	10,0%
6 meses a un año	11	36,7%
1 a 2 años	7	23,3%
Más de 2 años	5	16,7%
Total	30	100%

Tabla 2.4. Tabulación de resultados de la pregunta 1

Elaborado por: El investigador

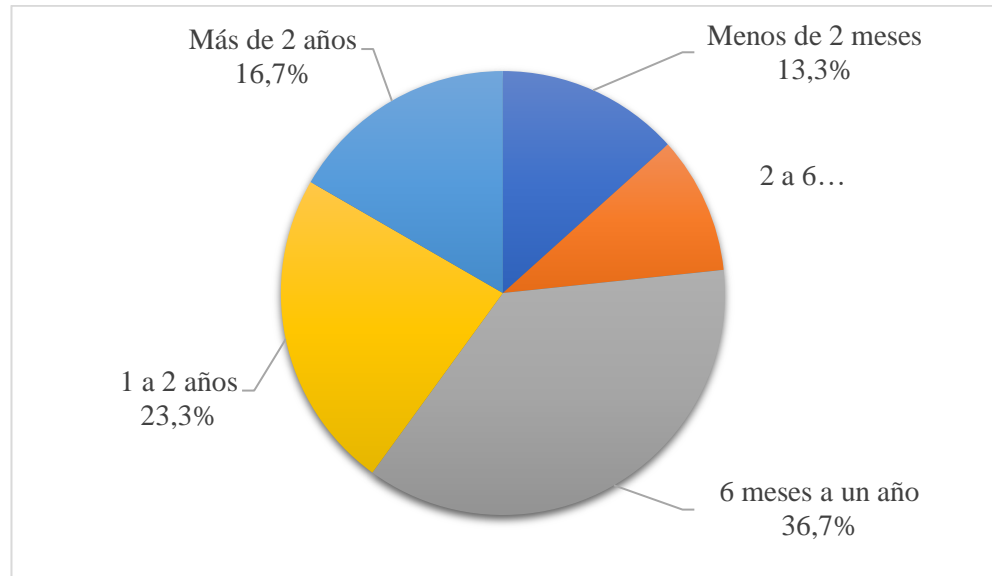


Gráfico 2.1. Tiempo que acuden los pacientes al centro odontológico

Elaborado por: El investigador

Análisis e Interpretación: Según los datos presentados en el gráfico 2.1, el 36.7% indica que los pacientes acuden en el lapso de 6 meses a un año, el 23.3% son pacientes que acuden entre 1 a 2 años, el 16.7% acuden por más de 2 años, el 13.3% son pacientes que acuden por menos de 2 meses y finalmente el 10% de los pacientes acuden entre 2 a 6 meses. Lo cual evidencia que la mayor parte de los pacientes encuestados han acudido con anterioridad al centro odontológico “DENTALSIS” pero también se observa que existen pacientes nuevos, los cuales acuden por recomendaciones de parte de familiares o amigos.

Pregunta 2: ¿Con qué frecuencia acude al centro odontológico a realizarse una consulta?

INDICADORES	FRECUENCIA	PORCENTAJE
1 vez a la semana	0	0,0%
1 vez al mes	10	33,3%
1 vez al año	8	26,7%
Cuando presenta molestias	12	40,0%
Total	30	100%

Tabla 2.5. Tabulación de resultados de la pregunta 2

Elaborado por: El investigador

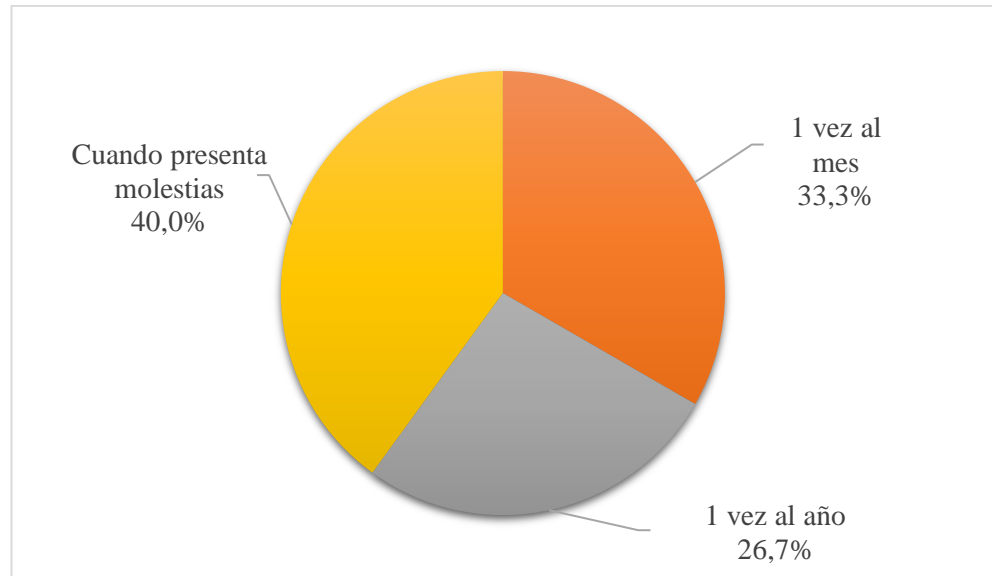


Gráfico 2.2. Frecuencia de los pacientes a realizarse una consulta

Elaborado por: El investigador

Análisis e Interpretación: Según los datos presentados en el gráfico 2.2, denota que el 40% de los pacientes acuden cuando presentan alguna molestia, el 33.3% acuden 1 vez al mes y el 26.7% acuden una vez al año, por lo tanto, la mayoría de los pacientes encuestados acuden a una consulta solo cuando presentan molestias en sus muelas o dientes, por otro lado, un grupo de pacientes se realizan anualmente chequeos o consultas.

Pregunta 3: ¿Se siente satisfecho con la atención brindada por el centro odontológico?

INDICADORES	FRECUENCIA	PORCENTAJE
Muy satisfecho	17	56,7%
Algo satisfecho	8	26,7%
Ni satisfecho/ Ni insatisfecho	5	16,7%
Algo insatisfecho	0	0,0%
Muy insatisfecho	0	0,0%
Total	30	100%

Tabla 2.6. Tabulación de resultados de la pregunta 3

Elaborado por: El investigador

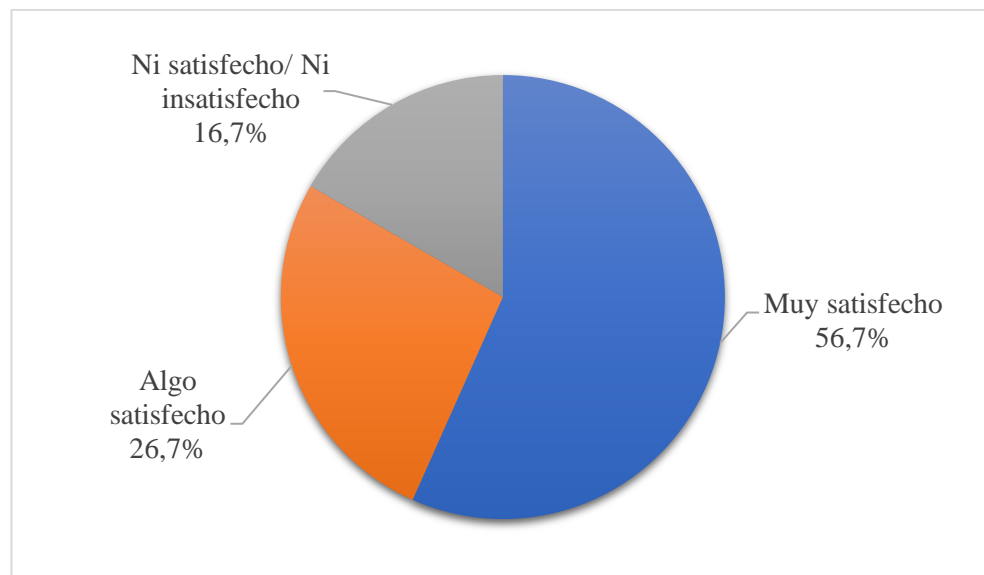


Gráfico 2.3. Nivel de satisfacción referente a la atención

Elaborado por: El investigador

Análisis e Interpretación: Según los datos presentados en el gráfico 2.3, indica que el 56.7% de los pacientes se encuentran muy satisfechos, el 26.7% están algo satisfechos y el 16.7% no están ni satisfechos ni insatisfechos con la atención que brinda el centro odontológico, entonces se puede decir que la atención es de total agrado para los pacientes a pesar de tener ciertos inconvenientes.

Pregunta 4: ¿Cree usted que el consultorio debe seguir llevando el control de la documentación de forma física?

INDICADORES	FRECUENCIA	PORCENTAJE
Si	22	73,3%
No	8	26,7%
Total	30	100%

Tabla 2.7. Tabulación de resultados de la pregunta 4

Elaborado por: El investigador

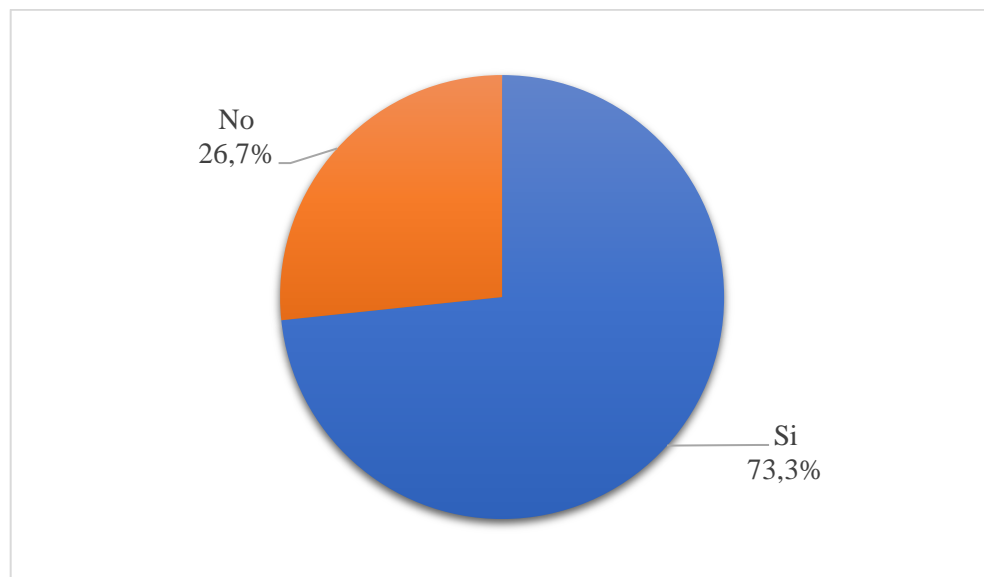


Gráfico 2.4. ¿Cree usted que el consultorio debe seguir llevando el control de la documentación de forma física?

Elaborado por: El investigador

Análisis e Interpretación: Según los datos presentados en el gráfico 2.4, la mayor parte de pacientes representados por el 73.3% creen que la mejor manera de llevar la documentación del consultorio es de forma física, frente a un 26.7% que opina lo contrario. Se determina que al llevar la documentación de forma física se puede correr el riesgo de pérdida de información.

Pregunta 5: ¿Tiene dificultad para reservar una cita?

INDICADORES	FRECUENCIA	PORCENTAJE
Siempre	1	3,3%
Casi siempre	6	20,0%
En ocasiones	7	23,3%
Casi nunca	3	10,0%
Nunca	13	43,3%
Total	30	100%

Tabla 2.8. Tabulación de resultados de la pregunta 5

Elaborado por: El investigador

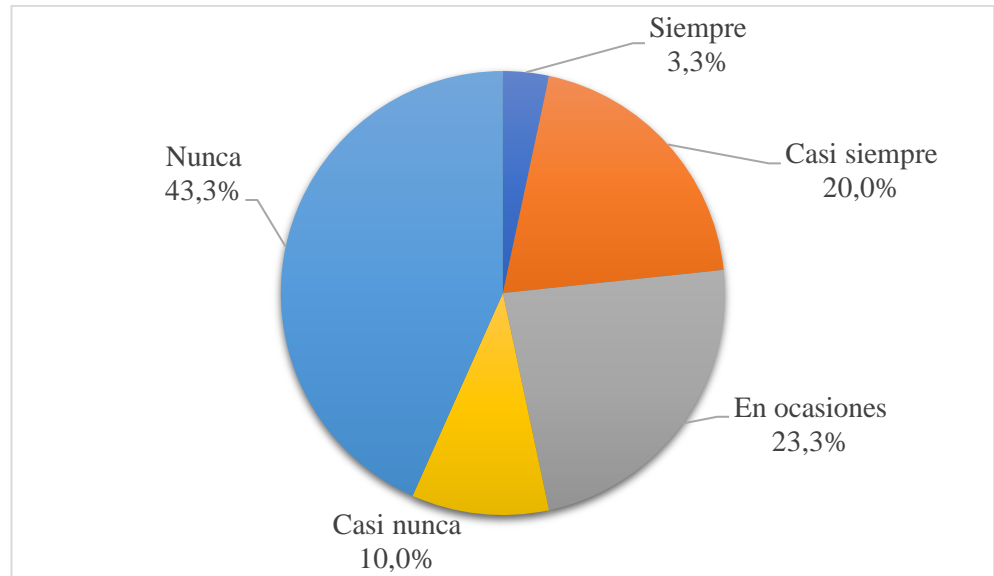


Gráfico 2.5. Dificultad para reservar una cita

Elaborado por: El investigador

Análisis e Interpretación: Según los datos presentados en el gráfico 2.5, el 43.3% nunca ha presentado dificultades, el 23.3% en ocasiones presenta dificultades, el 20% casi siempre presenta dificultades, el 10% casi nunca presenta dificultades y el 3.3% siempre presenta dificultades en reserva de citas. Se evidencia que la mayor parte de pacientes no ha presentado quejas acerca de problemas al momento de reservar una cita.

Pregunta 6: ¿Cree usted que se está realizando de forma ordenada y eficaz las reservas de citas?

INDICADORES	FRECUENCIA	PORCENTAJE
Si	26	86,7%
No	4	13,3%
Total	30	100%

Tabla 2.9. Tabulación de resultados de la pregunta 6

Elaborado por: El investigador

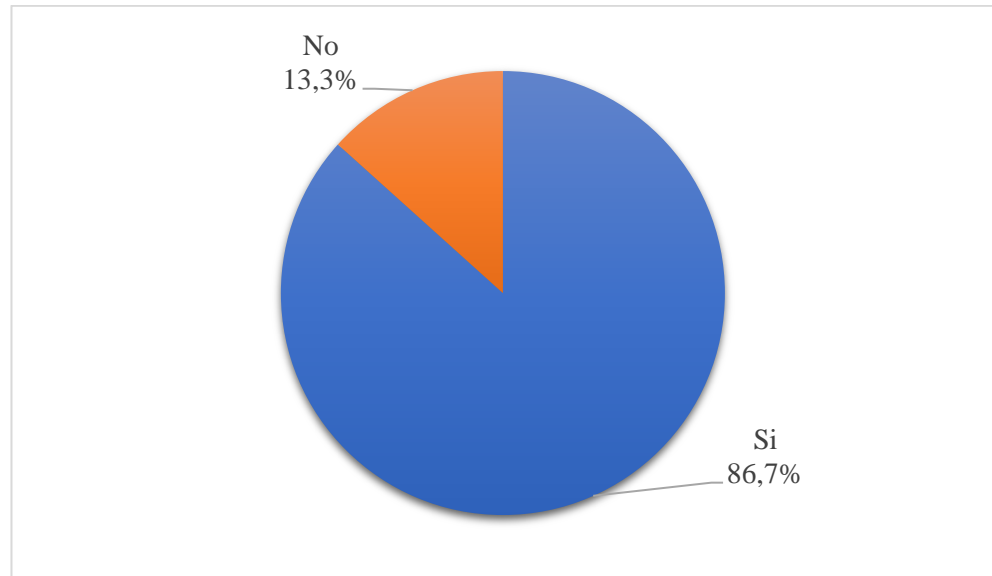


Gráfico 2.6. ¿Cree usted que se está realizando de forma ordenada y eficaz las reservas de citas?

Elaborado por: El investigador

Análisis e Interpretación: Según los datos presentados en el gráfico 2.6, el 86.7% de los pacientes encuestados opinan que se está realizando de forma ordenada y eficaz la reserva de citas, mientras que el 13.3% opina lo contrario. Por lo tanto, el proceso de reserva de citas logra su propósito de una manera ordenada.

Pregunta 7: ¿De qué modo realiza la reserva de una cita en el consultorio?

INDICADORES	FRECUENCIA	PORCENTAJE
Vía online	8	26,7%
Vía telefónica	18	60,0%
Consultorio odontológico	4	13,3%
Total	30	100%

Tabla 2.10. Tabulación de resultados de la pregunta 7

Elaborado por: El investigador

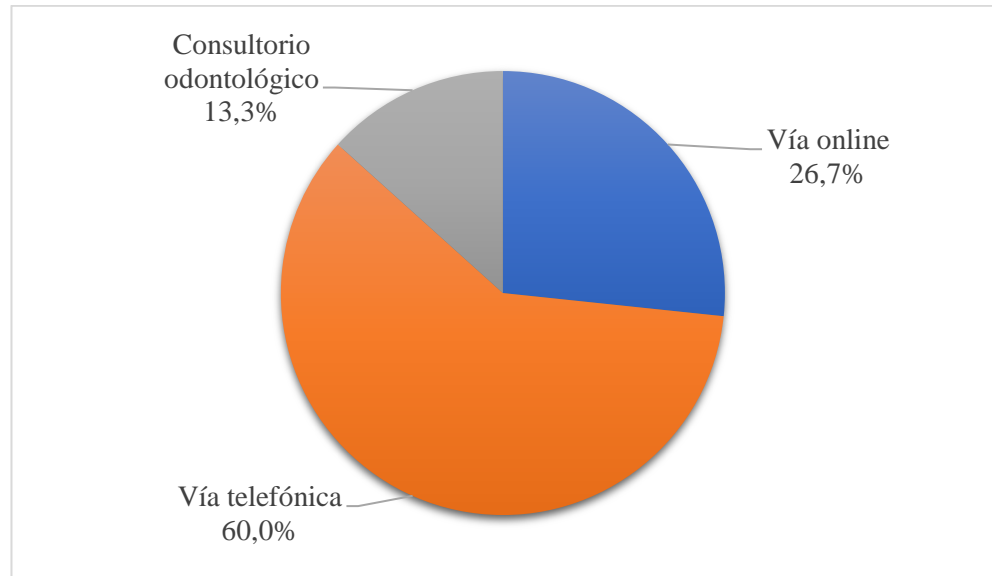


Gráfico 2.7. Formas que se realiza la reserva de una cita en el consultorio

Elaborado por: El investigador

Análisis e Interpretación: Según los datos presentados en el gráfico 2.7, el 60% de los pacientes reservan su cita vía telefónica, el 26.7% reservan su cita vía online y el 13.3% lo realizan acudiendo al consultorio odontológico. Se deduce que los pacientes buscan la forma más sencilla, rápida y eficiente para reservar una cita en el consultorio.

Pregunta 8: ¿Cree usted qué es necesario un sistema informático para registrar la información dentro del consultorio para evitar perdida de datos?

INDICADORES	FRECUENCIA	PORCENTAJE
Si	29	96,7%
No	1	3,3%
Total	30	100%

Tabla 2.11. Tabulación de resultados de la pregunta 8

Elaborado por: El investigador

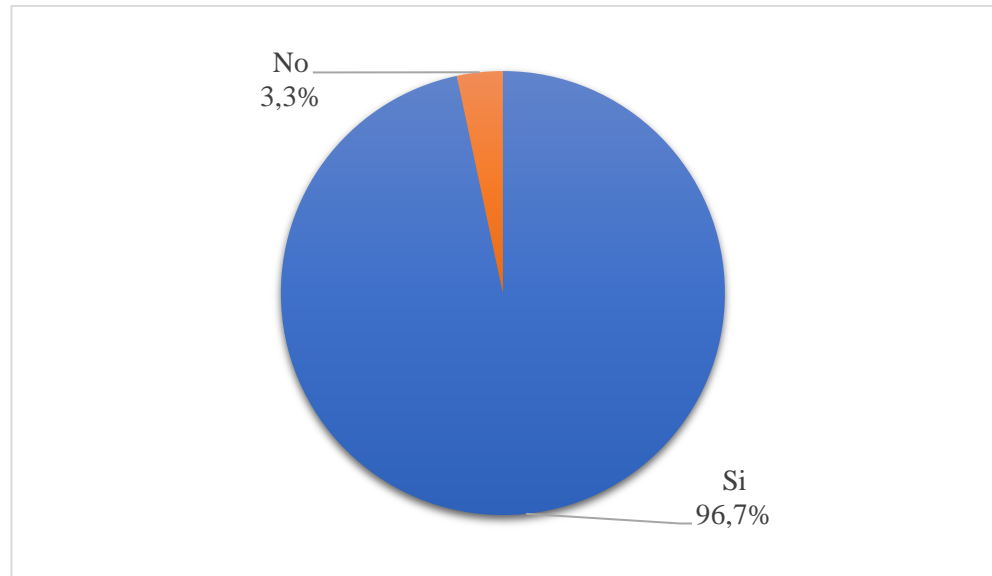


Gráfico 2.8. ¿Cree usted qué es necesario un sistema informático para registrar la información dentro del consultorio para evitar pérdida de datos?

Elaborado por: El investigador

Análisis e Interpretación: Según los datos presentados en el gráfico 2.8, el 96.7% de los pacientes encuestados opinan que es necesario un sistema informático para registrar la información dentro del consultorio para evitar pérdida de datos mientras que el 3.3% opinan lo contrario. Se determina que gracias al avance tecnológico la información se puede conservar de forma más segura, para así evitar pérdidas de datos.

Pregunta 9: ¿Considera que el tiempo de toma de información personal es excesivo?

INDICADORES	FRECUENCIA	PORCENTAJE
Si	11	36,7%
No	19	63,3%
Total	30	100%

Tabla 2.12. Tabulación de resultados de la pregunta 9

Elaborado por: El investigador

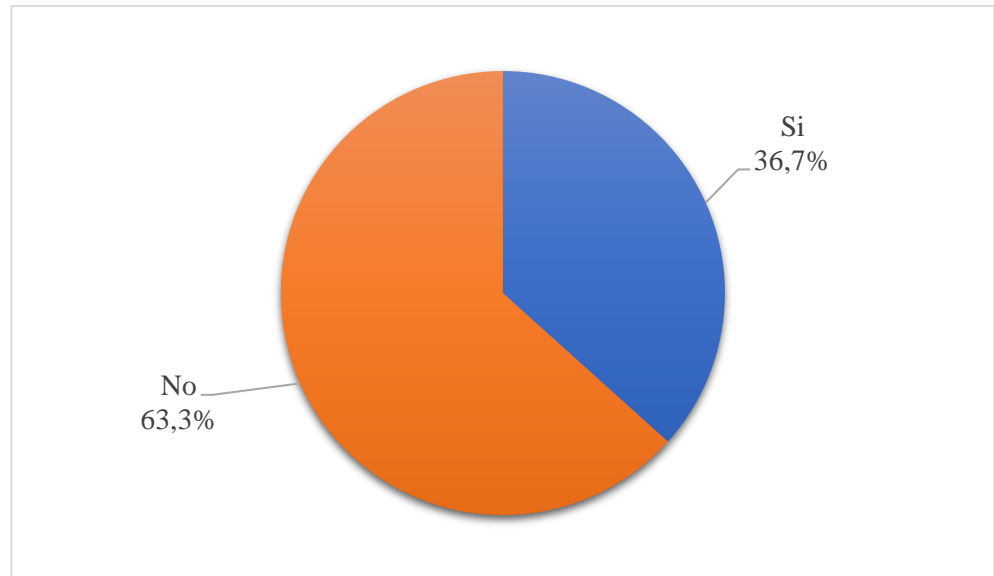


Gráfico 2.9. Toma del tiempo se considera excesivo

Elaborado por: El investigador

Análisis e Interpretación: Según los datos presentados en el gráfico 2.9, el 63.3% considera que el tiempo de toma de información personal no es excesivo, mientras que el 36.7% opina lo contrario. Se evidencia que la mayor parte de los pacientes encuestados están de acuerdo que el tiempo es el necesario para tomar los datos personales.

Pregunta 10: ¿Cuánto tiempo usualmente tiene que esperar para ser atendido en su cita?

INDICADORES	FRECUENCIA	PORCENTAJE
Menos de 10 minutos	12	40,0%
10 a 20 minutos	7	23,3%
20 a 30 minutos	6	20,0%
Más de una hora	5	16,7%
Total	30	100%

Tabla 2.13. Tabulación de resultados de la pregunta 10

Elaborado por: El investigador

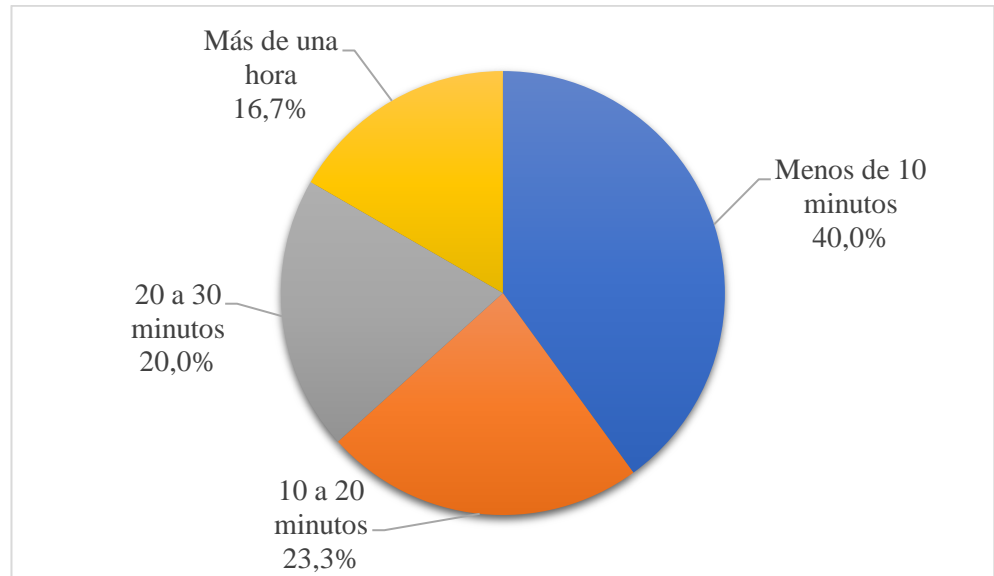


Gráfico 2.10. Tiempo de espera para ser atendido en la cita

Elaborado por: El investigador

Análisis e Interpretación: Según los datos presentados en el gráfico 2.10, el 40% de los pacientes esperan menos de 10 minutos para ser atendidos, el 23.3% de los pacientes esperan de 10 a 20 minutos para ser atendidos, el 20% de los pacientes esperan de 20 a 30 minutos para ser atendidos y el 16.7% esperan más de una hora. En conclusión, se evidencia que la mayoría de pacientes no debe esperar demasiado tiempo para ser atendido.

Pregunta 11: ¿Considera usted que se podría agilizar el proceso de asignar turnos del centro odontológico?

INDICADORES	FRECUENCIA	PORCENTAJE
Si	24	80,0%
No	6	20,0%
Total	30	100%

Tabla 2.14. Tabulación de resultados de la pregunta 11

Elaborado por: El investigador

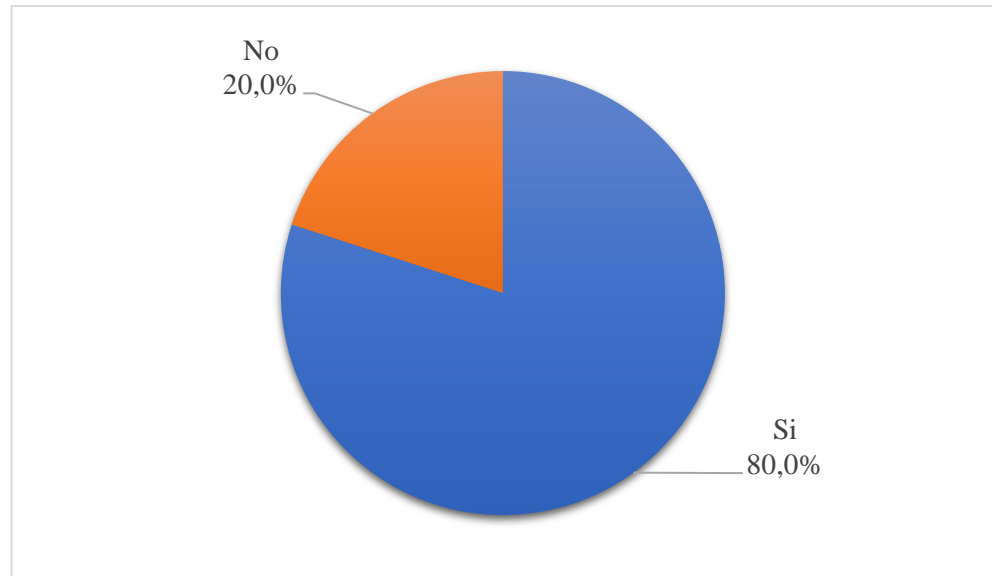


Gráfico 2.11. ¿Considera usted que se podría agilizar el proceso de asignar turnos del centro odontológico?

Elaborado por: El investigador

Análisis e Interpretación: Según los datos presentados en el gráfico 2.11, el 80% de los pacientes encuestados considera que se podría agilizar el proceso de asignar turnos del centro odontológico a diferencia del 20% que opinan lo contrario. En conclusión, se evidencia que se podría agilizar el proceso de asignar turnos del centro odontológico.

Pregunta 12: ¿Considera positivo que un sistema informático mejore el proceso de agendamiento de citas?

INDICADORES	FRECUENCIA	PORCENTAJE
Totalmente de acuerdo	13	43,3%
De acuerdo	12	40,0%
Ni de acuerdo / Ni en desacuerdo	5	16,7%
En desacuerdo	0	0,0%
Totalmente en desacuerdo	0	0,0%
Total	30	100%

Tabla 2.15. Tabulación de resultados de la pregunta 12

Elaborado por: El investigador

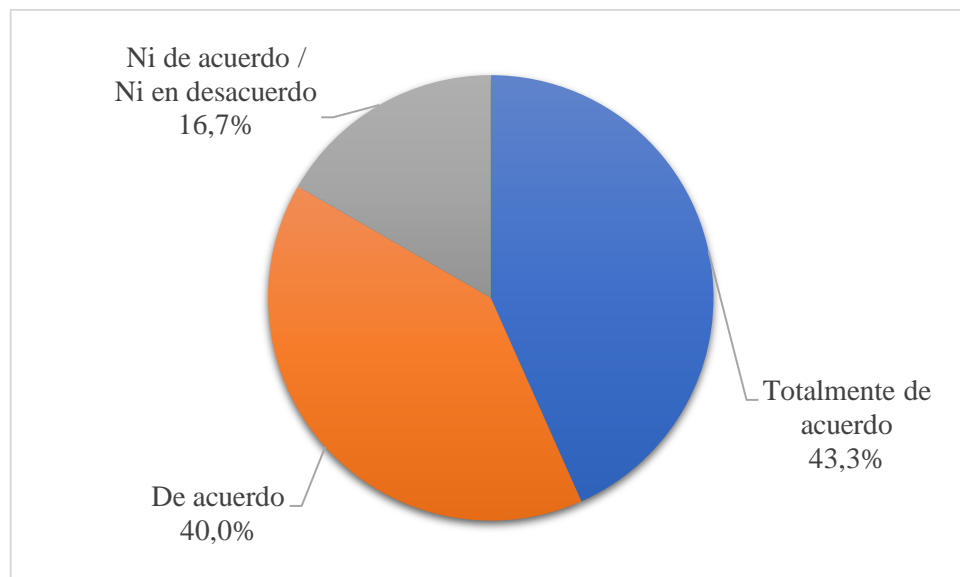


Gráfico 2.12. ¿Considera positivo que un sistema informático mejore el proceso de agendamiento de citas?

Elaborado por: El investigador

Análisis e Interpretación: Según los datos presentados en el gráfico 2.12, el 43.3% de los pacientes encuestados están totalmente de acuerdo que un sistema informático mejore el proceso de agendamiento de citas, el 40% de los pacientes encuestados están de acuerdo que un sistema informático mejore el proceso de agendamiento de citas y el 16.7% de los pacientes encuestados no están de acuerdo ni en desacuerdo que un sistema informático mejore el proceso de agendamiento de citas. Se evidencia que la mayor parte de los pacientes encuestados están de acuerdo que un sistema informático mejore el proceso de agendamiento de citas en el centro odontológico “DENTALSIS”.

Resultados de la entrevista

<p>DATOS PERSONALES Apellidos y Nombres: Toapanta Chávez Magda Pamela Institución: Centro Odontológico “DENTALSIS” Cargo: Odontólogo</p>			
<p>Objetivo: Recopilar información acerca del proceso de agendamiento de citas en el Centro Odontológico “DENTALSIS”.</p>			
Nº	Pregunta	Respuesta	Observación
1	¿Qué sector de la población por lo general acude a reservar una cita?	El sector de la población que acude a reservar las citas son personas adultas entre los 25 años en adelante, además un pequeño porcentaje de personas de la tercera edad.	Para los tratamientos de odontopediatría la reserva de citas lo realiza los padres o representantes del menor.
2	¿Qué información del paciente es necesaria para su registro?	La información del paciente que es necesaria para su registro consta en: <ul style="list-style-type: none"> • Cédula • Nombres y Apellidos completos • Número de teléfono • Sector en el que vive (dirección) 	En la atención del paciente se pregunta acerca de antecedentes médicos que sean esenciales para el odontólogo.

3	¿Cómo es el proceso actual de agendamiento de citas?	<p>Para el agendamiento de citas existe tres modos, los cuales son:</p> <ul style="list-style-type: none"> • Vía WhatsApp: El paciente escribe un mensaje al número del consultorio solicitando una cita. • Vía llamada telefónica: El paciente realiza una llamada al número del consultorio solicitando una cita. • Personalmente: El paciente se acerca al consultorio a solicitar una cita. <p>Luego se verifica la disponibilidad de los horarios y se agenda.</p>	Entre los modos de agendamiento el más solicitado es mediante WhatsApp ya que los clientes no deben acercarse de modo físico al consultorio, cuando la consulta de un paciente ya finalizo puede agendar una próxima cita.
4	¿Cuánto tiempo aproximadamente se demora en el agendamiento de citas?	El tiempo aproximado es de 5 a 10 minutos ya que solo se pregunta el nombre y número de celular. Al momento que se encuentre en el consultorio se llena la historia con el paciente.	
5	¿Cuáles son los inconvenientes que se han presentado al momento del agendamiento de citas?	El inconveniente que se ha presentado ha sido el cruce de horarios con diferentes pacientes a causa de la herramienta del calendario de Google.	

6	Utiliza alguna aplicación para realizar el agendamiento de citas	La aplicación que se utiliza es el calendario de Google por lo que envía una notificación con 30 minutos previos a la cita agendada.	
7	¿Qué problema ocasiona este tipo de proceso manual?	El problema que se tiene a menudo es cuando la aplicación no se sincroniza con las citas ya agendadas y existen un duplicado de turnos por varias personas o aparecen citas fantasmas.	Las citas fantasmas afectan al centro odontológico ya que en el horario que se encuentre no se brindara atención a ningún paciente.
8	¿Qué persona se encarga del agendamiento y registro de pacientes?	Quien se encarga de agendamiento y registro de pacientes es mi persona, soy encargada de toda la gestión del centro odontológico.	Actualmente, la gestión la realiza una sola persona lo cual puede disminuir el número de citas, ya que el mismo odontólogo debe realizar varios cargos en el consultorio.
9	¿Quiénes tienen acceso a la información?	Solo mi persona.	Existe probabilidades de robo de información ya que no se implementa medidas de seguridad para perdidas/extracción de información o ingreso de terceras personas al consultorio.

10	¿Cuál es su opinión acerca de implementar una solución informática que ayude en la gestión de las actividades que se realizan en el centro odontológico?	Sería conveniente y práctico contar con una solución informática que me ayude en la gestión de las actividades que se realizan en el centro odontológico.	Es apropiado implementar una solución informática debido a que el registro de pacientes y agendamiento de citas utiliza un excesivo lapso de tiempo, además así se optimizara los procesos del centro odontológico.
<p>Conclusión: De acuerdo con la entrevista aplicada se puede determinar que al implementar una solución informática para el agendamiento de citas sería un aporte muy significativo ya que optimizaría el tiempo en los procesos de reservar una cita y registrar a los pacientes, además presentara la información de una manera ordenada y se evitaría perdidas de datos o duplicar citas en los mismos horarios.</p>			

Tabla 2.16. Resultados de la entrevista

Elaborado por: El investigador

2.2.4. Procesamiento y análisis de datos

En base a la información obtenida a través de instrumentos de recolección aplicadas al Centro Odontológico “DENTALSIS” se determina que:

- La mayoría de pacientes acuden al centro odontológico solo cuando presentan molestias y han sido atendidos durante 6 meses a un año.
- La documentación de los pacientes es manejada de forma física y con la aplicación que se está utilizando genera problemas y conflictos en la reserva de citas.
- Los pacientes y persona del consultorio están de acuerdo que el proceso actual para el agendamiento de citas no es el más optimo ya que se lo realiza de forma manual y no garantiza la disminución de tiempo en el registro de pacientes.
- Según los resultados obtenidos en las encuestas se concluye que el tiempo de espera para la atención es menor a 10 minutos y no existen dificultades al reservar una cita, lo cual genera un nivel alto de satisfacción de parte de los pacientes.
- Según los resultados del modo que se realiza una reserva de cita el más conveniente para los pacientes es vía telefónica en especial por medio del WhatsApp ya que es una forma sencilla, rápida y eficiente.

Capítulo III.- RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de los resultados

3.1.1. Proceso actual del agendamiento de citas

Después que se adquirió la información acerca del proceso de agendamiento de citas se realizó un análisis para elaborar el diagrama de procesos que se muestra a continuación:

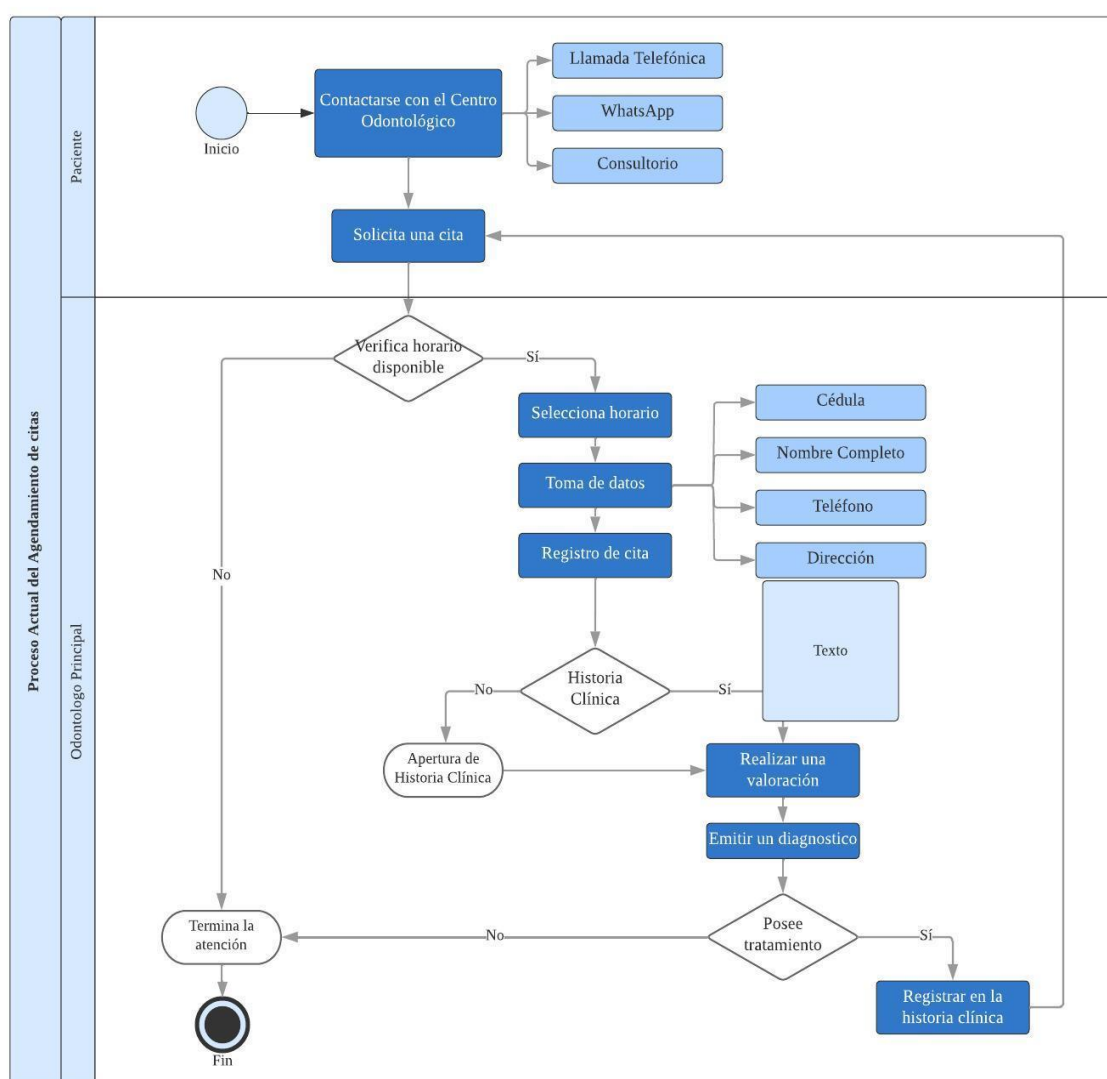


Gráfico 3.1. Proceso actual del agendamiento de citas

Elaborado por: El investigador

En el gráfico 3.1, se puede evidenciar el proceso actual del agendamiento de citas en el que consta las fases que se detallan a continuación:

1. El paciente se pone en contacto con el centro odontológico a través de llamada telefónica o enviando un mensaje a través de la aplicación WhatsApp o personalmente al consultorio.
2. Se solicita una cita al odontólogo.
3. La persona encargada verifica el horario disponible, si existe horarios disponibles se procede con el agendamiento de citas, de lo contrario termina la atención.
4. El paciente selecciona un horario disponible, se le toma los datos básicos como cédula, nombre completo, teléfono y dirección, se registra la cita.
5. En el momento que se acerca al consultorio se verifica si dispone de una historia clínica.
 - 5.1. Si el paciente posee una historia clínica, se verifica que los datos se encuentren actualizados.
 - 5.2. Si el paciente no posee una historia clínica, se procede a la creación de una llenando con la información solicitada por el odontólogo.
6. Luego de poseer una historia clínica, el odontólogo realiza una valoración y emite un diagnóstico al paciente.
7. Si el paciente en el diagnóstico consta que debe realizar un tratamiento se registra en la historia clínica y se agenda una cita nuevamente, caso contrario se termina la atención.

3.1.2. Proceso de Agendamiento de citas optimizado

El proceso de agendamiento convencional exponía ciertos problemas al momento de verificar los horarios disponibles, en el registro de los pacientes y tratamientos, ya que se los efectuaba de forma manual. Por lo tanto, se aplicaron mejoras en el proceso para optimizar tiempo y presentar de forma ordenada la información.

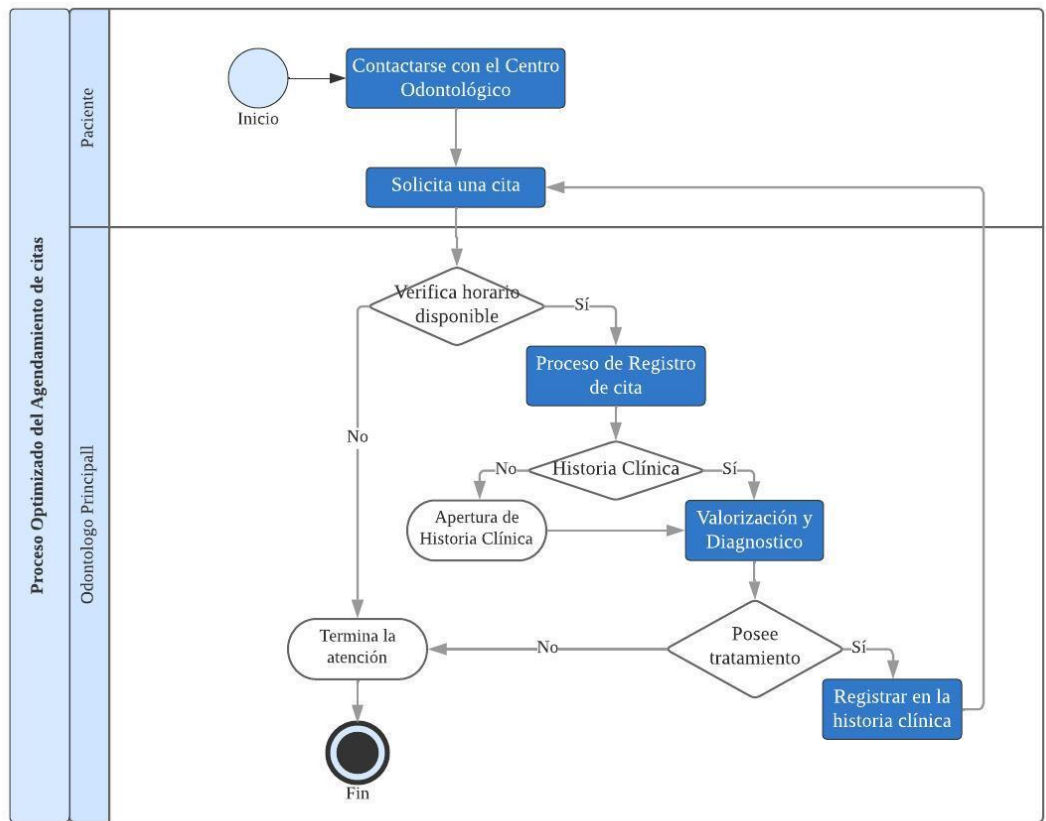


Gráfico 3.2. Proceso de Agendamiento de citas optimizado

Elaborado por: El investigador

En el gráfico 3.2, se muestra que se efectuaron cambios en el proceso convencional, haciendo hincapié en el registro de citas en los horarios disponibles y en las historias clínicas de los pacientes.

Una de las modificaciones que se realizó fue que para contactarse con el centro odontológico la mayor parte de los pacientes lo realiza mediante WhatsApp, lo cual nos proporcionara información básica de los pacientes para registrar una cita. Además, el paciente en cada consulta va a contar con un diagnóstico el cual facilitara el agendamiento de citas futuras.

3.1.3. Análisis de la Tecnología Responsive Web Design (RWD)

Las tecnologías Responsive Web Design (RWD) y Adaptive Web Design (AWD) son enfoques para crear sitios web que comparten el mismo objetivo, que es garantizar que los usuarios tengan una experiencia positiva al usar los sitios en cualquier dispositivo que elijan. Para analizar las ventajas, desventajas y características entre estas tecnologías se procede a realizar la siguiente tabla comparativa.

Características	RWD	AWD
Enfoque	Proporciona un diseño web que se ajusta a diferentes tamaños de pantalla	Proporciona múltiples versiones del sitio web adaptadas a dispositivos específicos o grupos de dispositivos
Adaptación del diseño	Se adapta al tamaño de la pantalla.	Diseños personalizados para dispositivos específicos.
Imágenes	Utiliza imágenes flexibles que se ajustan al tamaño de la pantalla	Requiere imágenes específicas para cada versión del sitio web
Adaptación del contenido	Misma versión de contenido para todos los dispositivos.	Contenido personalizado para diferentes dispositivos.
Complejidad de desarrollo	Menor complejidad, un único diseño.	Mayor complejidad, múltiples versiones y redireccionamiento.
Mantenimiento	Menos mantenimiento, actualizaciones centralizadas.	Mayor mantenimiento, actualizaciones individuales para cada versión.
Experiencia del usuario	Buena experiencia general en diferentes dispositivos.	Experiencia altamente optimizada para cada dispositivo.
Flexibilidad	Menor flexibilidad, cambios pueden afectar a todos los dispositivos.	Mayor flexibilidad, cambios específicos para cada versión.

Requisitos de detección	Depende de consultas de medios y CSS.	Detección de características del dispositivo y contexto del usuario.
Costos	Su implementación es menos costosa porque existen múltiples plantillas en las principales plataformas web.	Es más costoso porque cada pantalla se personaliza tanto en la creación como en el mantenimiento.
Eficiencia de carga	Mayor carga de contenido en dispositivos móviles.	Carga de contenido optimizado para cada dispositivo.
SEO	Mejora el SEO debido a una sola URL para todas las versiones del sitio web	No mejora el SEO debido a múltiples URL para cada versión del sitio web
Ejemplos de implementación	Sitios web que se ajustan a diferentes tamaños de pantalla.	Aplicaciones web que ofrecen versiones personalizadas para diferentes dispositivos.

Tabla 3.1. Tabla comparativa entre Responsive Web Design (RWD) y Adaptive Web Design (AWD)

Elaborado por: El investigador

De acuerdo con la tabla 3.1, se puede concluir que la Tecnología Responsive Web Design es una técnica muy utilizada en la actualidad debido a que proporciona múltiples ventajas significativas, tales como mejorara la experiencia de usuario, ajustar el diseño web a diferentes tamaños de pantalla, utilizar una única versión de contenido e imágenes para todos los dispositivos, reducir el nivel de complejidad en el desarrollo, disminución de costos y mantenimiento. Estas razones hacen que sea una elección sólida para crear sitios web modernos y eficaces.

3.1.4. Tecnologías Front-end

Las tecnologías front-end desempeñan un papel crucial en el desarrollo de aplicaciones con Responsive Web Design (Diseño Web Responsivo). Estas tecnologías se centran en crear la interfaz de usuario y gestionar la interacción del usuario con el sitio web en diferentes dispositivos y tamaños de pantalla. Algunas de las tecnologías más comunes que permiten la adaptabilidad y la interactividad son las siguientes:

Tecnología	Descripción	Ventajas	Desventajas
HTML5	La última versión del lenguaje de marcado HTML, que incluye nuevas características y elementos semánticos	Estructura clara y semántica	Puede requerir conocimientos adicionales para aprovechar todas las características de HTML5
CSS3	La última versión del lenguaje de hojas de estilo CSS, que incluye nuevas características y capacidades	Flexibilidad en la presentación y diseño	Puede haber problemas de compatibilidad con navegadores más antiguos
JavaScript	Lenguaje de programación utilizado para agregar interactividad y dinamismo a las páginas web	Permite la creación de aplicaciones web altamente interactivas	Puede requerir conocimientos de programación más avanzados
Media Queries	Característica de CSS que permite aplicar estilos específicos en función de características del dispositivo	Permite adaptar la apariencia y el diseño a diferentes pantallas	Requiere escribir y gestionar múltiples consultas de medios

Grid Systems	Sistemas de cuadrícula CSS como Bootstrap o Foundation que facilitan la creación de diseños responsivos	Ayuda a mantener un diseño coherente y ordenado	Puede limitar la flexibilidad y personalización del diseño si se depende demasiado del sistema
Frameworks CSS	Frameworks como Bootstrap, Foundation o Bulma que proporcionan estilos y componentes predefinidos para RWD	Aceleran el desarrollo y proporcionan un enfoque consistente	Pueden requerir un aprendizaje adicional y limitar la flexibilidad de diseño según los estilos predefinidos

Tabla 3.2. Tabla comparativa entre las tecnologías utilizadas en el desarrollo de aplicaciones RWD

Elaborado por: El investigador

De acuerdo con la tabla 3.2, los Frameworks CSS proporcionan una variedad de beneficios como una serie de estilos, componentes, sistemas de cuadrícula y media queries predefinidos que permiten acelerar el proceso de desarrollo y facilitan la creación de diseños responsivos. Además, los frameworks CSS tienen comunidades activas de desarrolladores y usuarios que proporcionan documentación, recursos y soporte.

3.1.5. Frameworks de Diseño

Los frameworks de diseño son herramientas ampliamente utilizados en el desarrollo de aplicaciones Responsive Web Design, los cuales proporcionan una base sólida, eficiencia en el desarrollo, consistencia visual, adaptabilidad y compatibilidad entre navegadores, lo que contribuye a crear interfaces de usuario responsivos y atractivas en diferentes dispositivos. En la siguiente tabla comparativa se realiza un análisis de ventajas, desventajas y algunas características esenciales de estos frameworks.

Framework	Creador	Año de Lanzamiento	Descripción	Ventajas	Desventajas
Bootstrap	Twitter, Mark Otto y Jacob Thornton	2011	Framework de diseño front-end que ofrece una estructura y componentes RWD.	Amplia documentación y comunidad de usuarios.	Puede generar sitios web con una apariencia similar debido a su popularidad.
Foundation	Empresa ZURB	2011	Framework CSS y JS que se centra en la creación de diseños responsivos.	Personalizable y flexible.	Requiere más conocimientos técnicos para aprovechar todas sus características.
Bulma	Jeremy Thomas	2016	Framework CSS ligero y moderno que proporciona una base para sitios RWD.	Fácil de aprender y personalizar.	Menos funciones y opciones en comparación con Bootstrap y Foundation.
Tailwind CSS	Adam Wathan, Jonathan Reinink, David Hemphill y Steve Schoger	2017	Framework CSS utilitario que se basa en clases predefinidas para facilitar la construcción de interfaces.	Ofrece una gran flexibilidad y permite un desarrollo rápido.	Requiere más tiempo y esfuerzo para personalizar y mantener los estilos debido a la necesidad de trabajar con muchas clases.
Semantic UI	Jack Lukic	2013	Framework CSS y JS que se enfoca en la claridad y la semántica del código	Sintaxis clara y comprensible	Menos popularidad y comunidad en comparación con otros frameworks
Material-UI	Equipo de desarrolladores liderado por Olivier	2014	Implementación de los principios de diseño Material Design de Google, con componentes predefinidos y estilos.	Ofrece una gran cantidad de componentes de diseño.	Requiere más conocimientos para personalizar y adaptar los estilos a las necesidades específicas del proyecto.

	Tassinari y Matt Brookes				
UIKit	Equipo de desarrollo de YOOtheme	2013	Framework CSS y JS que ofrece componentes y estilos modernos y responsivos	Diseño limpio y moderno	Menos documentación y recursos en comparación con otros frameworks

Tabla 3.3. Tabla comparativa entre los frameworks utilizados en el desarrollo de aplicaciones RWD

Elaborado por: El investigador

Conforme con la tabla 3.3, se optó que el mejor es Bootstrap debido a que es uno de los más reconocido y usado en la tecnología responsive web design, ya que proporciona un sistema de cuadrícula que permite organizar y estructurar el contenido de una manera flexible, asimismo brinda una extensa gama de estilos, temas y componentes HTML y CSS, además es compatible con los principales navegadores web y posee una gran comunidad de desarrolladores.

3.1.6. Frameworks Backend

Los frameworks Backend son herramientas esenciales en el desarrollo de aplicaciones web, ya que se enfocan en la parte del servidor y la lógica de negocio. Estos frameworks proporcionan una estructura organizada, funcionalidades predefinidas, mayor seguridad y una comunidad de soporte, lo que ayuda a los desarrolladores a crear aplicaciones de manera eficiente, segura y escalable. En la siguiente tabla comparativa se muestra algunas de las características importantes de los Frameworks Backend ASP.NET, Laravel, Angular, Express.js, Ruby on Rails y Django.

Características	ASP.NET	Laravel	Angular	Express.js	Ruby on Rails	Django
Lenguaje	C#	PHP	TypeScript	Javascript	Ruby	Python
RWD	Si	Si	Si	Si	Si	Si
Licencia	MS-PL	MIT	MIT	MIT	MIT	BSD.
Mecanismos de Seguridad	<ul style="list-style-type: none"> • Autenticación y autorización. • ASP.NET Identity. • Cross-Site Scripting (XSS) y SQL Injection. • Configuración de seguridad. • Validación de formularios. • Encriptación y protección de datos. 	<ul style="list-style-type: none"> • Autenticación y autorización. • Protección contra Cross-Site Scripting (XSS). • Prevención de ataques de inyección SQL. • Encriptación de datos. • Protección contra falsificación de solicitudes entre sitios (CSRF). • Gestión de errores segura. • Configuración de seguridad 	<ul style="list-style-type: none"> • Autenticación y autorización. • Guards • Sanitización de datos. • Protección contra ataques de falsificación de solicitudes entre sitios (CSRF) • Configuración de encabezados de seguridad. • Validación de formularios 	<ul style="list-style-type: none"> • Middleware de seguridad. • Validación de datos de entrada. • Gestión de autenticación y autorización. • Configuración de encabezados de seguridad. • Protección de sesión. • Prevención de ataques de fuerza bruta 	<ul style="list-style-type: none"> • Protección contra ataques de inyección SQL. • Protección contra Cross-Site Scripting (XSS). • Protección contra ataques CSRF. • Validación de datos de entrada. • Control de acceso y autorización. • Configuración de encabezados de seguridad. 	<ul style="list-style-type: none"> • Protección contra ataques de inyección SQL. • Protección contra Cross-Site Scripting (XSS). • Protección contra ataques CSRF. • Autenticación y autorización. • Configuración de encabezados de seguridad. • Validación de datos de entrada
Nivel de Popularidad	Alta	Alta	Alta	Alta	Media	Alta

Curva de Aprendizaje	Medio	Leve	Medio	Leve	Alta	Leve
Escalable	Si	Si	Si	Si	Si	Si
Compatibilidad con Base de Datos	<ul style="list-style-type: none"> • Microsoft SQL Server • MySQL • PostgreSQL • Oracle • MongoDB 	<ul style="list-style-type: none"> • MySQL • PostgreSQL • SQLite • SQL Server • MongoDB 	Utiliza Wen API para consumir datos.	<ul style="list-style-type: none"> • MongoDB • MySQL • PostgreSQL • SQLite 	<ul style="list-style-type: none"> • MySQL • PostgreSQL • SQLite • SQL Server • MongoDB 	<ul style="list-style-type: none"> • MySQL • PostgreSQL • SQLite • Oracle • MongoDB
ORM	<ul style="list-style-type: none"> • Entity Framework • LINQ to SQL 	Eloquent ORM	No aplica	No aplica	ActiveRecord	Django ORM

Tabla 3.4. Tabla comparativa entre Frameworks Backend para el desarrollo de aplicaciones Responsive Web Design (RWD)

Elaborado por: El investigador

Conforme con la tabla 3.4, se escogió que para el framework Backend ASP.NET presenta mejores características en el desarrollo de aplicaciones Responsive Web Design, ya que se puede implementar la arquitectura en capas, tiene una leve curva de aprendizaje, provee múltiples mecanismos de seguridad, es compatible con bases de datos relacionales y no relacionales, también posee dos tecnologías de mapeo objeto-relacional para simplificar el acceso y manipulación de datos en aplicaciones orientadas a objetos.

3.1.7. Gestores de base de datos

Los gestores de bases de datos son software diseñados para gestionar y organizar grandes volúmenes de datos de manera eficiente. A continuación, se menciona algunas características de los gestores de bases de datos compatibles con el Frameworks Backend.

Características	Microsoft SQL Server	MySQL	MongoDB
Tamaño de la base de datos	524,272 terabytes (TB)	No tiene limite	No tiene limite
Licencia	Propietaria	Dual (Propietaria y Open Source)	Open Source
Soporte para SQL	Sí	Sí	No (utiliza un lenguaje de consulta propio llamado BSON)
Transacciones ACID	Sí	Sí	Sí
Escalabilidad	Buena	Buena	Excelente
Alta disponibilidad	Sí	Sí	Sí
Integración con .NET	Excelente	Excelente	Limitada

Tabla 3.5. Tabla comparativa entre los gestores de bases de datos

Elaborado por: El investigador

De acuerdo con la tabla 3.5, se optó por el gestor de datos MySQL, ya que no posee un límite específico para el tamaño máximo de la base de datos, soporta el lenguaje SQL, su licencia es Open Source, es escalable y tiene una excelente integración con .NET.

3.1.8. Metodologías de Desarrollo Web

Las metodologías de desarrollo web son enfoques estructurados y organizados que proporcionan directrices y mejores prácticas para la construcción exitosa de sitios web y aplicaciones. Estas metodologías tienen como objetivo asegurar una planificación eficiente, una colaboración efectiva y una entrega exitosa, permitiendo adaptabilidad, calidad y satisfacción del cliente. A continuación, se presentan algunas de las metodologías más utilizadas en el desarrollo web:

Características	Relationship Management Methodology (RMM)	Object-Oriented Hypermedia Design Methodology (OOHDM)	Extreme Programming (XP)	Rational Unified Process (RUP)
Enfoque	Gestión de relaciones cliente-proyecto	Orientado a hipertexto	Ágil	Basada en procesos
Etapas	<ul style="list-style-type: none"> • Inicio y análisis. • Definición de requerimientos. • Planificación y diseño. • Desarrollo y entrega iterativa. • Retroalimentación y adaptación. • Cierre y evaluación 	<ul style="list-style-type: none"> • Análisis de requerimientos • Diseño conceptual • Diseño navegacional • Diseño de presentación • Implementación • Pruebas y evaluación • Mantenimiento 	<ul style="list-style-type: none"> • Planeación • Diseño • Codificación • Pruebas • Retroalimentación • Integración continua • Despliegue 	<ul style="list-style-type: none"> • Inicio • Elaboración • Transición • Construcción
Tamaño de Equipo	Pequeño a grande	Pequeño a grande	Pequeño a mediano	Pequeño a grande

Alcance del proyecto	Aplicable a cualquier tamaño	Aplicable a cualquier tamaño	Proyectos pequeños / medianos	Proyectos medianos / grandes
Iterativo	No se basa en un enfoque iterativo.	Se basa en un enfoque iterativo y evolutivo.	Se basa en un enfoque iterativo e incremental.	Se basa en un enfoque iterativo e incremental.
Ciclo de vida	No está especificado un ciclo de vida	Se siguen ciclos iterativos de diseño, implementación y evaluación	No está especificado un ciclo de vida	Fases específicas y bien definidas.
Entorno de desarrollo	No especificado	No especificado	No especificado	No especificado
Roles	<ul style="list-style-type: none"> • Gerente de proyecto • Cliente • Equipo de desarrollo • Analista de Negocios • Diseñador de Experiencia de Usuario (UX Designer) • Experto en Relaciones Cliente-Proyecto 	<ul style="list-style-type: none"> • Analista de requerimientos • Diseñador conceptual • Diseñador navegacional • Diseñador de presentación • Desarrolladores • Probadores • Gerente de proyecto 	<ul style="list-style-type: none"> • Cliente • Programador • Tester • Tracker • Coach de X 	<ul style="list-style-type: none"> • Gerente de Proyecto (Project Manager) • Analista de Negocios (Business Analyst) • Arquitecto (Architect) • Analista (Analyst) • Diseñador (Designer) • Desarrollador (Developer) • Tester (Tester) • Usuario Final (End User)
Documentación	Documentación moderada a pesada para garantizar una buena gestión de las relaciones cliente-proyecto.	Documentación moderada a pesada para la construcción de sistemas hipermediales.	Documentación ligera y enfocada en las necesidades del proyecto.	Documentación pesada y detallada para cubrir todas las etapas del proceso de desarrollo

Adaptación al cambio	Adaptabilidad dependiente de las relaciones cliente-proyecto	Flexibilidad para la adaptación al cambio	Alta adaptabilidad y flexibilidad	Adaptación al cambio a través de iteraciones
Comunicación	Fuerte enfoque en la comunicación con el cliente	Comunicación frecuente y colaborativa	Fuerte énfasis en la comunicación y colaboración	Comunicación estructurada y documentada
Flexibilidad	Alta	Moderada	Alta	Moderada
Adaptabilidad	Alta	Moderada	Alta	Alta

Tabla 3.6. Tabla comparativa entre las Metodologías de Desarrollo Web

Elaborado por: El investigado

De acuerdo con la tabla 3.6 se eligió la metodología XP (Extreme Programming) dado que es una metodología ágil que se centra en la adaptabilidad y la capacidad de respuesta a los cambios, permite a los equipos de desarrollo ajustar los requisitos y las funcionalidades a medida que avanzan en el proyecto, promueve la comunicación estrecha y continua entre los miembros del equipo de desarrollo, así como con el cliente. Además, utiliza ciclos cortos de desarrollo, conocidos como iteraciones o sprints, en los que se planifica, se desarrolla, se prueba y se entrega una funcionalidad incrementada del software, es por eso que Extreme Programming garantizan la entrega de un proyecto de alta calidad.

3.2. Desarrollo de la propuesta

La metodología elegida tiene como objetivo principal desarrollar un producto que cumpla con los requisitos específicos del centro odontológico "DENTALSIS". Esta metodología se divide en cinco etapas fundamentales: planificación, diseño, codificación, pruebas e implantación. A continuación, se detalla cada una de estas fases:

3.2.1. Fase I: Planificación

3.2.1.1. Levantamiento de la información

Para adquirir los datos necesarios se utilizaron dos técnicas de recolección de información, las cuales fueron: entrevista destinada al odontólogo y una encuesta realizada a los pacientes del Centro Odontológico "DENTALSIS". Mediante estas técnicas se han definido realizar una tabla de los requisitos funcionales con el objetivo de documentar de manera clara y organizada los distintos aspectos funcionales que debe tener la aplicación.

3.2.1.2. Requisitos Funcionales del Sistema

De acuerdo con la información recopilada se identificó los requisitos funcionales que se detallan a continuación:

Requisito	Descripción	Prioridad
RF001	Control de Acceso	Alta
RF002	Gestión de Pacientes	Alta
RF003	Gestión de Odontólogos	Alta
RF004	Gestión Horario	Alta
RF005	Gestión de Reserva de Citas	Alta
RF006	Gestión de la atención de citas	Alta
RF007	Comprobante	Alta
RF008	Cerrar Sesión:	Media

Tabla 3.7. Requisitos Funcionales del Sistema

Elaborado por: El investigador

A continuación, se proporciona una descripción más detallada de cada uno de los requisitos funcionales del sistema:

- **Control de Acceso:** El odontólogo será capaz de registrarse, para ingresar a los módulos se debe introducir su usuario y contraseña.
- **Gestión de Pacientes:** Se presentará un listado de los pacientes registrados en el centro odontológico, de los cuales se podrán modificar o eliminar, además de agregar más pacientes.
- **Gestión de Odontólogos:** Se presentará un listado de los odontólogos registrados, de los cuales se podrán modificar o eliminar, además de agregar nuevos.
- **Gestión Horario:** se visualizará los horarios disponibles digitando la cedula de un odontólogo ya registrado, además valdrá editar y eliminar.
- **Gestión de Reserva de Citas:** Se solicitará el número de cedula del paciente y la fecha en la cual se desea realizar el agendamiento de la cita.
- **Gestión de la atención de citas:** Se mostrará una lista de las citas pendientes de atención, las cuales se las puede atender o cancelar.
- **Comprobante:** Se presenta una interfaz para la generación de un comprobante de pago de la consulta en la cual tiene datos del paciente como el valor total de la consulta.
- **Cerrar Sesión:** El usuario podrá finalizar la conexión con el sistema para prevenir el retorno a paginas anteriores.

3.2.1.3. Roles Asignados en el proyecto

En la metodología XP se asignan diferentes roles a los miembros del equipo para facilitar la colaboración, para lo cual se eligieron únicamente los que se adaptan al desarrollo del proyecto.

Nombre	Rol	Función
Valeria Logroño	Programadora	Realiza el análisis, la planificación, diseño, desarrollo y codificación, pruebas funcionales para la entrega del sistema, adicional el manual de usuario.
Ing. Franklin Mayorga	Entrenador - Rastreador	Supervisa continuamente las actividades y avances del proyecto con el propósito de entregar un sistema web de calidad, aporta alternativas más favorables basadas en su experiencia.
Dra. Pamela Toapanta	Cliente - Tester	Responsable que define las especificaciones ya que conoce las reglas del negocio y los procesos que se van a automatizar, realiza las pruebas del sistema web.

Tabla 3.8. Definición de Roles XP

Elaborado por: El investigador

3.2.1.4. Arquitectura de la aplicación

La arquitectura de la aplicación se forma de dos frameworks, para el back-end se empleará Asp.Net y para tener un buen control en el Front-end se usará Bootstrap. Para la gestión de base de datos se utilizará MySQL.

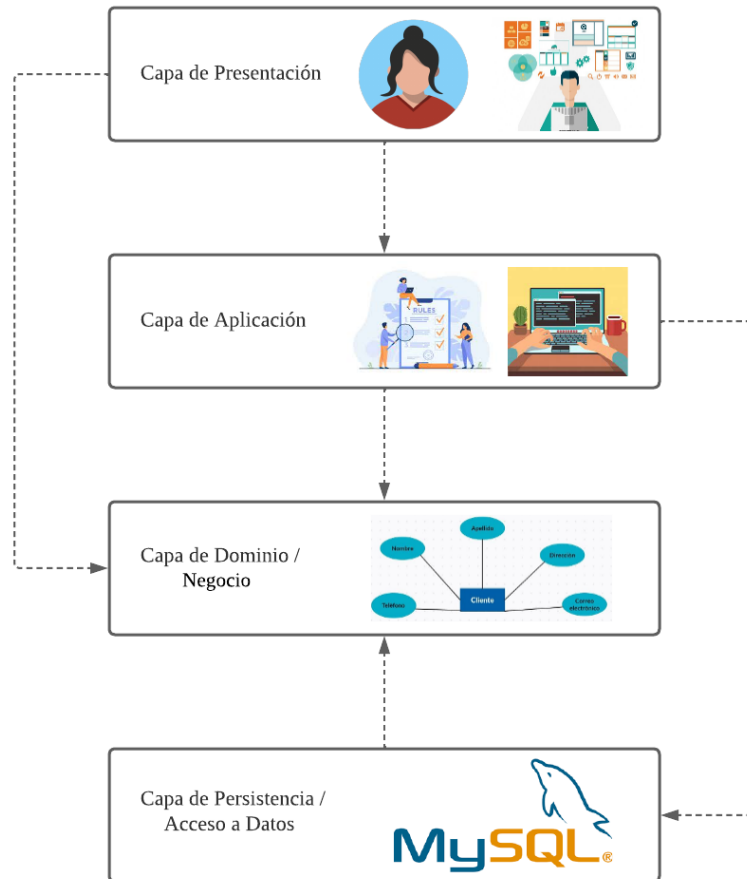


Gráfico 3.3. Arquitectura del Sistema

Elaborado por: El investigador

Niveles de la arquitectura N-Capas - Estratificación Flexible

- **Capa de Presentación:** Representa a las interfaces que serán visualizadas por el usuario final.
- **Capa de Aplicación:** Se encarga recibir y procesar las solicitudes del usuario.
- **Capa de Dominio/Negocio:** Implementa la funcionalidad principal de la aplicación, procesar y manipular los datos de acuerdo con las reglas y requisitos del sistema.
- **Capa de Persistencia/Acceso a Datos:** Representa a la base de datos, en donde se va almacenar la información de los pacientes y odontólogos.

3.2.1.5. Requerimientos de Software

Las herramientas para la elaboración del sistema se detallan a continuación:

- MySQL Workbench: Es una herramienta de administración y diseño de bases de datos con interfaz gráfica de usuario intuitiva y completa para el sistema de gestión de bases de datos MySQL.
- Visual Studio: Es un entorno de desarrollo integrado (IDE) con numerosas características que dan respaldo a muchos aspectos del desarrollo de software.
- Bootstrap: Es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web.
- CSS: Son hojas de estilo en cascada, permite controlar el diseño, el estilo y la presentación visual de los elementos HTML en una página web.

3.2.1.6. Historias de Usuario

La elaboración de las historias de usuario será por lado del cliente y el programador, deben ser detalladas en un lenguaje sencillo y fácil de comprender con el propósito de capturar las necesidades del usuario final y priorizar los requisitos y funcionalidades de la aplicación web.

El modelo para las historias de usuarios que se usaran tendrá el siguiente diseño:

Historia de Usuario	
Número:	Usuario:
Nombre Historia:	
Prioridad en Negocio:	Riesgo en Desarrollo:
Puntos Estimados:	Iteración Asignada:
Programador Responsable:	
Descripción:	
Observación:	

Tabla 3.9. Modelo de historia de usuario

Elaborado por: El investigador

A continuación, se detalla cada una de las especificaciones que posee las historias de usuario:

- Número: Identificador establecido para cada historia de usuario.
- Usuario: Individuo responsable de la historia de usuario.
- Nombre Historia: Determinación del título de la historia de usuario.
- Prioridad en Negocio: Medidor del nivel de necesidad del usuario (Alta, Media, Baja).
- Riesgo en Desarrollo: Medidor del nivel de riesgo que representa (Alta, Media, Baja).
- Puntos Estimados: Lapso de tiempo (días) que fue implementado en el desarrollo de la actividad.
- Iteración Asignada: Iteración designada por el desarrollo o a la fase perteneciente.
- Programador Responsable: Persona responsable del desarrollo de la historia de usuario.
- Descripción: Comentario corto del cliente acerca de la actividad de la historia de usuario.
- Observación: Notas extras acerca de aclaraciones de procesos de la historia de usuario.

Para la elaboración del proyecto se definieron las siguientes historias de usuario:

Historia de Usuario	
Número: 1	Usuario: Cliente
Nombre Historia: Ingreso al sistema	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 4	Iteración Asignada: 1
Programador Responsable: Valeria Logroño	
Descripción: En el ingreso al sistema se requiere el usuario y contraseña al usuario para permitir el acceso, también existe la opción de recuperar contraseña.	
Observación: El campo de usuario corresponderá al correo electrónico del usuario.	

Tabla 3.10. Historia de usuario para el ingreso al sistema

Elaborado por: El investigador

Historia de Usuario	
Número: 2	Usuario: Cliente
Nombre Historia: Recuperar Contraseña	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 5	Iteración Asignada: 1
Programador Responsable: Valeria Logroño	
Descripción: El usuario podrá solicitar la recuperación de la contraseña si es necesario.	
Observación: Para recuperar la contraseña es necesario el correo registrado en la base de datos del usuario	

Tabla 3.11. Historia de usuario para la recuperación de contraseña

Elaborado por: El investigador

Historia de Usuario	
Número: 3	Usuario: Cliente
Nombre Historia: Página Principal	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración Asignada: 1
Programador Responsable: Valeria Logroño	
Descripción: La página principal contendrá la imagen y logo del consultorio, barra lateral con los módulos definidos.	
Observación: Ninguna	

Tabla 3.12. Historia de usuario para la creación de la página principal

Elaborado por: El investigador

Historia de Usuario	
Número: 4	Usuario: Cliente
Nombre Historia: Gestión Pacientes	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 4	Iteración Asignada: 2
Programador Responsable: Valeria Logroño	
Descripción: Creación del módulo, en el cual se mostrará los pacientes registrados. Se podrá crear, modificar o eliminar si es necesario, la búsqueda de pacientes se lo realizara por su número de cedula.	
Observación: La información del registro de pacientes deberá contener todo lo que solicito el cliente.	

Tabla 3.13. Historia de usuario para la gestión pacientes

Elaborado por: El investigador

Historia de Usuario	
Número: 5	Usuario: Cliente
Nombre Historia: Gestión Horarios de Atención	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 4	Iteración Asignada: 2
Programador Responsable: Valeria Logroño	
Descripción: Creación del módulo, en el cual se mostrará los horarios de atención que posee un odontólogo. Se podrá crear, modificar o eliminar si es necesario.	
Observación: La visualización de los horarios de atención dependerá del ingreso previo de la cedula del odontólogo.	

Tabla 3.14. Historia de usuario para la gestión de horarios de atención

Elaborado por: El investigador

Historia de Usuario	
Número: 6	Usuario: Cliente
Nombre Historia: Reserva de Citas	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 4	Iteración Asignada: 2
Programador Responsable: Valeria Logroño	
Descripción: Creación del módulo, en el cual se deberá seleccionar el paciente, la fecha y especialidad que se requiera.	
Observación: Para la fecha se implementará un calendario.	

Tabla 3.15. Historia de usuario para la reserva de citas

Elaborado por: El investigador

Historia de Usuario	
Número: 7	Usuario: Cliente
Nombre Historia: Gestión Atención Pacientes	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Media
Puntos Estimados: 3	Iteración Asignada: 3
Programador Responsable: Valeria Logroño	
Descripción: Creación del módulo, en el cual se mostrará una lista de las citas pendientes de atender con la información del paciente. Se podrá realizar la atención o cancelar.	
Observación: Ninguna.	

Tabla 3.16. Historia de usuario para la gestión de atención de pacientes

Elaborado por: El investigador

Historia de Usuario	
Número: 8	Usuario: Cliente
Nombre Historia: Gestión Odontólogos	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 4	Iteración Asignada: 3
Programador Responsable: Valeria Logroño	
Descripción: Creación del módulo, en el cual se mostrará los odontólogos registrados con sus especialidades y horarios disponibles. Se podrá crear, modificar o eliminar si es necesario, la búsqueda de odontólogos se lo realizara por su número de cedula.	
Observación: La información del registro de odontólogos contendrá los datos básicos.	

Tabla 3.17. Historia de usuario para la gestión de odontólogos

Elaborado por: El investigador

Historia de Usuario	
Número: 9	Usuario: Cliente
Nombre Historia: Comprobante	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración Asignada: 3
Programador Responsable: Valeria Logroño	
Descripción: Creación del módulo Comprobante, en el cual se busca el paciente por cedula e ingresar los tratamientos realizados con su respectivo precio.	
Observación: Los tratamiento y precios serán almacenados en un DataGridView, además se mostrará el total del comprobante.	

Tabla 3.18. Historia de usuario para la creación del Comprobante

Elaborado por: El investigador

Historia de Usuario	
Número: 10	Usuario: Cliente
Nombre Historia: Generar Pdf del Comprobante	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 5	Iteración Asignada: 4
Programador Responsable: Valeria Logroño	
Descripción: Creación de un formato de comprobante acerca de los tratamientos realizados al paciente.	
Observación: Al generar el pdf se debe tomar en cuenta que una copia es para el centro odontológico y otra para el paciente.	

Tabla 3.19. Historia de usuario para generar el pdf del comprobante

Elaborado por: El investigador

Historia de Usuario	
Número: 11	Usuario: Cliente
Nombre Historia: Finalizar Sesión	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Media
Puntos Estimados: 2	Iteración Asignada: 4
Programador Responsable: Valeria Logroño	
Descripción: Creación de una sección para que el usuario pueda finalizar la sesión activa.	
Observación: Para finalizar sesión el usuario deberá dar clic en el nombre de usuario.	

Tabla 3.20. Historia de usuario para finalizar sesión

Elaborado por: El investigador

3.2.1.7. Valoración / Estimación de Historias de Usuario

Una vez que se generaron las historias de usuario, se procedió a calcular el tiempo estimado para cada una de ellas. A continuación, se muestra la distribución del tiempo de ejecución para cada actividad:

- 1 punto representa 1 día de trabajo.
- 1 semana de trabajo comprende 5 días.
- 1 día corresponde a 5 horas diarias.

Nº	Historia de Usuario	Puntos Estimados	Tiempo Estimado		
			Horas	Días	Semanas
1	Ingreso al sistema	4	20	4	0,8
2	Recuperar Contraseña	5	25	5	1,0
3	Página Principal	2	10	2	0,4
4	Gestión Pacientes	4	20	4	0,8
5	Gestión Horarios de Atención	4	20	4	0,8
6	Reserva de Citas	4	20	4	0,8
7	Gestión Atención Pacientes	3	15	3	0,6
8	Gestión Odontólogos	4	20	4	0,8
9	Comprobante	2	10	2	0,4
10	Generar Pdf del Comprobante	5	25	5	1,0
11	Finalizar Sesión	2	10	2	0,4
Tiempo Estimado Total		39	195	39	7,80

Tabla 3.21. Estimación de historias de usuarios

Elaborado por: El investigador

3.2.1.8. Plan de entrega

Luego de realizar la estimación de las historias de usuarios, se diseña un plan de entrega para el desarrollo del proyecto indicando a la iteración que pertenece.

N°	Historia de Usuario	Tiempo Estimado		Iteraciones			
		Horas	Días	1	2	3	4
1	Ingreso al sistema	20	4	x			
2	Recuperar Contraseña	25	5	x			
3	Página Principal	10	2	x			
4	Gestión Pacientes	20	4		x		
5	Gestión Horarios de Atención	20	4		x		
6	Reserva de Citas	20	4		x		
7	Gestión Atención Pacientes	15	3			x	
8	Gestión Odontólogos	20	4			x	
9	Comprobante	10	2			x	
10	Generar Pdf del Comprobante	25	5				x
11	Finalizar Sesión	10	2				x

Tabla 3.22. Plan de entrega

Elaborado por: El investigador

3.2.2. Fase II: Diseño

3.2.2.1. Iteración 1

Conforme con la tabla 3.19, el plan de entrega se desglosa en 3 iteraciones, la primera está conformada por el ingreso al sistema, recuperación de contraseña y página principal.

Código	Historia de usuario	Prioridad	Riesgo
1	Ingreso al sistema	Alta	Alta
2	Recuperar Contraseña	Alta	Alta
3	Página Principal	Alta	Alta

Tabla 3.23. Historias de usuario de la iteración 1

Elaborado por: El investigador

1. Ingreso al sistema

Autoriza el ingreso al usuario solicitando el correo y contraseña.

Código de historia: 1	Responsable: Valeria Logroño
Actividades	
<ul style="list-style-type: none"> • Se comprueba en la base de datos que el usuario posea una contraseña para ingresar al sistema. • Diseño de la interfaz. 	

Tabla 3.24. Actividades de la historia de usuario 1

Elaborado por: El investigador

2. Recuperar Contraseña

El usuario podrá recuperar su contraseña digitando el correo registrado en la base de datos, si el correo no se encuentra registrado se mostrará un mensaje.

Código de historia: 2	Responsable: Valeria Logroño
Actividades	
<ul style="list-style-type: none"> • Consulta del correo del usuario a la base de datos. • Diseño de la interfaz. 	

Tabla 3.25. Actividades de la historia de usuario 2

Elaborado por: El investigador

3. Página Principal

El usuario podrá visualizar su nombre, una imagen y logo del consultorio, una barra lateral con los módulos definidos.

Código de historia: 3	Responsable: Valeria Logroño
Actividades	
<ul style="list-style-type: none"> • Consulta del nombre de usuario a la base de datos. • Diseño de la página principal. 	

Tabla 3.26. Actividades de la historia de usuario 3

Elaborado por: El investigador

3.2.2.2. Iteración 2

La iteración 2 está conformada por gestión pacientes, gestión de horarios de atención y reserva de citas, que se explica a continuación:

Código	Historia de usuario	Prioridad	Riesgo
4	Gestión Pacientes	Alta	Alta
5	Gestión Horarios de Atención	Alta	Alta
6	Reserva de Citas	Alta	Alta

Tabla 3.27. Historias de usuario de la iteración 2

Elaborado por: El investigador

4. Gestión Pacientes

El usuario podrá agregar, modificar o eliminar al paciente según sea necesario.

Código de historia: 4	Responsable: Valeria Logroño
Actividades	
<ul style="list-style-type: none"> • Elaboración del CRUD pacientes. • Diseño de la interfaz. 	

Tabla 3.28. Actividades de la historia de usuario 4

Elaborado por: El investigador

5. Gestión Horarios de Atención

El usuario podrá agregar, modificar o eliminar al paciente según sea necesario.

Código de historia: 5	Responsable: Valeria Logroño
Actividades	
<ul style="list-style-type: none"> • Elaboración del CRUD horarios atención • Diseño de la interfaz. 	

Tabla 3.29. Actividades de la historia de usuario 5

Elaborado por: El investigador

6. Reserva de Citas

El usuario podrá agendar una cita en el horario disponible con el odontólogo que corresponda.

Código de historia: 6	Responsable: Valeria Logroño
Actividades	
<ul style="list-style-type: none"> • Consulta de horarios disponibles en la base de datos. • Consulta de odontólogos y especialidades correspondientes. • Diseño de la interfaz 	

Tabla 3.30. Actividades de la historia de usuario 6

Elaborado por: El investigador

3.2.2.3. Iteración 3

La iteración 3 está conformada por gestión de la atención a pacientes, gestión de odontólogos y la creación del comprobante, que se explica a continuación:

Código	Historia de usuario	Prioridad	Riesgo
7	Gestión Atención Pacientes	Media	Media
8	Gestión Odontólogos	Alta	Alta
9	Comprobante	Alta	Alta

Tabla 3.31. Historias de usuario de la iteración 3

Elaborado por: El investigador

7. Gestión Atención Pacientes

El usuario podrá visualizar una lista con las citas pendientes de atender con la información de cada paciente.

Código de historia: 7	Responsable: Valeria Logroño
Actividades	
<ul style="list-style-type: none"> • Consulta de las citas pendientes de atención. • Diseño de la interfaz 	

Tabla 3.32. Actividades de la historia de usuario 7

Elaborado por: El investigador

8. Gestión Odontólogos

El usuario podrá agregar, modificar o eliminar al odontólogo según sea necesario.

Código de historia: 8	Responsable: Valeria Logroño
Actividades	
<ul style="list-style-type: none">• Elaboración del CRUD de odontólogos• Diseño de la interfaz	

Tabla 3.33. Actividades de la historia de usuario 8

Elaborado por: El investigador

9. Comprobante

El usuario podrá buscar al paciente por número de cedula, registrar los tratamientos y precios que se realizaron.

Código de historia: 9	Responsable: Valeria Logroño
Actividades	
<ul style="list-style-type: none">• Elaboración de los métodos.• Diseño de la interfaz	

Tabla 3.34. Actividades de la historia de usuario 9

Elaborado por: El investigador

3.2.2.4. Iteración 4

La iteración 4 está conformada por la generación del pdf del comprobante y finalizar sesión, que se explica a continuación:

Código	Historia de usuario	Prioridad	Riesgo
10	Generar Pdf del Comprobante	Alta	Alta
11	Finalizar Sesión	Media	Media

Tabla 3.35. Historias de usuario de la iteración 4

Elaborado por: El investigador

10. Generar Pdf del Comprobante

Se crea el formato de comprobante de los tratamientos realizados, se generan dos copias una para el paciente y otra para el centro odontológico. El usuario podrá imprimir los comprobantes en una misma página.

Código de historia: 10	Responsable: Valeria Logroño
Actividades	
<ul style="list-style-type: none">• Transferir la información del comprobante al pdf.• Diseño del formato pdf.	

Tabla 3.36. Actividades de la historia de usuario 10

Elaborado por: El investigador

11. Finalizar Sesión

El usuario será capaz de finalizar la sesión activa.

Código de historia: 11	Responsable: Valeria Logroño
Actividades	
<ul style="list-style-type: none">• Se desconectará la sesión activa del usuario.• Diseño de la interfaz	

Tabla 3.37. Actividades de la historia de usuario 11

Elaborado por: El investigador

3.2.2.5. Tarjetas CRC

En el desarrollo del proyecto se elaboró tarjetas CRC (Clase, Responsabilidad y Colaboración) con el propósito de identificar las responsabilidades y colaboradores de cada historia de usuario con el fin de mejorar el enfoque para el diseño del software orientado a objetos .

Ingreso al sistema	
Responsabilidad	Colaboradores
<ul style="list-style-type: none"> • Presentar el formulario de acceso al sistema. • Validar el correo y contraseña del usuario. 	<ul style="list-style-type: none"> • Clase IngresoSistema • Clase IngresoDAO • Clase IngresoLN
Observaciones: El único usuario encargado de ingresar al sistema es el odontólogo principal.	

Tabla 3.38. Tarjeta CRC – Ingreso al sistema

Elaborado por: El investigador

Recuperar Contraseña	
Responsabilidad	Colaboradores
<ul style="list-style-type: none"> • Presentar el formulario de recuperar contraseña. • Validar el correo y envió de mensaje. 	<ul style="list-style-type: none"> • Clase RecuperarPasswordDAO • Clase SupportMail • Clase recoverPasswordLN
Observaciones: Ninguna.	

Tabla 3.39. Tarjeta CRC – Recuperar Contraseña

Elaborado por: El investigador

Página Principal	
Responsabilidad	Colaboradores
<ul style="list-style-type: none"> • Mostrar el menú del sistema. • Visualizar nombre y fotografía del usuario activo. 	<ul style="list-style-type: none"> • Clase IngresoSistema • Clase IngresoDAO • Clase IngresoLN
Observaciones: Ninguna.	

Tabla 3.40. Tarjeta CRC – Página Principal

Elaborado por: El investigador

Gestión Pacientes	
Responsabilidad	Colaboradores
<ul style="list-style-type: none"> • Presentar el módulo de los pacientes y acciones permitidas como eliminar o modificar. • Ventana para agregar nuevo paciente. 	<ul style="list-style-type: none"> • Clase Paciente • Clase PacienteDAO • Clase PacienteLN
Observaciones: Ninguna.	

Tabla 3.41. Tarjeta CRC – Gestión Pacientes

Elaborado por: El investigador

Gestión Horarios de Atención	
Responsabilidad	Colaboradores
<ul style="list-style-type: none"> • Presentar el módulo de los horarios de atención y acciones permitidas como eliminar o modificar. • Ventana para agregar nuevo horario de atención. 	<ul style="list-style-type: none"> • Clase HorarioAtencion • Clase Hora • Clase HorarioAtencionDAO • Clase HorarioAtencionLN
Observaciones: Ninguna.	

Tabla 3.42. Tarjeta CRC – Gestión Horarios de Atención

Elaborado por: El investigador

Reserva de Citas	
Responsabilidad	Colaboradores
<ul style="list-style-type: none"> • Presentar el formulario para la reserva de cita 	<ul style="list-style-type: none"> • Clase HorarioAtencion • Clase Especialidad • Clase Paciente • Clase Cita
Observaciones: Ninguna.	

Tabla 3.43. Tarjeta CRC – Reserva de Citas.

Elaborado por: El investigador

Gestión Atención Pacientes	
Responsabilidad	Colaboradores
<ul style="list-style-type: none"> • Presentar el módulo de atención de citas y acciones permitidas como realizar atención y cancelar. 	<ul style="list-style-type: none"> • Clase Cita • Clase CitaLN
Observaciones: Ninguna.	

Tabla 3.44. Tarjeta CRC – Gestión Atención Pacientes

Elaborado por: El investigador

Gestión Odontólogos	
Responsabilidad	Colaboradores
<ul style="list-style-type: none"> • Presentar el módulo de los odontólogos y acciones permitidas como eliminar o modificar. • Ventana para agregar nuevo odontólogo. 	<ul style="list-style-type: none"> • Clase Odontologo • Clase OdontologoLN • Clase Especialidad • Clase TipoEmpleado • Clase OdontologoDAO
Observaciones: Ninguna.	

Tabla 3.45. Tarjeta CRC – Gestión Odontólogos

Elaborado por: El investigador

Comprobante	
Responsabilidad	Colaboradores
<ul style="list-style-type: none"> • Presentar el módulo del comprobante. 	<ul style="list-style-type: none"> • Clase Paciente • Clase PacienteLN
Observaciones: Ninguna.	

Tabla 3.46. Tarjeta CRC – Comprobante

Elaborado por: El investigador

Generar Pdf del Comprobante	
Responsabilidad	Colaboradores
<ul style="list-style-type: none"> • Presentar el formato del comprobante en PDF. 	<ul style="list-style-type: none"> • Clase Paciente • Clase PacienteLN
Observaciones: Se debe generar dos copias en la misma página.	

Tabla 3.47. Tarjeta CRC – Generar Pdf del Comprobante

Elaborado por: El investigador

Finalizar Sesión	
Responsabilidad	Colaboradores
<ul style="list-style-type: none"> • Concluir con la sesión del usuario. 	<ul style="list-style-type: none"> • Método finalizarSesion
Observaciones:	

Tabla 3.48. Tarjeta CRC – Finalizar Sesión

Elaborado por: El investigador

3.2.2.6. Diseño de modelo de datos

El diseño de la base de datos se lo elaboro entre el personal y el investigador, en el cual se identifican las entidades con sus respectivos campos, el esquema de la base de datos se los desarrollo con la herramienta SQL2018.

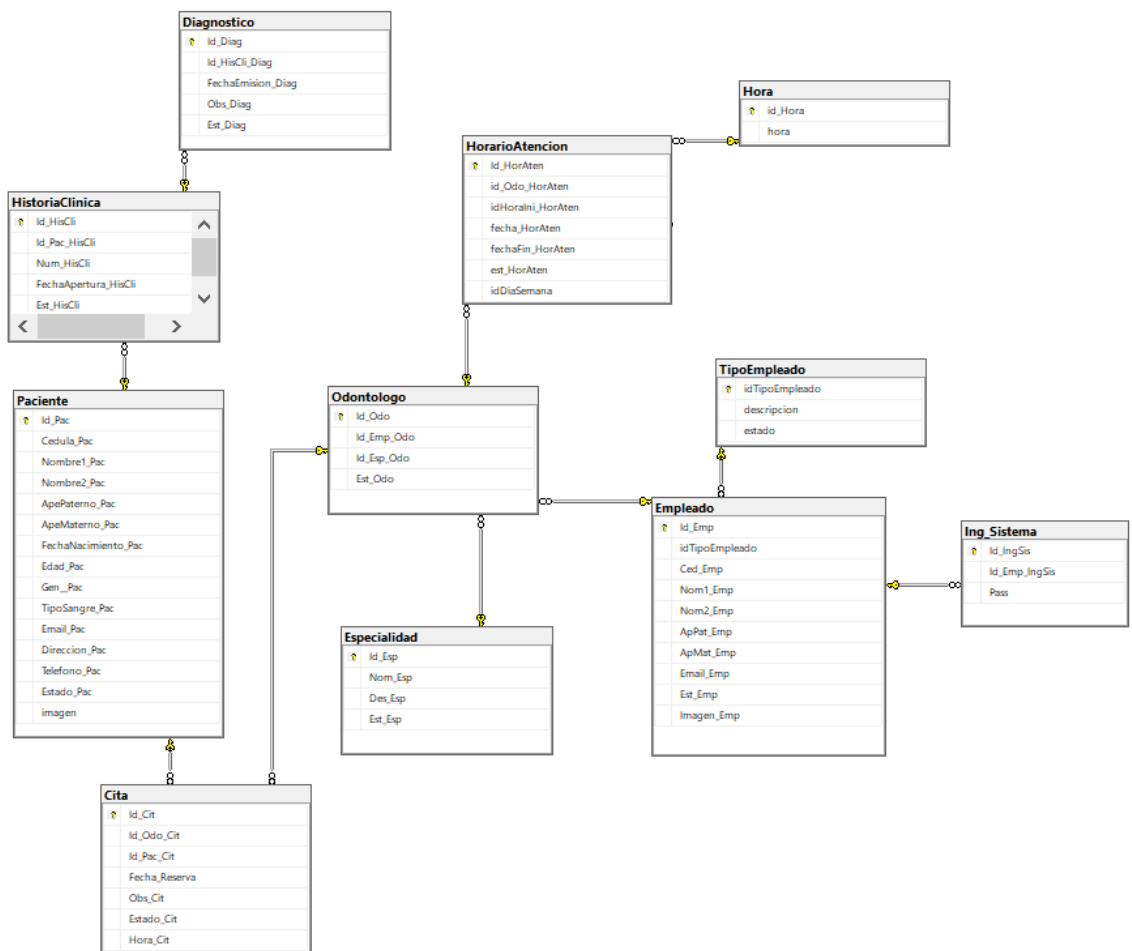


Gráfico 3.4. Modelo Entidad – Relación de la base de datos

Elaborado por: El investigador

3.2.2.7. Diseño de Iteraciones

A continuación, se presenta las interfaces de cada historia de usuario que conforman la aplicación web, fueron realizadas en MockFlow por poseer una interfaz intuitiva, amplia biblioteca de elementos y componentes.

1. Ingreso al sistema

Para el ingreso al sistema es necesario el correo electrónico y contraseña que se registró en la base de datos, existe la opción de recuperar su contraseña.

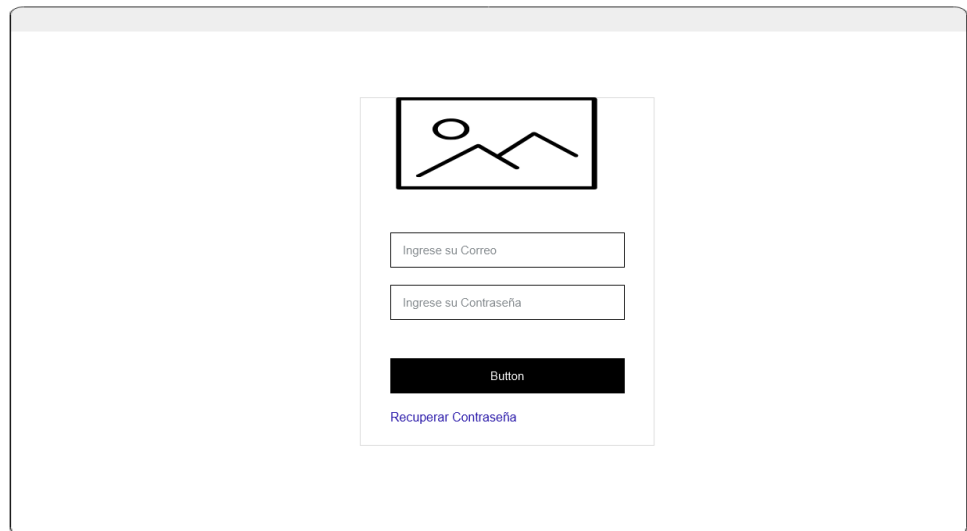


Gráfico 3.5. Inicio de Sesión

Elaborado por: El investigador

2. Recuperar Contraseña

En la página de recuperar contraseña, la aplicación le pide digitar el correo con el que se encuentra registrado. Luego se visualizará un mensaje dependiendo de la respuesta.

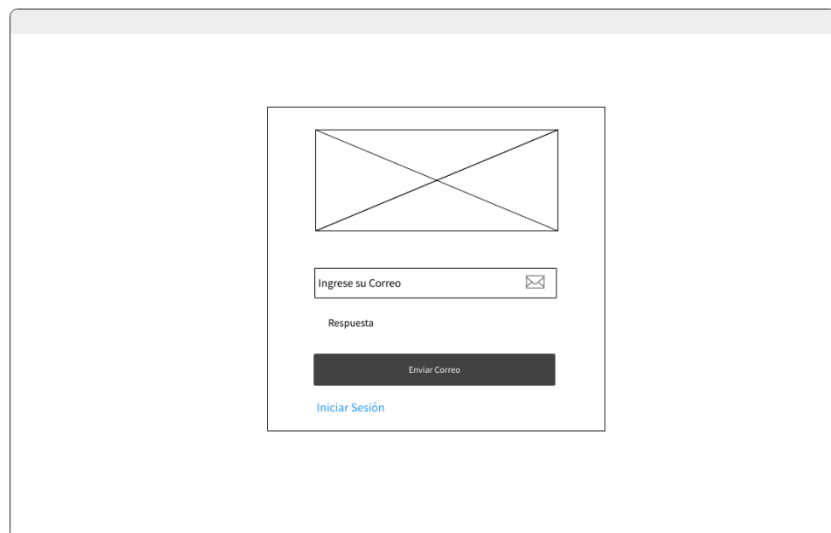


Gráfico 3.6. Recuperar Contraseña

Elaborado por: El investigador

3. Página Principal

En la página principal se muestra los distintos módulos que posee nuestro sistema. Además, se visualiza el usuario que está conectado con su respectiva foto.

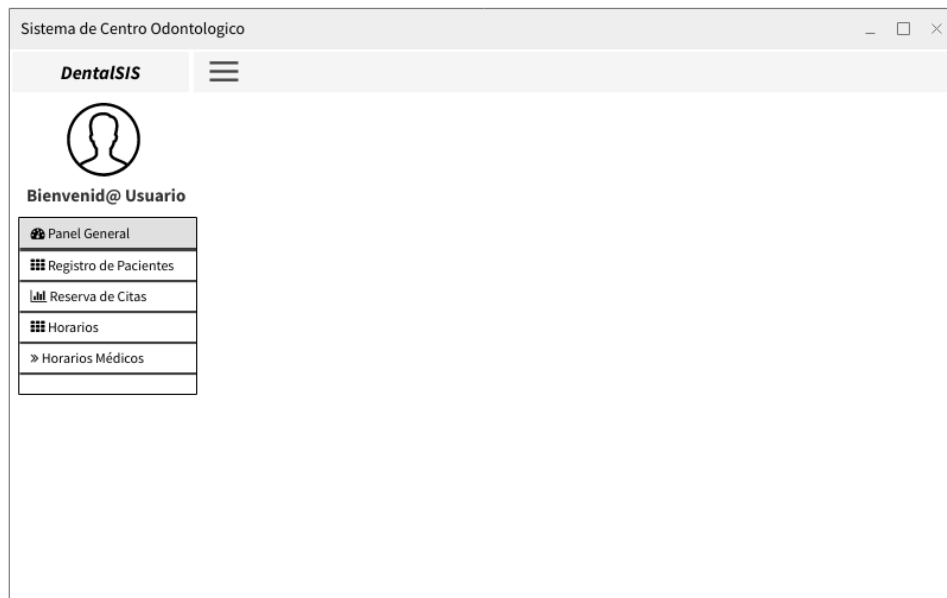


Gráfico 3.7. Página Principal

Elaborado por: El investigador

4. Gestión Pacientes

El usuario podrá registrar nuevos usuarios completando los campos correspondientes, en el listado se muestran acciones de eliminar y editar a pacientes.

Registro de Pacientes

Registrar
Cancelar

Lista de Pacientes

Cédula	Nombres	Apellidos	Email	Teléfono	Dirección	Acciones
1803589714	Janice Maribel	Torres Reyes	Janice_Monahan@y	0969-068-8439	Port Beulah, Iowa 90719, United States of America	
0578963214	Rollin Ayden	Vargas Lucero	Rollin_Fadel@gmail.	(017) 057-6055	Lake Matilde, Tennessee 74062, United States of	
1769254635	Lera Maylen	Chávez Rosero	Lera_Stroman3@gm	166-619-2267	Vicentaview, Mississippi 47576-9639, United	
1836974582	Adan Ryan	Zabala Gárces	Adan_Schiller19@ya	(699) 824-5724	VonRuedenberg, Delaware 99072-4003,	

Gráfico 3.8. Registro y listado de Pacientes

Elaborado por: El investigador

En las acciones del paciente, al seleccionar el icono de actualizar se muestra un modal con la información del paciente.

Actualizar Registro

Cancelar
Actualizar

Gráfico 3.9. Modal de Actualizar Paciente

Elaborado por: El investigador

5. Gestión Horarios de Atención

En la gestión de horarios el usuario debe ingresar un nombre y apellido de un odontólogo registrado para visualizar su información y los

horarios(fecha-hora) que posee, también puede agregar, eliminar o editar los horarios.

GESTIÓN DE HORARIOS DE ODONTÓLOGOS

Datos del Odontólogo

Apellido Nombre

Nombres:

Apellidos:

Especialidad:

Horario de Atención

		Fecha de Atención	Hora de Atención
<input type="button" value="✎"/>	<input type="button" value="🗑"/>	12/05/2022	09:00
<input type="button" value="✎"/>	<input type="button" value="🗑"/>	02/10/2022	11:30
<input type="button" value="✎"/>	<input type="button" value="🗑"/>	22/11/2022	16:00
<input type="button" value="✎"/>	<input type="button" value="🗑"/>	09/05/2023	17:30

Gráfico 3.10. Gestión de horarios de odontólogos

Elaborado por: El investigador

En la parte inferior de la sección de horario de atención se localiza el botón agregar horario, el cual al presionar muestra un modal con los datos respectivos.

Agregar Horario Atención

Fecha

Hora de inicio

Gráfico 3.11. Modal de Agregar Horario Atención

Elaborado por: El investigador

En los registros de los horarios de atención se encuentra dos botones (Actualizar - Eliminar), para la opción de actualizar se muestra el siguiente modal.

Editar Horario Atención

Fecha

Hora de Inicio

Editar Cancelar

Gráfico 3.12. Modal de Editar Horario Atención

Elaborado por: El investigador

6. Reserva de Citas

En la reserva de citas en la sección de datos el usuario debe ingresar el apellido y nombre del paciente para que se visualice su información, en la sección de horarios de atención se debe ingresar una fecha y seleccionar la especialidad que se solicite. Después de lo anterior se elige la hora y el odontólogo solicitado para luego reservar cita.

RESERVA DE CITAS

DATOS DEL PACIENTE

Apellido Nombre

TELÉFONO

NOMBRES

EDAD

APELLIDOS

GÉNERO

HORARIOS DE ATENCIÓN

FECHA ESPECIALIDAD

	HORA DE ATENCIÓN	ODONTÓLOGO
<input checked="" type="checkbox"/>	16:00	Janice Torres
<input checked="" type="checkbox"/>	10:00	Rollin Vargas
<input checked="" type="checkbox"/>	12:00	Lera Chávez

Gráfico 3.13. Gestión de reserva de citas

Elaborado por: El investigador

7. Gestión Atención Pacientes

El usuario podrá visualizar una lista con las citas pendientes de atender con los datos relevantes de cada paciente. Además, se puede realizar la atención o cancelar.

The screenshot displays a web interface titled "GESTIONAR ATENCIÓN ODONTOLÓGICA" with a sub-header "FECHA ACTUAL". It features two identical appointment cards stacked vertically. Each card contains a square placeholder with a large 'X' for a patient photo, a label "HORA DE ATENCIÓN:" below it, and a form on the right with fields for "ID CITA:", "NOMBRES:", "APELLIDOS:", "EDAD:", and "GÉNERO:". At the bottom of each card are two buttons: "Realizar Cita" and "Cancelar".

Gráfico 3.14. Gestión Atención Pacientes

Elaborado por: El investigador

En el momento que el usuario seleccione realizar la atención, se mostrará una ventana en la cual el usuario podrá registrar su diagnóstico y observaciones.

The screenshot shows a form titled "DATOS DEL PACIENTE". On the left is a square placeholder with a large 'X' for a patient photo. To its right are fields for "NOMBRES:", "APELLIDOS:", "EDAD:", and "GÉNERO:". On the right side of the form is a text area labeled "OBSERVACIONES". At the bottom is a large text area labeled "DIAGNÓSTICO". A "Registrar" button is located at the bottom center of the form.

Gráfico 3.15. Registrar Diagnostico

Elaborado por: El investigador

8. Gestión Odontólogos

El usuario podrá registrar nuevos odontólogos completando los campos correspondientes, en el listado se muestran acciones de eliminar y editar a odontólogos.

REGISTRO DE ODONTÓLOGOS

<input type="text" value="CÉDULA DE IDENTIDAD"/>	<input type="text" value="APELLIDO MATERNO"/>
<input type="text" value="PRIMER NOMBRE"/>	<input type="text" value="CORREO ELECTRÓNICO"/>
<input type="text" value="SEGUNDO NOMBRE"/>	<input type="text" value="ESPECIALIDAD"/>
<input type="text" value="APELLIDO PATERNO"/>	<input type="text" value="TELÉFONO"/>

LISTA DE ODONTÓLOGOS

Cédula	Nombres	Apellidos	Email	Especialidad	Acciones
1803589714	Janice Maribel	Torres Reyes	Janice_Monahan@yahoo.com	Periodoncia	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
0578963214	Rollin Ayden	Vargas Lucero	Rollin_Fadel@gmail.com	Endodoncia	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
1769254635	Lera Maylen	Chávez Rosero	Lera_Stroman3@gmail.com	Odontopediatría	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
1836974582	Adan Ryan	Zabala Gárces	Adan_Schiller19@yahoo.com	Periodoncia	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Gráfico 3.16. Registro y listado de Odontólogos

Elaborado por: El investigador

En las acciones del odontólogo, al seleccionar el icono de actualizar se muestra un modal con la información del odontólogo.

Actualizar Registro

<input type="text" value="Nombres y Apellidos"/>
<input type="text" value="Correo Electrónico"/>
<input type="text" value="Teléfono"/>

Gráfico 3.17. Modal de Actualizar Odontólogo

Elaborado por: El investigador

11. Finalizar Sesión

En la esquina superior derecha el usuario podrá cerrar sesión y salir del sistema.



Gráfico 3.20. Finalizar Sesión

Elaborado por: El investigador

3.2.3. Fase III: Codificación

3.2.3.1. Creación de la Solución en Visual Studio 2019

La solución del proyecto va a estar dividida en 4 capas: Acceso a Datos, Entidades, Lógica de Negocio y Presentación.

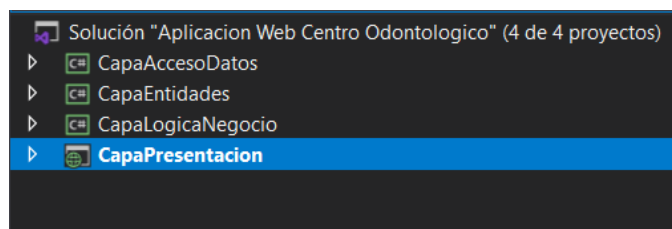


Gráfico 3.21. Capas de la solución

Elaborado por: El investigador

3.2.3.2. Conexión de base de datos

En la conexión a la base de datos se instaló el conector a MySQL a ASP.NET Web Applications. Se creó una clase conexión en la capa de Acceso a Datos, en la cual se implementó el patrón Singleton para instancia la conexión.

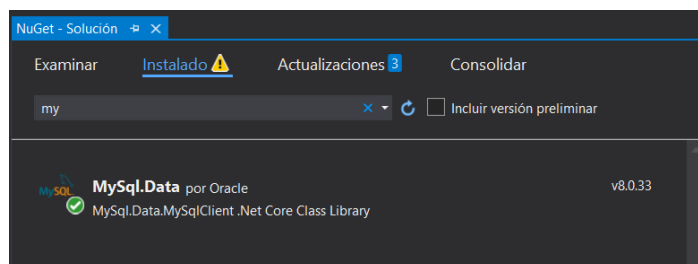


Gráfico 3.22. Conector MySQL para ASP.NET Web Applications

Elaborado por: El investigador

```

C# CapaAccesoDatos
CapaAccesoDatos.Conexion getInstance()
1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7 using MySql.Data.MySqlClient;
8
9
10 namespace CapaAccesoDatos
11 {
12     23 referencias
13     public class Conexion
14     {
15         #region "PATRON SINGLETON"
16         private static Conexion conexion = null;
17         private Conexion() { }
18         public static Conexion getInstance()
19         {
20             if(conexion == null)
21             {
22                 conexion = new Conexion();
23             }
24             return conexion;
25         }
26         #endregion
27         19 referencias
28         public MySqlConnection ConexionDB()
29         {
30             MySqlConnection conexion = new MySqlConnection();
31             // SqlConnection conexion = new SqlConnection();
32             conexion.ConnectionString= "server=localhost; database=prueba_co; user=root; password=valeria.17L";
33             return conexion;
34         }
35     }
36 }

```

Gráfico 3.23. Clase Conexión

Elaborado por: El investigador

3.2.3.3. Desarrollo Back - End

1. Creación de las entidades

Se establecen las entidades que representan la lógica del cliente.

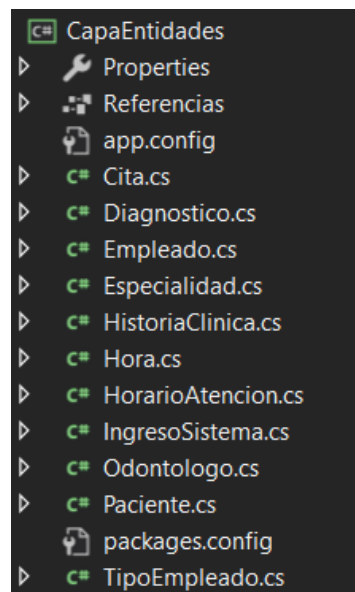


Gráfico 3.24. Entidades del proyecto

Elaborado por: El investigador

En las clases de cada entidad se implementa métodos get y set y constructores de los atributos especificados en la base de datos.

- Entidad Cita

```
public class Cita
{
    public int IdCita { get; set; }
    public Odontologo Odontologo { get; set; }
    public Paciente Paciente { get; set; }
    public DateTime FechaReserva { get; set; }
    public String ObsReserva { get; set; }
    public char EstReserva { get; set; }
    public String HoraReserva { get; set; }
    public Cita(){
        this.Paciente = new Paciente();
        this.Odontologo = new Odontologo();
    }
}
```

Gráfico 3.25. Entidad Cita

Elaborado por: El investigador

- Entidad Diagnostico

```
public class Diagnostico
{
    public int IdDiagnostico { get; set; }
    public HistoriaClinica HistoriaClinica { get; set; }
    public DateTime FechaEmision { get; set; }
    public String Observacion { get; set; }
    public String diagnostico { get; set; }
    public bool Estado { get; set; }
    public Diagnostico()
    {
        this.HistoriaClinica = new HistoriaClinica();
    }
}
```

Gráfico 3.26. Entidad Diagnostico

Elaborado por: El investigador

- Entidad Empleado

```
public class Empleado
{
    6 referencias
    public int ID { get; set; }
    1 referencia
    public TipoEmpleado RTipoEmpleado { get; set; }
    3 referencias
    public String Cedula { get; set; }
    7 referencias
    public String Nombre1 { get; set; }
    4 referencias
    public String Nombre2 { get; set; }
    7 referencias
    public String ApellidoPat { get; set; }
    4 referencias
    public String ApellidoMat { get; set; }
    4 referencias
    public String Email { get; set; }
    4 referencias
    public bool Estado { get; set; }
    1 referencia
    public String Imagen { get; set; }
    3 referencias
    public String Telefono { get; set; }

    2 referencias
    public Empleado()
    {
    }

    2 referencias
    public Empleado(int ID,TipoEmpleado RTipoEmpleado,String Cedula,String Nombre1,String Nombre2,
        String ApellidoPat,String ApellidoMat,String Email,bool Estado,String Imagen,
        String Telefono)
    {
        this.ID = ID;
        this.RTipoEmpleado = RTipoEmpleado;
        this.Cedula = Cedula;
        this.Nombre1 = Nombre1;
        this.Nombre2 = Nombre2;
        this.ApellidoPat = ApellidoPat;
        this.ApellidoMat = ApellidoMat;
        this.Email = Email;
        this.Estado = Estado;
        this.Imagen = Imagen;
        this.Telefono = Telefono;
    }
}
```

Gráfico 3.27. Entidad Empleado

Elaborado por: El investigador

- Entidad Especialidad

```
public class Especialidad
{
    3 referencias
    public int ID_Espec { get; set; }
    5 referencias
    public String Nombre { get; set; }
    1 referencia
    public String Descripcion { get; set; }
    1 referencia
    public bool Estado { get; set; }

    3 referencias
    public Especialidad()
    {
    }

    0 referencias
    public Especialidad(int ID_Espec, String Nombre,
        String Descripcion, bool Estado)
    {
        this.ID_Espec = ID_Espec;
        this.Nombre = Nombre;
        this.Descripcion = Descripcion;
        this.Estado = Estado;
    }
}
```

Gráfico 3.28. Entidad Especialidad

Elaborado por: El investigador

- Entidad HistoriaClinica

```

public class HistoriaClinica
{
    public int IdHistCli { get; set; }
    2 referencias
    public Paciente Paciente { get; set; }
    0 referencias
    public DateTime FechaApertura { get; set; }
    0 referencias
    public bool Estado { get; set; }

    1 referencia
    public HistoriaClinica()
    {
        this.Paciente = new Paciente();
    }
}

```

Gráfico 3.29. Entidad HistoriaClinica

Elaborado por: El investigador

- Entidad Hora

```

public class Hora
{
    2 referencias
    public int idHora { get; set; }
    7 referencias
    public String hora { get; set; }

    5 referencias
    public Hora() { }
}

```

Gráfico 3.30. Entidad Hora

Elaborado por: El investigador

- Entidad HorarioAtencion

```

public class HorarioAtencion
{
    public int IdHorarioAtencion { get; set; }
    6 referencias
    public Odontologo Odontologo { get; set; }
    7 referencias
    public Hora Hora { get; set; }
    7 referencias
    public DateTime Fecha { get; set; }
    1 referencia
    public bool Estado { get; set; }

    5 referencias
    public HorarioAtencion() { }
}

```

Gráfico 3.31. Entidad HorarioAtencion

Elaborado por: El investigador

- Entidad IngresoSistema

La clase de ingreso al sistema se hereda datos de la clase Empleado, por lo cual en el constructor de IngresoSistema se debe implementar el constructor de la clase Empleado.

```
public class IngresoSistema:Empleado
{
    public int ID_IngSis { get; set; }
    2 referencias
    public string Password { get; set; }

    2 referencias
    public IngresoSistema()
        :base()
    {
    }
}
0 referencias
public IngresoSistema(int ID, string Password)
    :base(0, new TipoEmpleado(), "", "", "", "", "", "", true, "", "")
{
    this.ID = ID;
    this.Password = Password;
}
}
```

Gráfico 3.32. Entidad IngresoSistema

Elaborado por: El investigador

- Entidad Odontólogo

La clase de odontólogo se hereda datos de la clase Empleado, por lo cual en el constructor de odontólogo se debe implementar el constructor de la clase Empleado.

```
public class Odontologo : Empleado
{
    11 referencias
    public int IdOdontologo { get; set; }
    4 referencias
    public Especialidad Especialidad { get; set; }
    1 referencia
    public bool EstOdontologo { get; set; }

    6 referencias
    public Odontologo()
        : base()
    {
    }
}
0 referencias
public Odontologo(int IdOdontologo,Especialidad Especialidad, bool EstOdontologo)
    :base(0, new TipoEmpleado(),"", "", "", "", "", "", true, "", "")
{
    this.IdOdontologo = IdOdontologo;
    this.Especialidad = Especialidad;
    this.EstOdontologo = EstOdontologo;
}
}
```

Gráfico 3.33. Entidad Odontólogo

Elaborado por: El investigador

- Entidad Paciente

```
public class Paciente
{
    11 referencias
    public int ID { get; set; }
    7 referencias
    public String Cedula { get; set; }
    8 referencias
    public String Nombre1 { get; set; }
    8 referencias
    public String Nombre2 { get; set; }
    8 referencias
    public String ApellidoPaterno { get; set; }
    8 referencias
    public String ApellidoMaterno { get; set; }
    4 referencias
    public DateTime FechaNacimiento { get; set; }
    8 referencias
    public int Edad { get; set; }
    8 referencias
    public String Genero { get; set; }
    4 referencias
    public String TipoSangre { get; set; }
    6 referencias
    public String Email { get; set; }
    7 referencias
    public String Direccion { get; set; }
    7 referencias
    public String Telefono { get; set; }
    4 referencias
    public bool Estado { get; set; }
    1 referencia
    public String Imagen { get; set; }

    7 referencias
    public Paciente()
    {
    }

    0 referencias
    public Paciente(
        int ID, String Cedula, String Nombre1, String Nombre2, String ApellidoPaterno,
        String ApellidoMaterno, DateTime FechaNacimiento, int Edad, String Genero,
        String TipoSangre, String Email, String Direccion, String Telefono, bool Estado,
        String Imagen
    )
    {
        this.ID=ID;
        this.Cedula = Cedula;
        this.Nombre1 = Nombre1;
        this.Nombre2 = Nombre2;
        this.ApellidoPaterno = ApellidoPaterno;
        this.ApellidoMaterno = ApellidoMaterno;
        this.FechaNacimiento = FechaNacimiento;
        this.Edad = Edad;
        this.Genero = Genero;
        this.TipoSangre = TipoSangre;
        this.Email = Email;
        this.Direccion = Direccion;
        this.Telefono = Telefono;
        this.Estado = Estado;
        this.Imagen = Imagen;
    }
}
```

Gráfico 3.34. Entidad Paciente

Elaborado por: El investigador

- Entidad TipoEmpleado

```
public class TipoEmpleado
{
    1 referencia
    public int ID { get; set; }
    1 referencia
    public String Descripcion { get; set; }
    1 referencia
    public bool Estado { get; set; }
    2 referencias
    public TipoEmpleado()
    {
        ...
    }
    0 referencias
    public TipoEmpleado(int ID, String Descripcion, bool Estado)
    {
        this.ID = ID;
        this.Descripcion = Descripcion;
        this.Estado = Estado;
    }
}
```

Gráfico 3.35. Entidad TipoEmpleado

Elaborado por: El investigador

2. Desarrollo de los objetos de acceso a datos

En cada una de las clases se implementó el patrón Singleton, además para la obtención de datos se utilizó procedimientos almacenados.

Estructura del Patrón Singleton

```
#region "PATRON SINGLETON"
//Instancia de la clase
private static Nombre_Clase _instance = null;
//Constructor
1 referencia
private Nombre_Clase() { }
//Propiedad publica que da acceso a la clase
9 referencias
public static Nombre_Clase getInstance()
{
    //solo si no existe ninguna instancia,
    //entonces se crea una nueva
    if (_instance == null)
    {
        _instance = new Nombre_Clase();
    }
    return _instance;
}
#endregion
```

Gráfico 3.36. Estructura del Patrón Singleton

Elaborado por: El investigador

- CitaDAO

Se codificó los métodos de registrar, listar y actualizar citas.

Método para registrar cita.

```

public bool RegistrarCita(Cita objCita)
{
    MySqlConnection con = null;
    MySqlCommand cmd = null;
    bool response = false;

    try
    {
        con = Conexion.getInstance().ConexionDB();
        cmd = new MySqlCommand("spRegistrarCita", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@prmIdOdontologo", objCita.Odontologo.IdOdontologo);
        cmd.Parameters.AddWithValue("@prmIdPaciente ", objCita.Paciente.ID);
        cmd.Parameters.AddWithValue("@prmFechaReserva", objCita.FechaReserva);
        cmd.Parameters.AddWithValue("@prmHoraReserva", objCita.HoraReserva);

        con.Open();

        int filas = cmd.ExecuteNonQuery();
        if (filas > 0) response = true;
    }
    catch (Exception ex)
    {
        response = false;
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return response;
}

```

Gráfico 3.37. Método registrar cita

Elaborado por: El investigador

Método para listar citas.

```

public List<Cita> ListarCitas()
{
    MySqlConnection con = null;
    MySqlCommand cmd = null;
    MySqlDataReader dr = null;
    List<Cita> listaCitas = new List<Cita>();

    try
    {
        con = Conexion.getInstance().ConexionDB();
        cmd = new MySqlCommand("spListarCitas", con);
        cmd.CommandType = CommandType.StoredProcedure;

        con.Open();
        dr = cmd.ExecuteReader();
        while (dr.Read())
        {
            //llenar la lista con las citas
            Cita cita = new Cita();

            cita.IdCita = Convert.ToInt32(dr["Id_Cit"].ToString());
            cita.Odontologo.IdOdontologo = Convert.ToInt32(dr["Id_Odo_Cit"].ToString());
            cita.Paciente.ID = Convert.ToInt32(dr["Id_Pac_Cit"].ToString());
            cita.FechaReserva = Convert.ToDateTime(dr["Fecha_Reserva"].ToString());
            cita.HoraReserva = dr["Hora_Cit"].ToString();

            cita.Paciente.Nombre1 = dr["Nombre1_Pac"].ToString();
            cita.Paciente.Nombre2 = dr["Nombre2_Pac"].ToString();
            cita.Paciente.ApellidoPaterno = dr["ApePaterno_Pac"].ToString();
            cita.Paciente.ApellidoMaterno = dr["ApeMaterno_Pac"].ToString();
            cita.Paciente.Edad = Convert.ToInt32(dr["Edad_Pac"].ToString());
            cita.Paciente.Genero = dr["Gen_Pac"].ToString();
            cita.Paciente.Cedula = dr["Cedula_Pac"].ToString();
            cita.Paciente.Direccion = dr["Direccion_Pac"].ToString();

            listaCitas.Add(cita);
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return listaCitas;
}

```

Gráfico 3.38. Método listar citas

Elaborado por: El investigador

Método para actualizar cita.

```
public bool ActualizarCita(Int32 idCita, string estado)
{
    MySqlConnection con = null;
    MySqlCommand cmd = null;
    bool response = false;

    try
    {
        con = Conexion.GetInstance().ConexionDB();
        cmd = new MySqlCommand("spActualizarEstadoCita", con);
        cmd.Parameters.AddWithValue("@prmIdCita", idCita);
        cmd.Parameters.AddWithValue("@prmEstado", estado);
        cmd.CommandType = CommandType.StoredProcedure;

        con.Open();

        //Ejecutar al procedimiento almacenado
        cmd.ExecuteNonQuery();

        response = true;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return response;
}
```

Gráfico 3.39. Método actualizar cita

Elaborado por: El investigador

- DiagnosticoDAO

Se codifico el registro de los diagnósticos de cada una de las citas.

```
public bool RegistrarDiagnostico(Diagnostico objDiagnostico)
{
    MySqlConnection con = null;
    MySqlCommand cmd = null;
    bool response = false;

    try
    {
        con = Conexion.GetInstance().ConexionDB();
        cmd = new MySqlCommand("spRegistrarDiagnostico", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@prmIdPaciente", objDiagnostico.HistoriaClinica.Paciente.ID);
        cmd.Parameters.AddWithValue("@prmObservacion", objDiagnostico.Observacion);
        cmd.Parameters.AddWithValue("@prmDiagnostico", objDiagnostico.diagnostico);

        con.Open();

        cmd.ExecuteNonQuery();
        response = true;
    }
    catch (Exception ex)
    {
        response = false;
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return response;
}
```

Gráfico 3.40. Método registrar diagnostico

Elaborado por: El investigador

- EspecialidadDAO

Se codifico el método para listar los nombres de cada una de las especialidades.

```

public List<Especialidad> ListarEspecialidades()
{
    MySqlConnection con = Conexion.getInstance().ConexionDB();
    MySqlCommand cmd = null;
    MySqlDataReader dr = null;
    List<Especialidad> Lista = null;

    try
    {
        Lista = new List<Especialidad>();
        cmd = new MySqlCommand("spListarEspecialidades", con);
        cmd.CommandType = CommandType.StoredProcedure;
        con.Open();
        dr = cmd.ExecuteReader();

        while (dr.Read())
        {
            Especialidad objEspecialidad = new Especialidad();
            objEspecialidad.ID_Espec = Convert.ToInt32(dr["Id_Esp"].ToString());
            objEspecialidad.Nombre = dr["Nom_Esp"].ToString();
            Lista.Add(objEspecialidad);
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return Lista;
}

```

Gráfico 3.41. Método listar especialidades

Elaborado por: El investigador

- HorarioAtencionDAO

Se codifico los métodos de registrar, actualizar, eliminar, listar horarios de atención y reserva.

Método registrar horario de atención.

```

public HorarioAtencion RegistrarHorarioAtencion(HorarioAtencion objHorarioAtencion)
{
    MySqlConnection con = Conexion.getInstance().ConexionDB();
    MySqlCommand cmd = null;
    MySqlDataReader dr = null;
    HorarioAtencion objHorario = null;

    try
    {
        cmd = new MySqlCommand("spRegistrarHorarioAtencion", con);
        cmd.Parameters.AddWithValue("@prmIdOdontologo", objHorarioAtencion.Odontologo.IdOdontologo);
        cmd.Parameters.AddWithValue("@prmHora", objHorarioAtencion.Hora.hora);
        cmd.Parameters.AddWithValue("@prmFecha", objHorarioAtencion.Fecha);
        cmd.CommandType = CommandType.StoredProcedure;

        con.Open();

        dr = cmd.ExecuteReader();

        if (dr.Read())
        {
            //Generar el objeto de HorarioAtencion
            //HA_Id_HorAten,HA_fecha_HorAten,H_id_Hora,H_hora,HA_est_HorAten
            objHorario = new HorarioAtencion()
            {
                IdHorarioAtencion = Convert.ToInt32(dr["Id_HorAten"].ToString()),
                Fecha = Convert.ToDateTime(dr["fecha_HorAten"].ToString()),
                Hora = new Hora()
                {
                    IdHora = Convert.ToInt32(dr["id_Hora"].ToString()),
                    hora = dr["hora"].ToString()
                },
                Estado = Convert.ToBoolean(dr["est_HorAten"].ToString())
            };
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return objHorario;
}

```

Gráfico 3.42. Método registrar horario atención

Elaborado por: El investigador

Método actualizar horario de atención.

```
public bool ActualizarHorarioAtencion(HorarioAtencion objHorario)
{
    MySqlConnection con = Conexion.getInstance().ConexionDB();
    MySqlCommand cmd = null;
    bool ok = false;

    try
    {
        cmd = new MySqlCommand("spActualizarHorarioAtencion", con);
        cmd.Parameters.AddWithValue("@prmIdOdontologo", objHorario.Odontologo.IdOdontologo);
        cmd.Parameters.AddWithValue("@prmIdHorario", objHorario.IdHorarioAtencion);
        cmd.Parameters.AddWithValue("@prmFecha", objHorario.Fecha);
        cmd.Parameters.AddWithValue("@prmHora", objHorario.Hora.hora);
        cmd.CommandType = CommandType.StoredProcedure;

        con.Open();

        cmd.ExecuteNonQuery();

        ok = true;
    }
    catch (Exception ex)
    {
        ok = false;
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return ok;
}
```

Gráfico 3.43. Método actualizar horario de atención.

Elaborado por: El investigador

Método eliminar horario de atención.

```
public bool EliminarHorarioAtencion(int IdHorarioAtencion)
{
    MySqlConnection con = Conexion.getInstance().ConexionDB();
    MySqlCommand cmd = null;
    bool ok = false;

    try
    {
        cmd = new MySqlCommand("spEliminarHorarioAtencion", con);
        cmd.Parameters.AddWithValue("@prmIdHorarioAtencion", IdHorarioAtencion);
        cmd.CommandType = CommandType.StoredProcedure;

        con.Open();

        cmd.ExecuteNonQuery();

        ok = true;
    }
    catch (Exception ex)
    {
        ok = false;
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return ok;
}
```

Gráfico 3.44. Método eliminar horario de atención.

Elaborado por: El investigador

Método listar horario de atención por id del odontólogo.

```
public List<HorarioAtencion> ListarHorariosAtencion(Int32 IdOdontologo)
{
    MySqlConnection con = Conexion.GetInstance().ConexionDB();
    MySqlCommand cmd = null;
    MySqlDataReader dr = null;
    List<HorarioAtencion> Lista = null;

    try
    {
        cmd = new MySqlCommand("spListarHorariosAtencion", con);
        cmd.Parameters.AddWithValue("@prmIdOdontologo", IdOdontologo);
        cmd.CommandType = CommandType.StoredProcedure;

        con.Open();

        dr = cmd.ExecuteReader();

        Lista = new List<HorarioAtencion>();

        while (dr.Read())
        {
            // llenamos los objetos
            HorarioAtencion objHorario = new HorarioAtencion();
            objHorario.IdHorarioAtencion = Convert.ToInt32(dr["Id_HorAten"].ToString());
            objHorario.Fecha = Convert.ToDateTime(dr["Fecha_HorAten"].ToString());
            objHorario.Hora = new Hora()
            {
                hora = dr["hora"].ToString()
            };

            Lista.Add(objHorario);
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return Lista;
}
```

Gráfico 3.45. Método listar horario de atención por id del odontólogo

Elaborado por: El investigador

Método listar horario de reserva por fecha.

```
public List<HorarioAtencion> ListarHorarioReserva(Int32 IdEspecialidad, DateTime Fecha)
{
    MySqlConnection con = Conexion.GetInstance().ConexionDB();
    MySqlCommand cmd = null;
    MySqlDataReader dr = null;
    List<HorarioAtencion> Lista = null;

    try
    {
        cmd = new MySqlCommand("spListarHorarioAtencionPorFecha", con);
        cmd.Parameters.AddWithValue("@prmIdEspecialidad", IdEspecialidad);
        cmd.Parameters.AddWithValue("@prmFecha", Fecha);
        cmd.CommandType = CommandType.StoredProcedure;

        con.Open();

        dr = cmd.ExecuteReader();

        Lista = new List<HorarioAtencion>();

        while (dr.Read())
        {
            // llenamos los objetos
            HorarioAtencion objHorarioAtencion = new HorarioAtencion();
            Odontologo objOdontologo = new Odontologo();
            Hora objHora = new Hora();

            objHora.IdHora = Convert.ToInt32(dr["id_Hora"].ToString());
            objHora.hora = dr["hora"].ToString();
            objHorarioAtencion.Hora = objHora;

            objOdontologo.IdOdontologo = Convert.ToInt32(dr["Id_Odo"].ToString());
            objOdontologo.Nombre1 = dr["Nom1_Emp"].ToString();
            objOdontologo.ApellidoPat = dr["ApPat_Emp"].ToString();
            objHorarioAtencion.Odontologo = objOdontologo;

            objHorarioAtencion.IdHorarioAtencion = Convert.ToInt32(dr["Id_HorAten"].ToString());
            objHorarioAtencion.Fecha = Convert.ToDateTime(dr["Fecha_HorAten"].ToString());

            Lista.Add(objHorarioAtencion);
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return Lista;
}
```

Gráfico 3.46. Método listar horario de atención por fecha

Elaborado por: El investigador

- IngresoDAO

Se codifico el acceso al sistema y obtener los nombres del usuario que ingreso.

Método para acceder al sistema.

```
public IngresoSistema AccesoSistema(String correo, String clave)
{
    MySqlConnection con = null;
    MySqlCommand cmd = null;
    IngresoSistema objingresoSistema = null;
    MySqlDataReader dr = null;

    try
    {
        con = Conexion.getInstance().ConexionDB();
        cmd = new MySqlCommand("spAccesoSistema", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@prmEmail", correo);
        cmd.Parameters.AddWithValue("@prmPass", clave);
        con.Open();
        dr = cmd.ExecuteReader();
        if(dr.Read())
        {
            objingresoSistema = new IngresoSistema()
            {
                ID_IngSis = Convert.ToInt32(dr["Id_Emp_IngSis"].ToString()),
                Password = dr["Pass"].ToString(),
                ID = Convert.ToInt32(dr["Id_Emp"].ToString()),
                Email = dr["Email_Emp"].ToString()
            };
        }
    }
    catch(Exception ex)
    {
        objingresoSistema = null;
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return objingresoSistema;
}
```

Gráfico 3.47. Método para acceder al sistema.

Elaborado por: El investigador

Método para obtener el nombre del Odontólogo

```
public IngresoSistema NombreOdontologo(int32 idEmpleado)
{
    MySqlConnection con = null;
    MySqlCommand cmd = null;
    IngresoSistema objingresoSistema = null;
    MySqlDataReader dr = null;

    try
    {
        con = Conexion.getInstance().ConexionDB();
        cmd = new MySqlCommand("spBuscarEmpleado", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@prmIdEmpleado", idEmpleado);
        con.Open();
        dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            objingresoSistema = new IngresoSistema()
            {
                ID = Convert.ToInt32(dr["Id_Emp"].ToString()),
                Nombre1 = dr["Nom1_Emp"].ToString(),
                ApellidoPat = dr["ApPat_Emp"].ToString()
            };
        }
    }
    catch (Exception ex)
    {
        objingresoSistema = null;
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return objingresoSistema;
}
```

Gráfico 3.48. Método para obtener el nombre del odontólogo

Elaborado por: El investigador

- OdontologoDAO

Se codifico los métodos de buscar, listar, registrar, actualizar y eliminar.

Método para buscar odontólogo por cedula de identidad.

```
public Odontologo BuscarOdontologo(String cedula)
{
    MySqlConnection con = null;
    MySqlCommand cmd = null;
    MySqlDataReader dr = null;
    Odontologo objOdontologo = null;

    try
    {
        con = Conexion.getInstance().ConexionDB();
        cmd = new MySqlCommand("spBuscarOdontologo", con);
        cmd.Parameters.AddWithValue("@prmCedula", cedula);
        cmd.CommandType = CommandType.StoredProcedure;

        con.Open();

        dr = cmd.ExecuteReader();

        if (dr.Read())
        {
            //instanciar objeto
            objOdontologo = new Odontologo()
            {
                IdOdontologo = Convert.ToInt32(dr["Id_Odo"].ToString()),
                ID = Convert.ToInt32(dr["Id_Emp"].ToString()),
                Nombre1 = dr["Nom1_Emp"].ToString(),
                Nombre2 = dr["Nom2_Emp"].ToString(),
                ApellidoPat = dr["ApPat_Emp"].ToString(),
                ApellidoMat = dr["ApMat_Emp"].ToString(),
                Especialidad = new Especialidad()
                {
                    ID_Espec= Convert.ToInt32(dr["Id_Esp"].ToString()),
                    Nombre= dr["Nom_Esp"].ToString()
                },
                Estado = Convert.ToBoolean(dr["Est_Odo"].ToString())
            };
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return objOdontologo;
}
```

Gráfico 3.49. Método para buscar odontólogo por cedula de identidad.

Elaborado por: El investigador

Método para registrar odontólogo.

```

public bool RegistrarOdontologo(Odontologo objOdontologo)
{
    MySqlConnection con = null;
    MySqlCommand cmd = null;
    bool response = false;

    try
    {
        con = Conexion.getInstance().ConexionDB();
        cmd = new MySqlCommand("spRegistrarOdontologo", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@prmCedula", objOdontologo.Cedula);
        cmd.Parameters.AddWithValue("@prmNombre1", objOdontologo.Nombre1);
        cmd.Parameters.AddWithValue("@prmNombre2", objOdontologo.Nombre2);
        cmd.Parameters.AddWithValue("@prmApellidoPat", objOdontologo.ApellidoPat);
        cmd.Parameters.AddWithValue("@prmApellidoMat", objOdontologo.ApellidoMat);
        cmd.Parameters.AddWithValue("@prmEmail", objOdontologo.Email);
        cmd.Parameters.AddWithValue("@prmTelefono", objOdontologo.Telefono);
        cmd.Parameters.AddWithValue("@prmEstado", objOdontologo.Estado);
        cmd.Parameters.AddWithValue("@prmNombreEspecialidad", objOdontologo.Especialidad.Nombre);

        con.Open();

        int filas = cmd.ExecuteNonQuery();
        if (filas > 0) response = true;
    }
    catch (Exception ex)
    {
        response = false;
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return response;
}

```

Gráfico 3.50. Método para registrar odontólogo

Elaborado por: El investigador

Método para listar odontólogos.

```

public List<Odontologo> ListarOdontologos()
{
    List<Odontologo> Lista = new List<Odontologo>();
    MySqlConnection con = null;
    MySqlCommand cmd = null;
    MySqlDataReader dr = null;

    try
    {
        con = Conexion.getInstance().ConexionDB();
        cmd = new MySqlCommand("spListarOdontologos", con);
        cmd.CommandType = CommandType.StoredProcedure;
        con.Open();
        dr = cmd.ExecuteReader();

        while (dr.Read())
        {
            //Crear objeto de tipo Odontologo
            Odontologo objOdontologo = new Odontologo();
            objOdontologo.Idodontologo = Convert.ToInt32(dr["Id_Odo"].ToString());
            objOdontologo.Cedula = dr["Ced_Emp"].ToString();
            objOdontologo.Nombre1 = dr["Nom1_Emp"].ToString();
            objOdontologo.Nombre2 = dr["Nom2_Emp"].ToString();
            objOdontologo.ApellidoPat = dr["ApPat_Emp"].ToString();
            objOdontologo.ApellidoMat = dr["ApMat_Emp"].ToString();
            objOdontologo.Email = dr["Email_Emp"].ToString();
            objOdontologo.Telefono = dr["Telefono_Emp"].ToString();
            objOdontologo.Estado = true;
            objOdontologo.Especialidad = new Especialidad()
            {
                Nombre = dr["Nom_Esp"].ToString()
            };
            //AÑadir a la lista de objetos
            Lista.Add(objOdontologo);
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return Lista;
}

```

Gráfico 3.51. Método para listar odontólogos

Elaborado por: El investigador

Método para eliminar odontólogo.

```
public bool EliminarOdontologo(string cedula)
{
    MySqlConnection con = null;
    MySqlCommand cmd = null;
    bool response = false;
    try
    {
        con = Conexion.getInstance().ConexionDB();
        cmd = new MySqlCommand("spEliminarOdontologo", con);
        cmd.Parameters.AddWithValue("@prmCedula", cedula);
        cmd.CommandType = CommandType.StoredProcedure;

        con.Open();

        cmd.ExecuteNonQuery();

        response = true;
    }
    catch (Exception ex)
    {
        response = false;
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return response;
}
```

Gráfico 3.52. Método para eliminar odontólogo

Elaborado por: El investigador

Método para actualizar odontólogo.

```
public bool ActualizarOdontologo(Odontologo objOdontologo)
{
    bool ok = false;
    MySqlConnection con = null;
    MySqlCommand cmd = null;

    try
    {
        con = Conexion.getInstance().ConexionDB();
        cmd = new MySqlCommand("spActualizarOdontologo", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@prmCedula", objOdontologo.Cedula);
        cmd.Parameters.AddWithValue("@prmEmail", objOdontologo.Email);
        cmd.Parameters.AddWithValue("@prmTelefono", objOdontologo.Telefono);

        con.Open();

        cmd.ExecuteNonQuery();

        ok = true;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
    }
    return ok;
}
```

Gráfico 3.53. Método para actualizar odontólogo

Elaborado por: El investigador

- PacienteDAO

Se codifico los métodos de registrar, eliminar, listar, buscar y actualizar.

Método para registrar paciente.

```

public bool RegistrarPaciente(Paciente objPaciente)
{
    SqlConnection con = null;
    SqlCommand cmd = null;
    bool response = false;

    try
    {
        con = Conexion.GetInstance().ConexionDB();
        cmd = new SqlCommand("spRegistrarPaciente", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@prmCedula", objPaciente.Cedula);
        cmd.Parameters.AddWithValue("@prmNombre1", objPaciente.Nombre1);
        cmd.Parameters.AddWithValue("@prmNombre2", objPaciente.Nombre2);
        cmd.Parameters.AddWithValue("@prmApellidoPat", objPaciente.ApellidoPaterno);
        cmd.Parameters.AddWithValue("@prmApellidoMat", objPaciente.ApellidoMaterno);
        cmd.Parameters.AddWithValue("@prmFechaNacimiento", objPaciente.FechaNacimiento);
        cmd.Parameters.AddWithValue("@prmEdad", objPaciente.Edad);
        cmd.Parameters.AddWithValue("@prmGenero", objPaciente.Genero);
        cmd.Parameters.AddWithValue("@prmTipoSangre", objPaciente.TipoSangre);
        cmd.Parameters.AddWithValue("@prmEmail", objPaciente.Email);
        cmd.Parameters.AddWithValue("@prmDireccion", objPaciente.Direccion);
        cmd.Parameters.AddWithValue("@prmTelefono", objPaciente.Telefono);
        cmd.Parameters.AddWithValue("@prmEstado", objPaciente.Estado);
        con.Open();

        int filas = cmd.ExecuteNonQuery();
        if (filas > 0) response = true;
    }
    catch (Exception ex)
    {
        response = false;
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return response;
}

```

Gráfico 3.54. Método para registrar paciente

Elaborado por: El investigador

Método eliminar paciente por cedula de identidad.

```

public bool EliminarPaciente(string cedula)
{
    SqlConnection con = null;
    SqlCommand cmd = null;
    bool ok = false;
    try
    {
        con = Conexion.GetInstance().ConexionDB();
        cmd = new SqlCommand("spEliminarPaciente", con);
        cmd.Parameters.AddWithValue("@prmCedula", cedula);
        cmd.CommandType = CommandType.StoredProcedure;

        con.Open();

        cmd.ExecuteNonQuery();

        ok = true;
    }
    catch (Exception ex)
    {
        ok = false;
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return ok;
}

```

Gráfico 3.55. Método para eliminar paciente por cedula de identidad

Elaborado por: El investigador

Método listar pacientes activos.

```
public List <Paciente> ListaPacientes()
{
    List<Paciente> Lista = new List<Paciente>();
    SqlConnection con = null;
    SqlCommand cmd = null;
    SqlDataReader dr = null;

    try
    {
        con = Conexion.GetInstance().ConexionDB();
        cmd = new SqlCommand("spListarPacientes", con);
        cmd.CommandType = CommandType.StoredProcedure;
        con.Open();
        dr = cmd.ExecuteReader();

        while (dr.Read())
        {
            //Crear objetos de tipo Paciente
            Paciente objPaciente = new Paciente();
            objPaciente.ID = Convert.ToInt32(dr["Id_Pac"].ToString());
            objPaciente.Cedula = dr["Cedula_Pac"].ToString();
            objPaciente.Nombre1 = dr["Nombre1_Pac"].ToString();
            objPaciente.Nombre2 = dr["Nombre2_Pac"].ToString();
            objPaciente.ApellidoPaterno = dr["ApePaterno_Pac"].ToString();
            objPaciente.ApellidoMaterno = dr["ApeMaterno_Pac"].ToString();
            objPaciente.FechaNacimiento = Convert.ToDateTime(dr["FechaNacimiento_Pac"].ToString());
            objPaciente.Edad = Convert.ToInt32(dr["Edad_Pac"].ToString());
            objPaciente.Genero = dr["Gen_Pac"].ToString();
            objPaciente.TipoSangre = dr["TipoSangre_Pac"].ToString();
            objPaciente.Email = dr["Email_Pac"].ToString();
            objPaciente.Direccion = dr["Direccion_Pac"].ToString();
            objPaciente.Telefono = dr["Telefono_Pac"].ToString();
            objPaciente.Estado = true;

            //Añadir a la lista de objetos
            Lista.Add(objPaciente);
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return Lista;
}
```

Gráfico 3.56. Método para listar pacientes activos

Elaborado por: El investigador

Método actualizar paciente.

```
public bool ActualizarPaciente(Paciente objPaciente)
{
    bool ok = false;
    SqlConnection con = null;
    SqlCommand cmd = null;

    try
    {
        con = Conexion.GetInstance().ConexionDB();
        cmd = new SqlCommand("spActualizarPaciente", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@prmCedula", objPaciente.Cedula);
        cmd.Parameters.AddWithValue("@prmEmail", objPaciente.Email);
        cmd.Parameters.AddWithValue("@prmTelefono", objPaciente.Telefono);
        cmd.Parameters.AddWithValue("@prmDireccion", objPaciente.Direccion);

        con.Open();

        cmd.ExecuteNonQuery();
        ok = true;
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return ok;
}
```

Gráfico 3.57. Método para actualizar paciente

Elaborado por: El investigador

Método buscar paciente por cedula de identidad.

```
public Paciente BuscarPacienteCedula(string cedula)
{
    MySqlConnection con = null;
    MySqlCommand cmd = null;
    MySqlDataReader dr = null;
    Paciente objPaciente = null;

    try
    {
        con = Conexion.getInstance().ConexionDB();
        cmd = new MySqlCommand("spBuscarPacienteCedula", con);
        cmd.Parameters.AddWithValue("@prmCedula", cedula);
        cmd.CommandType = CommandType.StoredProcedure;

        con.Open();
        dr = cmd.ExecuteReader();

        if (dr.Read())
        {
            objPaciente = new Paciente()
            {
                ID = Convert.ToInt32(dr["Id_Pac"].ToString()),
                Nombre1 = dr["Nombre1_Pac"].ToString(),
                Nombre2 = dr["Nombre2_Pac"].ToString(),
                ApellidoPaterno = dr["ApePaterno_Pac"].ToString(),
                ApellidoMaterno = dr["ApeMaterno_Pac"].ToString(),
                Telefono = dr["Telefono_Pac"].ToString(),
                Edad = Convert.ToInt32(dr["Edad_Pac"].ToString()),
                Genero = dr["Gen_Pac"].ToString()
            };
        }
    }
    catch (Exception ex)
    {
        objPaciente = null;
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return objPaciente;
}
```

Gráfico 3.58. Método para buscar paciente por cedula de identidad

Elaborado por: El investigador

Método buscar paciente por id de la Cita.

```
public Paciente BuscarPacienteIdCita(Int32 IdCita)
{
    MySqlConnection con = null;
    MySqlCommand cmd = null;
    MySqlDataReader dr = null;
    Paciente objPaciente = null;

    try
    {
        con = Conexion.getInstance().ConexionDB();
        cmd = new MySqlCommand("spBuscarPacienteIdCita", con);
        cmd.Parameters.AddWithValue("@prmIdCita", IdCita);
        cmd.CommandType = CommandType.StoredProcedure;

        con.Open();
        dr = cmd.ExecuteReader();

        if (dr.Read())
        {
            objPaciente = new Paciente()
            {
                ID = Convert.ToInt32(dr["Id_Pac"].ToString()),
                Nombre1 = dr["Nombre1_Pac"].ToString(),
                Nombre2 = dr["Nombre2_Pac"].ToString(),
                ApellidoPaterno = dr["ApePaterno_Pac"].ToString(),
                ApellidoMaterno = dr["ApeMaterno_Pac"].ToString(),
                Edad = Convert.ToInt32(dr["Edad_Pac"].ToString()),
                Genero = dr["gen_Pac"].ToString()
            };
        }
    }
    catch (Exception ex)
    {
        objPaciente = null;
        throw ex;
    }
    finally
    {
        con.Close();
    }
    return objPaciente;
}
```

Gráfico 3.59. Método para buscar paciente por ide de la cita

Elaborado por: El investigador

- RecuperarPasswordDAO

En esta clase tenemos los atributos y métodos para recuperar la contraseña mediante un correo.

```
//Atributos
private SmtplibClient smtpClient;

3 referencias
protected string senderMail { get; set; }
2 referencias
protected string password { get; set; }
2 referencias
protected string host { get; set; }
2 referencias
protected int port { get; set; }
2 referencias
protected bool ssl { get; set; }
```

Gráfico 3.60. Atributos de la clase RecuperarPasswordDAO

Elaborado por: El investigador

```
protected void initializeSmtplibClient()
{
    smtpClient = new SmtplibClient();
    smtpClient.Credentials = new NetworkCredential(senderMail, password);
    smtpClient.Host = host;
    smtpClient.Port = port;
    smtpClient.EnableSsl = ssl;
}
```

Gráfico 3.61. Método para inicializar el cliente SMTP

Elaborado por: El investigador

```
public void sendMail(string subject, string body, List<string> recipientMail)
{
    var mailMessage = new MailMessage();
    try
    {
        mailMessage.From = new MailAddress(senderMail);
        foreach (string mail in recipientMail)
        {
            mailMessage.To.Add(mail);
        }
        mailMessage.Subject = subject;
        mailMessage.Body = body;
        mailMessage.Priority = MailPriority.Normal;
        smtpClient.Send(mailMessage); //Enviar mensaje
    }
    catch (Exception ex)
    {
        throw ex;
    }
    finally
    {
        mailMessage.Dispose();
        smtpClient.Dispose();
    }
}
```

Gráfico 3.62. Método de la estructura del correo

Elaborado por: El investigador

```

public string recoverPassword(string userRequesting)
{
    using (var con = Conexion.GetInstance().ConexionDB())
    {
        con.Open();

        using (var cmd = new SqlCommand())
        {
            cmd.Connection = con;
            cmd.CommandText = "spRecuperarPassword";
            cmd.Parameters.AddWithValue("@prmEmail", userRequesting);
            cmd.CommandType = CommandType.StoredProcedure;
            SqlDataReader dr = cmd.ExecuteReader();

            if (dr.Read())
            {
                string name = dr["Nom1_Emp"].ToString() + " " + dr["ApPat_Emp"].ToString();
                string userMail = dr["Email_Emp"].ToString();
                string accountPassword = dr["Pass"].ToString();

                var mailService = new SupportMail();
                mailService.sendMail(
                    subject: "SISTEMA: Solicitud de recuperación de contraseña",
                    body: "Hola, " + name + "\nUsted solicitó recuperar su contraseña.\n" +
                        "Su contraseña actual es: " + accountPassword,
                    recipientMail: new List<string> { userMail }
                );

                return "Hola, " + name + "\nUsted solicitó recuperar su contraseña.\n" +
                    "Por favor revise su correo: " + userMail;
            }
            else
            {
                return "Lo sentimos, no tiene una cuenta con ese correo o nombre de usuario";
            }
        }
    }
}

```

Gráfico 3.63. Método para recuperar la contraseña por correo

Elaborado por: El investigador

- SupportMail

En esta clase enviamos por default los atributos heredados de la clase RecuperarPasswordDAO.

```

2 referencias
internal class SupportMail: RecuperarPasswordDAO
{
    1 referencia
    public SupportMail()
    {
        senderMail = "dental.sis2611@gmail.com";
        password = "tvfngybsoxhuawdd";
        host = "smtp.gmail.com";
        port = 587;
        ssl = true;
        initializeSmtpClient();
    }
}

```

Gráfico 3.64. Envío de los atributos heredados de RecuperarPasswordDAO.

Elaborado por: El investigador

3. Planteamiento de la Lógica de Negocio

En esta capa se definieron las responsabilidades(métodos) implementados en el sistema y también se implementó el patrón Singleton. En esta capa se instanciaron los métodos que se desarrollaron en la capa de Acceso a Datos.

- CitaLN

Método para registrar, listar y actualizar cita.

```
public class CitaLN
{
    "PATRON SINGLETON"

    1 referencia
    public bool RegistrarCita(Cita objCita)
    {
        try
        {
            return CitaDAO.getInstance().RegistrarCita(objCita);
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }

    1 referencia
    public List<Cita> ListarCitas()...

    2 referencias
    public bool ActualizarCita(Int32 idCita, string estado)...
```

Gráfico 3.65. Clase de lógica de negocio de Cita

Elaborado por: El investigador

- DiagnosticoLN

Método para registrar diagnóstico.

```
public class DiagnosticoLN
{
    "PATRON SINGLETON"

    1 referencia
    public bool RegistrarDiagnostico(Diagnostico objDiagnostico)
    {
        try
        {
            return DiagnosticoDAO.getInstance().RegistrarDiagnostico(objDiagnostico);
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
}
```

Gráfico 3.66. Clase de lógica de negocio de Diagnostico

Elaborado por: El investigador

- EspecialidadLN

Método para listar especialidades.

```
public class EspecialidadLN
{
    "PATRON SINGLETON"

    1 referencia
    public List<Especialidad> ListarEspecialidades()
    {
        try
        {
            return EspecialidadesDAO.getInstance().ListarEspecialidades();
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
}
```

Gráfico 3.67. Clase de lógica de negocio de Especialidad

Elaborado por: El investigador

- HorarioAtencionLN

Método para registrar, listar, eliminar y actualizar horarios de atención.

```
public class HorarioAtencionLN
{
    "PATRON SINGLETON"
    1 referencia
    public HorarioAtencion RegistrarHorarioAtencion(HorarioAtencion objHorarioAtencion)
    {
        try
        {
            return HorarioAtencionDAO.getInstance().RegistrarHorarioAtencion(objHorarioAtencion);
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }

    1 referencia
    public List<HorarioAtencion> ListarHorariosAtencion(Int32 IdOdontologo)...

    1 referencia
    public List<HorarioAtencion> ListarHorarioReserva(Int32 IdEspecialidad, DateTime Fecha)...

    1 referencia
    public bool EliminarHorarioAtencion(int IdHorarioAtencion)...

    1 referencia
    public bool ActualizarHorarioAtencion(HorarioAtencion objHorario)...
}
```

Gráfico 3.68. Clase de lógica de negocio de HorarioAtencion

Elaborado por: El investigador

- IngresoLN

Método para Acceder al Sistema y obtener datos del usuario.

```
public class IngresoLN
{
    "PATRON SINGLETON"
    2 referencias
    public IngresoSistema AccesoSistema(String correo, String clave)
    {
        try
        {
            return IngresoDAO.getInstance().AccesoSistema(correo, clave);
        }
        catch(Exception ex)
        {
            throw ex;
        }
    }

    1 referencia
    public IngresoSistema NombreOdontologo(Int32 idEmpleado)...
}
```

Gráfico 3.69. Clase de lógica de negocio de Ingreso

Elaborado por: El investigador

- OdontologoLN

Método para buscar, registrar, eliminar y actualizar odontólogos.

```

public class OdontologoLN
{
    "PATRON SINGLETON"

    1 referencia
    public Odontologo BuscarOdontologo(String cedula)
    {
        try
        {
            return OdontologoDAO.getInstance().BuscarOdontologo(cedula);
        }
        catch(Exception ex)
        {
            throw ex;
        }
    }

    1 referencia
    public bool RegistrarOdontologo(Odontologo objOdontologo)...
    1 referencia
    public List<Odontologo> ListarOdontologos(...

    1 referencia
    public bool EliminarOdontologo(string cedula)...

    1 referencia
    public bool ActualizarOdontologo(Odontologo objOdontologo)...
}

```

Gráfico 3.70. Clase de lógica de negocio de Odontólogo

Elaborado por: El investigador

- PacienteLN

Método para registrar, eliminar, listar, actualizar, buscar pacientes por cedula e id de la cita.

```

public class PacienteLN
{
    "PATRON SINGLETON"

    1 referencia
    public bool RegistrarPaciente(Paciente objPaciente)
    {
        try
        {
            return PacienteDAO.getInstance().RegistrarPaciente(objPaciente);
        }
        catch(Exception ex)
        {
            throw ex;
        }
    }

    1 referencia
    public bool EliminarPaciente(string cedula)...

    1 referencia
    public List<Paciente> ListaPacientes(...

    1 referencia
    public bool ActualizarPaciente(Paciente objPaciente)...

    1 referencia
    public Paciente BuscarPacienteCedula(string cedula)...

    1 referencia
    public Paciente BuscarPacienteIdCita(Int32 IdCita)...
}

```

Gráfico 3.71. Clase de lógica de negocio de Paciente

Elaborado por: El investigador

- recoverPasswordLN

Método para recuperar la contraseña del usuario.

```

public class recoverPasswordLN
{
    RecuperarPasswordDAO recuperardao = new RecuperarPasswordDAO();

    1 referencia
    public string recoverPassword(string userRequesting)
    {
        return recuperardao.recoverPassword(userRequesting);
    }
}

```

Gráfico 3.72. Clase de lógica de negocio de recoverPassword

Elaborado por: El investigador

3.2.3.4. Desarrollo Front-end

Desarrollo de la capa de presentación

En esta capa se realizaron las peticiones Ajax, las cuales se implementaron para crear y llenar datos en las tablas, también los eventos de los botones utilizados en los formularios y modales. Además, se realizaron las validaciones de solo números y letras.

```

//Validar que solo se ingresen numeros
function validateNumericInput(event) {
    var charCode = (event.which) ? event.which : event.keyCode;
    if (charCode > 31 && (charCode < 48 || charCode > 57)) {
        return false;
    }
    return true;
}

```

Gráfico 3.73. Validación de solo ingreso de números

Elaborado por: El investigador

```

//Validar Texto Mayusculas
function validarTexto(input) {
    var regex = /^[A-Za-záéíóúÁÉÍÓÚñÑ\s]+$/;
    if (!regex.test(input.value)) {
        input.value = '';
    } else {
        input.value = input.value.toUpperCase();
    }
}

```

Gráfico 3.74. Validación de solo ingreso de letras mayúsculas

Elaborado por: El investigador

- HorarioOdontologo.js

```

//Configuracion de timepicker y date
$("#[data-mask]").inputmask();
$("#.timepicker").timepicker(...);

var tabla;

function initDateTable(...){
    initDateTable();

    //Buscar Odontogo por cedula
    $("#btnBuscar").on(...);

    function llenarDatosOdontologo(obj){...

        //Agregar un Horario
        $("#btnAgregar").on(...);

        //Agregar filas a la tabla
        function addRow(obj){...

            //Formato de fecha
            function formatDate(date){...

                //Listar Horarios
                function listHorarios(IdOdontologo){...

                    //Evento click para eliminar registros
                    $(document).on(...);

                    function deleteDataAjax(data){...

                        //Evento click para actualizar registros
                        $(document).on(...);

                        function llenarDatosHorario(data){...

                            //Eliminar Horario
                            $("#btnEditar").click(...);
    
```

Gráfico 3.75. Archivo Javascript de HorarioOdontologo

Elaborado por: El investigador

- Paciente.js

```

//Configuracion de date
$("#[data-mask]").inputmask();

var tabla, data;

// funcion para añadir fila de pacientes a la tabla
function addRowDT(data){...

//AJAX LISTAR, ACTUALIZAR, ELIMINAR PACIENTE
function sendDataAjax(...){...

function updateDataAjax(...){...

function deleteDataAjax(data){...

//Evento click para actualizar registros
$(document).on(...);

//Evento click para eliminar registros
$(document).on(...);

//Cargar Datos en el Modal
function fillModalData(...){...

//Enviar la informacion al servidor
$("#btnActualizar").click(...);

// Llamando a la funcion de ajax al cargar el documento
sendDataAjax();
    
```

Gráfico 3.76. Archivo Javascript de Paciente

Elaborado por: El investigador

- Reserva.js

```
//Configuracion de date
$("#[data-mask]").inputmask();

//Evento click para buscar paciente
$("#btnBuscar").on(...

//Funcion para buscar pacientes por cedula
function searchPacienteCedula(cedula)...

function llenarDatosPaciente(obj)...

function limpiarDatosPaciente()...
```

Gráfico 3.77. Archivo Javascript de Reserva

Elaborado por: El investigador

- Odontologo.js

```
var tabla, data;

// funcion para añadir fila de odontologos a la tabla
function addRowDT(data)...

//AJAX LISTAR, ACTUALIZAR, ELIMINAR PACIENTE
function sendDataAjax(...

function updateDataAjax(...

function deleteDataAjax(data)...

//Evento click para actualizar registros
$(document).on(...;

//Evento click para eliminar registros
$(document).on(...;

//Cargar Datos en el Modal
function fillModalData(...

//Enviar la informacion al servidor
$("#btnActualizar").click(...;

// Llamando a la funcion de ajax al cargar el documento
sendDataAjax();
```

Gráfico 3.78. Archivo Javascript de Odontólogo

Elaborado por: El investigador

- logout.js

```
//Evento clic para salir
$(document).on('click', '#close', function (e){
    e.preventDefault();

    $.ajax({
        type: "POST",
        url: "PanelGeneral.aspx/Logout",
        dataType: "json",
        contentType: 'application/json; charset=utf-8',
        error: function (xhr, ajaxOptions, thrownError) {
            console.log(xhr.status + " \n" + xhr.responseText, "\n" + thrownError);
        },
        success: function (response) {
            if (response.d == true) {
                window.location.href = "InicioSesion.aspx";
            }
        }
    });
});
```

Gráfico 3.79. Archivo Javascript de Logout

Elaborado por: El investigador

3.2.4. Fase IV: Pruebas

3.2.4.1. Pruebas de aceptación

Las pruebas de aceptación se efectúan con el propósito de determinar la validez de la aplicación web basándose en los requerimientos descritos en las historias de usuario.

Prueba de aceptación	
Número: 1	Historia de Usuario: 1
Nombre: Ingreso al sistema	
Descripción: Se ingresa a la aplicación con el correo y contraseña del usuario para validar y permitir el acceso.	
Condiciones de ejecución: El usuario debe estar registrado en la base de datos y contar con un correo y contraseña.	
Entrada: <ul style="list-style-type: none">• El usuario ingresa su correo y contraseña.• Clic en el botón Ingresar.	
Resultado esperado: Ingreso exitoso al sistema con las credenciales (correo, contraseña) del usuario.	
Evaluación de prueba: Satisfactoria	

Tabla 3.49. Prueba de Aceptación 1 - Ingreso al sistema

Elaborado por: El investigador

Prueba de aceptación	
Número: 2	Historia de Usuario: 1
Nombre: Ingreso al sistema	
Descripción: Se ingresa a la aplicación con el correo y contraseña del usuario que no se encuentra registrado en la base de datos.	
Condiciones de ejecución: El usuario debe estar registrado en la base de datos y contar con un correo y contraseña.	
Entrada: <ul style="list-style-type: none">• El usuario ingresa su correo y contraseña.• Clic en el botón Ingresar.	
Resultado esperado: Se muestra una alerta de que el usuario y/o contraseña son incorrectos.	
Evaluación de prueba: Satisfactoria	

Tabla 3.50. Prueba de Aceptación 2 - Ingreso al sistema

Elaborado por: El investigador

Prueba de aceptación	
Número: 3	Historia de Usuario: 2
Nombre: Recuperar contraseña	
Descripción: El usuario puede recuperar su contraseña ingresando su correo, en el cual le llegará un mensaje con la contraseña de ingreso.	
Condiciones de ejecución: Existe un único usuario que puede recuperar la contraseña de acceso.	
Entrada: <ul style="list-style-type: none"> • El usuario ingresa su correo. • Clic en el botón Enviar Correo. 	
Resultado esperado: En la misma ventana se le visualiza un mensaje.	
Evaluación de prueba: Satisfactoria	

Tabla 3.51. Prueba de Aceptación 3 - Recuperar contraseña

Elaborado por: El investigador

Prueba de aceptación	
Número: 4	Historia de Usuario: 3
Nombre: Página Principal	
Descripción: En la interfaz de página principal se visualiza el logo de la institución, un menú lateral con los módulos requeridos, la opción de cerrar sesión, anuncio de bienvenida, la imagen y nombre del usuario activo.	
Condiciones de ejecución: Ninguna.	
Entrada: <ul style="list-style-type: none"> • Se muestra información del usuario y módulos. 	
Resultado esperado: Página principal antes descrita.	
Evaluación de prueba: Satisfactoria	

Tabla 3.52. Prueba de Aceptación 4 - Página Principal

Elaborado por: El investigador

Prueba de aceptación	
Número: 5	Historia de Usuario: 4
Nombre: Gestión Pacientes	
Descripción: El usuario tiene la capacidad de registrar, actualizar o eliminar datos del paciente en función de su disponibilidad. Además, el sistema cuenta con una función de búsqueda que facilita la localización de los pacientes.	
Condiciones de ejecución: El usuario encargado de la gestión de pacientes es el gerente de la institución.	
Entrada: <ul style="list-style-type: none"> • Completar los campos requeridos para el registro del paciente. • En la tabla de los pacientes en cada fila existe la posibilidad de actualizar o eliminar. • El botón actualizar muestra un modal de los campos modificables. 	
Resultado esperado: Manejo efectivo de la información de los pacientes.	
Evaluación de prueba: Satisfactoria	

Tabla 3.53. Prueba de Aceptación 5 - Gestión Pacientes

Elaborado por: El investigador

Prueba de aceptación	
Número: 6	Historia de Usuario: 5
Nombre: Gestión Horarios de Atención	
Descripción: El usuario puede agregar, eliminar o actualizar los horarios de los odontólogos.	
Condiciones de ejecución: El usuario encargado de la creación de horarios es el gerente de la institución.	
Entrada: <ul style="list-style-type: none"> • El usuario realiza una búsqueda por cedula del odontólogo. • En cada fila de los horarios se visualiza los botones de actualizar y eliminar. • Bajo la tabla de horarios hay la opción de agregar uno nuevo. 	
Resultado esperado: Manejo efectivo de los horarios de atención de cada odontólogo.	
Evaluación de prueba: Satisfactoria	

Tabla 3.54. Prueba de Aceptación 6 - Gestión Horarios de Atención

Elaborado por: El investigador

Prueba de aceptación	
Número: 7	Historia de Usuario: 6
Nombre: Reserva de Citas	
Descripción: El usuario puede buscar un horario de atención disponible con la fecha y especialidad requerida.	
Condiciones de ejecución: El usuario encargado de la reserva de citas es el gerente de la institución.	
Entrada:	
<ul style="list-style-type: none"> • El usuario debe ingresar la cedula del paciente. • Es necesario el ingreso de fecha y selección de especialidad para que se visualice los horarios disponibles. • El usuario debe marcar el checkbox (casilla de verificación) para seleccionar el horario y poder reservar la cita. 	
Resultado esperado: Manejo efectivo de la reserva de citas.	
Evaluación de prueba: Satisfactoria	

Tabla 3.55. Prueba de Aceptación 7 - Reserva de Citas

Elaborado por: El investigador

Prueba de aceptación	
Número: 8	Historia de Usuario: 7
Nombre: Gestión Atención Pacientes	
Descripción: Se visualiza una lista de las citas pendientes de atender.	
Condiciones de ejecución: El usuario encargado de la atención de citas es el gerente de la institución.	
Entrada:	
<ul style="list-style-type: none"> • El usuario puede realizar la atención como cancelar. • Al momento que se cancela se visualiza una alerta para confirmar. • Al realizar la atención se muestra un formulario en el cual el usuario 	
Resultado esperado:	
Evaluación de prueba: Satisfactoria	

Tabla 3.56. Prueba de Aceptación 8 - Gestión Atención Pacientes

Elaborado por: El investigador

Prueba de aceptación	
Número: 9	Historia de Usuario: 8
Nombre: Gestión Odontólogos	
Descripción: El usuario tiene la capacidad de registrar, actualizar o eliminar datos de los odontólogos en función de su disponibilidad. Además, el sistema cuenta con una función de búsqueda que facilita la localización de los odontólogos.	
Condiciones de ejecución: El usuario encargado de la gestión de odontólogos es el gerente de la institución.	
Entrada: <ul style="list-style-type: none"> • Completar los campos requeridos para el registro del odontólogo. • En la tabla de los odontólogos en cada fila existe la posibilidad de actualizar o eliminar. • El botón actualizar muestra un modal de los campos modificables. 	
Resultado esperado: Manejo efectivo de la información de los pacientes.	
Evaluación de prueba: Satisfactoria	

Tabla 3.57. Prueba de Aceptación 9 - Gestión Odontólogos

Elaborado por: El investigador

Prueba de aceptación	
Número: 10	Historia de Usuario: 9
Nombre: Comprobante	
Descripción: En la interfaz de Comprobante se visualiza un formulario de los tratamientos que se realizó en la consulta el paciente.	
Condiciones de ejecución: El usuario encargado de llenar el comprobante es el gerente de la institución.	
Entrada: <ul style="list-style-type: none"> • El usuario debe ingresar la cedula del paciente, tratamiento y costo respectivo. • Al dar clic en el botón agregar se transfieren al DataGridView. 	
Resultado esperado: Los tratamientos y costos se visualizan en un DataGridView con el precio total de la consulta.	
Evaluación de prueba: Satisfactoria	

Tabla 3.58. Prueba de Aceptación 10 - Comprobante

Elaborado por: El investigador

Prueba de aceptación	
Número: 11	Historia de Usuario: 10
Nombre: Generar Pdf del Comprobante	
Descripción: Se genera y descarga un archivo pdf del comprobante.	
Condiciones de ejecución: Ninguna.	
Entrada:	
<ul style="list-style-type: none"> • El usuario deberá dar clic en el botón generar PDF y se descargará el archivo. 	
Resultado esperado: El archivo contiene dos copias del comprobante, la una copia para el centro odontológico y la otra para el paciente.	
Evaluación de prueba: Satisfactoria	

Tabla 3.59. Prueba de Aceptación 11 - Generar Pdf del Comprobante

Elaborado por: El investigador

Prueba de aceptación	
Número: 12	Historia de Usuario: 11
Nombre: Finalizar Sesión	
Descripción: El usuario tiene la opción de cerrar la sesión desde el menú superior derecho.	
Condiciones de ejecución: Una vez que el usuario ha cerrado la sesión, será necesario que inicie sesión nuevamente si desea acceder al sistema.	
Entrada:	
<ul style="list-style-type: none"> • Clic en el botón cerrar sesión. 	
Resultado esperado: Después de que el usuario cierre el sistema, será redirigido automáticamente a la página de Login.	
Evaluación de prueba: Satisfactoria	

Tabla 3.60. Prueba de Aceptación 12 - Finalizar Sesión

Elaborado por: El investigador

3.2.4.2. Pruebas de usabilidad

Se llevaron a cabo pruebas de usabilidad mediante la aplicación de un test con el objetivo de evaluar la facilidad de uso y experiencia del odontólogo principal que utilizará el sistema web. Se empleo la siguiente escala de Likert para evaluar el grado de satisfacción.

- Totalmente de Acuerdo
- De acuerdo
- Neutral
- En desacuerdo
- Totalmente en desacuerdo

A continuación, se presentan los resultados obtenidos de las pruebas realizadas al odontólogo del centro odontológico "DENTALSIS".

Resultados prueba de usabilidad	
Tipo/Rol de usuario: Odontólogo	Código: 1
El sistema le permite el registro de citas odontológicas en horarios disponibles de los odontólogos.	Totalmente de Acuerdo
La interfaz está diseñada de manera agradable y adecuada para su uso, tanto en términos de colores, botones, estructura y menú.	Totalmente de Acuerdo
El diseño de la interfaz es adaptable para cualquier tipo de dispositivo.	Totalmente de Acuerdo
En general, el manejo y uso del sistema es de fácil.	De acuerdo
La descarga de los documentos alojados en la aplicación es rápida y sencilla.	Totalmente de Acuerdo
Los distintos módulos se encuentran organizados y estructurados.	Totalmente de Acuerdo

Tabla 3.61. Prueba de Usabilidad

Elaborado por: El investigador

Con la prueba de usabilidad realizada al odontólogo principal se puede deducir que la aplicación web presenta una estructura y organización adecuada para evitar confusiones, es adaptable a todo tipo de dispositivo, el diseño es agradable para el usuario, la descarga de documentación se realiza de forma rápida y referente al manejo del sistema presenta una leve dificultad ya que el odontólogo no está tan familiarizado con el uso de sistemas en el consultorio.

3.2.5. Fase V: Implantación

Para implementación de la aplicación web, se realizó la adquisición de un servidor web en smarterasp.net con las siguientes características:

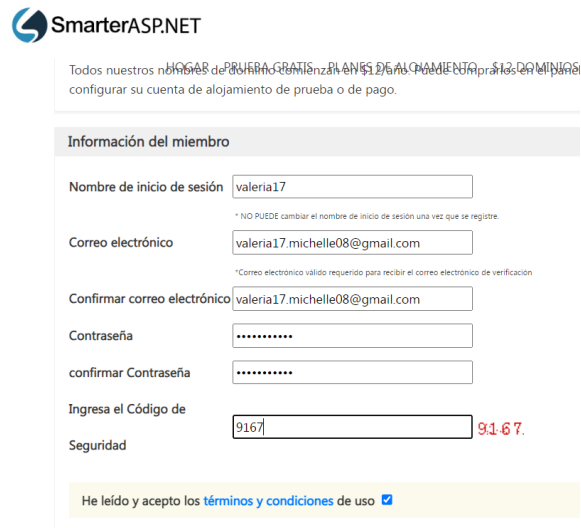
		
Alojamiento Windows Server 2022	1 base de datos MS SQL	1 base de datos MySQL
Sin publicidad	MS SQL 2019, 2017, 2016, 2014, 2012, 2008	Tamaño de la base de datos MySQL: 1GB
Capacidad de almacenamiento: 5GB	Tamaño de la base de datos MS SQL: 1GB	Copia de seguridad, restauración: SÍ
Ancho de banda: 10 GB/mes	Tamaño de registro de MS SQL: 1 GB	Conexión remota: SÍ
Sitios web: 1	Copia de seguridad, restauración: SÍ	phpMyAdmin: SÍ
ASP.NET, .Net Core, ASP, PHP	Conexión remota: SÍ	
¡MS Access y MÁS!	Estudio de gestión SQL: SÍ	

Gráfico 3.80. Características del servidor de Smarterasp.net

Elaborado por: Smarterasp.net

Smarterasp.net es una compañía que provee alojamiento gratuito de ASP.NET por 60 días completos antes de decidir si se desea adquirir el servidor web, además el dominio temporal es gratuito, las configuraciones y activación son instantáneas.

Proceso de Configuración



SmarterASP.NET

HOY ES BUENA OPORTUNIDAD PARA COMPRAR EL PLAN DE ALOJAMIENTO \$12 DOMINIOS TODOS NUESTROS NOMBRES DE DOMINIO COMIENZAN EN \$12.95. PUEDE COMPRARLOS EN EL PLAN de configuración de alojamiento de prueba o de pago.

Información del miembro

Nombre de inicio de sesión

Correo electrónico

Confirmar correo electrónico

Contraseña

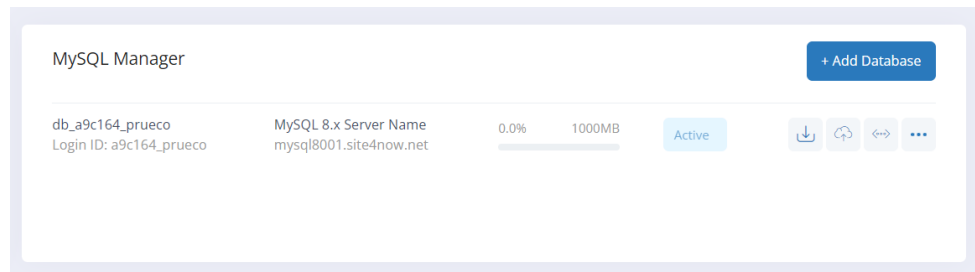
confirmar Contraseña

Ingresa el Código de Seguridad 91-67

He leído y acepto los [términos y condiciones de uso](#)

Gráfico 3.81. Creación de la cuenta en Smarterasp.net

Elaborado por: El investigador



MySQL Manager

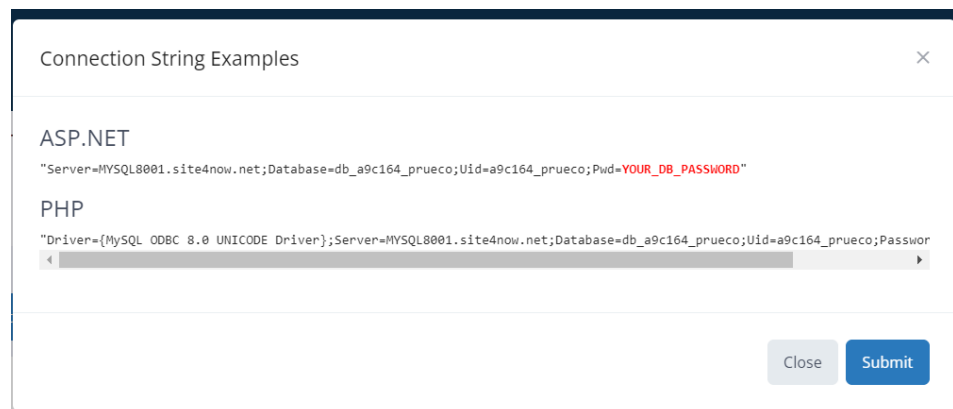
+ Add Database

db_a9c164_prueco MySQL 8.x Server Name 0.0% 1000MB Active

Login ID: a9c164_prueco mysql8001.site4now.net

Gráfico 3.82. Importación de la Base de Datos MySQL

Elaborado por: El investigador



Connection String Examples

ASP.NET

```
"Server=MYSQL8001.site4now.net;Database=db_a9c164_prueco;Uid=a9c164_prueco;Pwd=YOUR_DB_PASSWORD"
```

PHP

```
"Driver={MySQL ODBC 8.0 UNICODE Driver};Server=MYSQL8001.site4now.net;Database=db_a9c164_prueco;Uid=a9c164_prueco;Passwor"
```

Close Submit

Gráfico 3.83. Credenciales de la Base de Datos MySQL

Elaborado por: El investigador

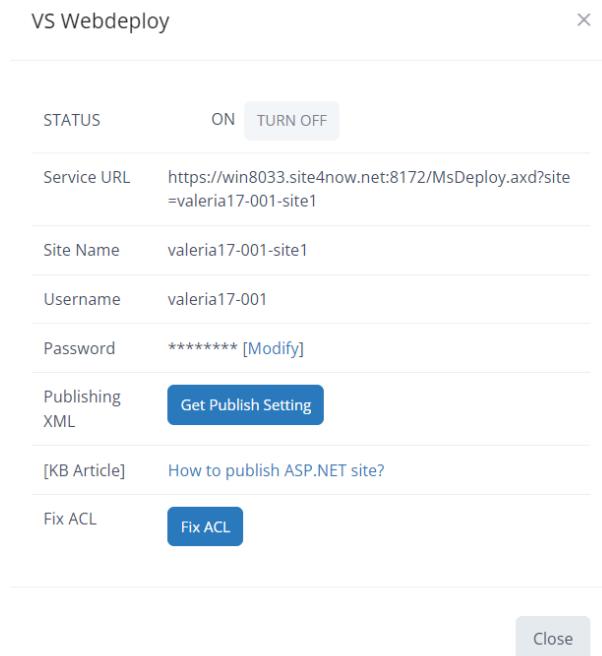


Gráfico 3.84. Credenciales del dominio

Elaborado por: El investigador

Referente a la base de datos se realizó la importación mediante phpMyAdmin y con las credenciales proporcionales se estableció una nueva cadena de conexión.

```
public MySqlConnection ConexionDB()
{
    MySqlConnection conexion = new MySqlConnection();
    conexion.ConnectionString = "Server=MYSQ18001.site4now.net;Database=db_a9c164_prueco;Uid=a9c164_prueco;Pwd=DentalSis2023";
    return conexion;
}
```

Gráfico 3.85. Cadena de conexión

Elaborado por: El investigador

Luego de configurar la cadena de conexión se procedió a publicar mediante la opción de Servidor Web.

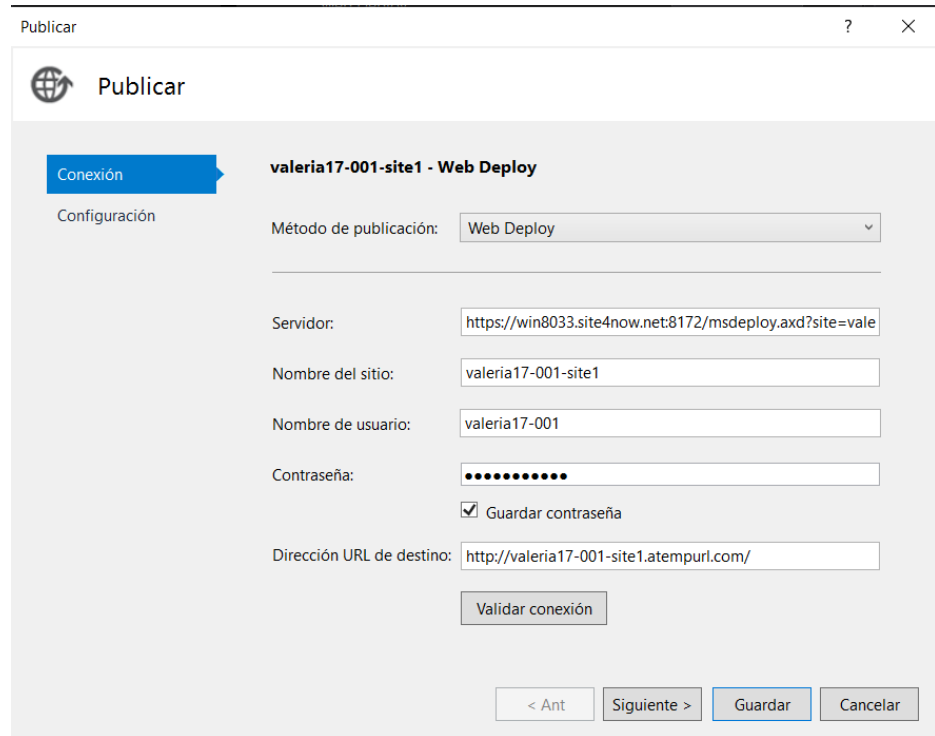


Gráfico 3.86. Configuraciones de la Conexión del Web Deploy

Elaborado por: El investigador

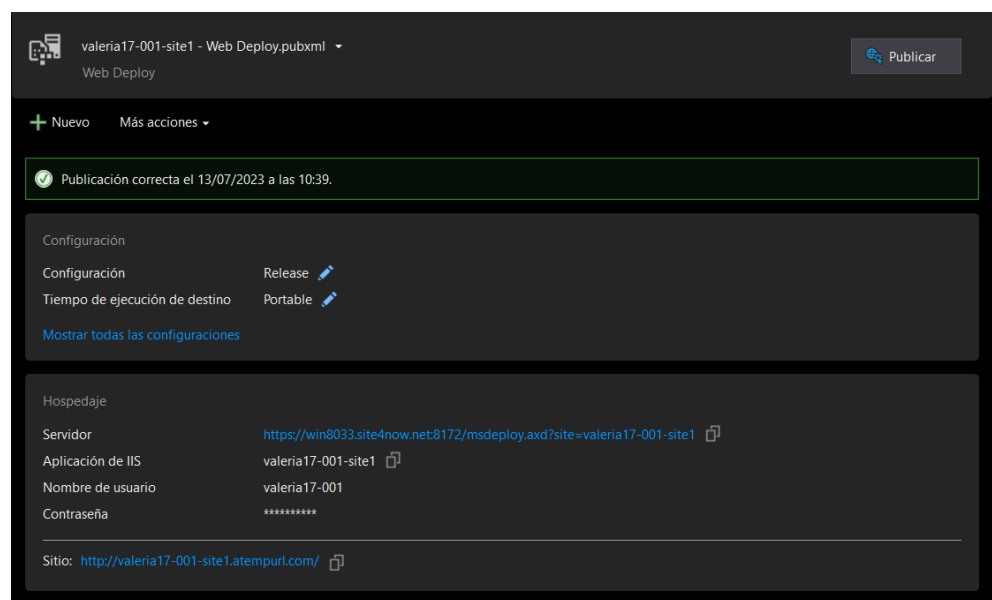


Gráfico 3.87. Publicación de la Aplicación web con servidor Smarterasp.net

Elaborado por: El investigador

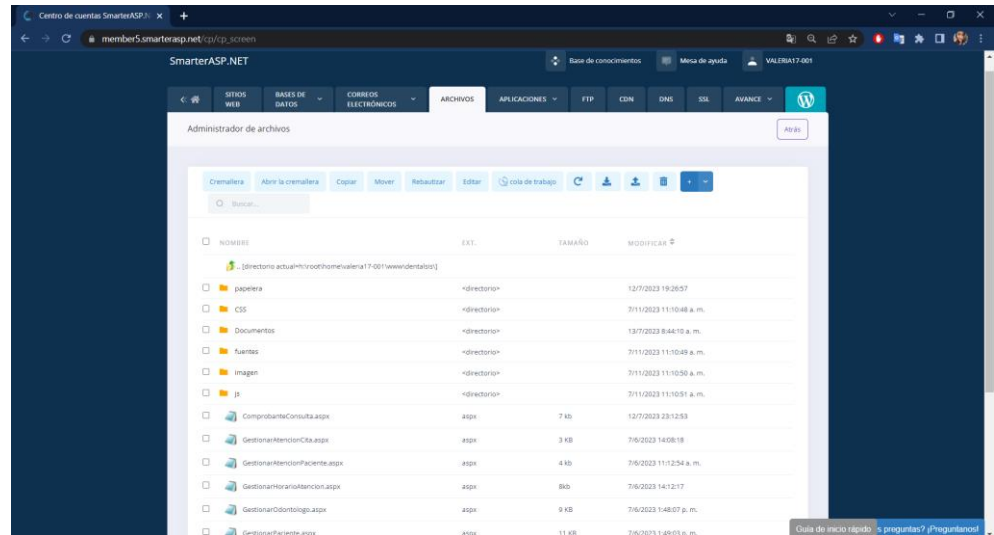


Gráfico 3.88. Archivos y carpetas publicados en el servidor Smarterasp.net

Elaborado por: El investigador

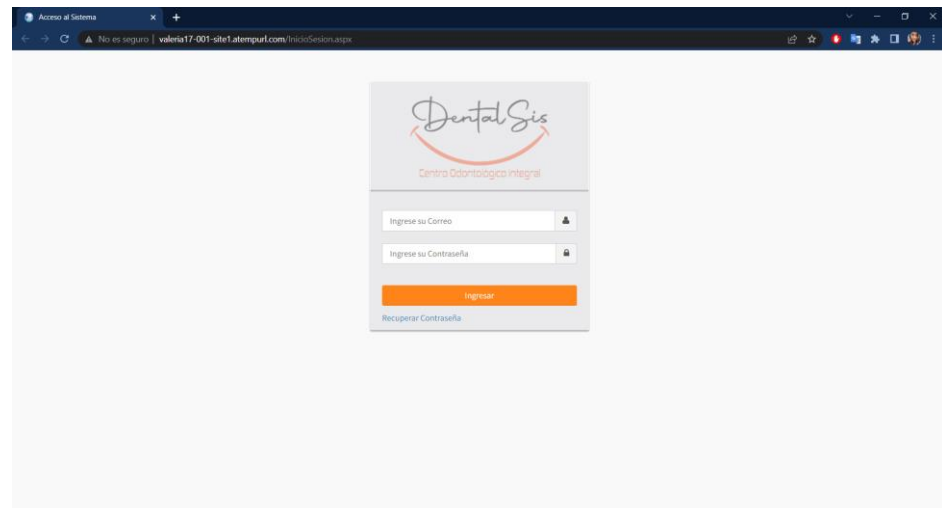


Gráfico 3.89. Aplicación implementada en el sitio web

Elaborado por: El investigador

Una vez finalizada la implementación de la aplicación web, se llevó a cabo un programa de capacitación dirigido al personal del Centro Odontológico “DENTALSIS”.

Tema	Instructor	Destinado a	Fecha de Inicio	Fecha de Fin	13/07/2023
Ingreso al Sistema	Valeria Logroño	Odontólogo principal	13/07/2023	13/07/2023	
Restablecer Contraseña	Valeria Logroño	Odontólogo principal	13/07/2023	13/07/2023	
Gestión Pacientes	Valeria Logroño	Odontólogo principal	13/07/2023	13/07/2023	
Gestión Odontólogos	Valeria Logroño	Odontólogo principal	13/07/2023	13/07/2023	
Gestión Horarios de Odontólogos	Valeria Logroño	Odontólogo principal	13/07/2023	13/07/2023	
Reserva de Citas	Valeria Logroño	Odontólogo principal	13/07/2023	13/07/2023	
Atención de Citas	Valeria Logroño	Odontólogo principal	13/07/2023	13/07/2023	
Comprobante	Valeria Logroño	Odontólogo principal	13/07/2023	13/07/2023	
Finalizar Sesión	Valeria Logroño	Odontólogo principal	13/07/2023	13/07/2023	

Tabla 3.62. Cronograma de capacitación referente a la aplicación web

Elaborado por: El investigador

Capítulo IV.- CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Luego de aplicar herramientas y técnicas de recolección de información, se determinó que la mayoría de pacientes están de acuerdo en la implementación de un sistema informático que mejore el proceso actual empleado en el Centro Odontológico “DENTALSIS”, ya que ayudara a evitar la pérdida de información de los pacientes y la duplicidad de las citas.
- La tecnología Responsive Web Design referente a la creación y desarrollo de aplicaciones provee una solución flexible y eficiente al brindar una experiencia más cómoda al usuario, debido a que la visualización del contenido y diseño se adapta a diferentes dispositivos y tamaños de pantalla.
- La metodología Extreme Programming (XP) demostró ser altamente eficiente al supervisar cada actividad del proyecto, por lo que facilitó la identificación de las funcionalidades de la aplicación mediante una planificación basada en historias de usuario. Además, en cada entrega de las iteraciones se llevaron a cabo retroalimentaciones que contribuyeron al cumplimiento de los requisitos solicitados por la empresa.
- La implementación de la aplicación mejoró la comunicación entre el odontólogo y el paciente a través de la innovación tecnológica, atendiendo las sugerencias de los usuarios con el fin de proporcionar un servicio de alta calidad. Además, esta aplicación interactiva cumple con los requisitos establecidos.

4.2. Recomendaciones

- Se recomienda utilizar herramientas y técnicas de recolección que aporten ideas claras acerca de las características que tendrá la aplicación. El empleo de escalas de Likert en encuestas evita la ambigüedad en las respuestas de los participantes lo cual facilita el proceso de recopilación y análisis de los datos, la elaboración de una guía de entrevista ayuda a organizar las preguntas de forma secuencial para obtener información relevante y específica acerca de la investigación.
- Para ahorrar tiempo en la creación de diseños desde cero, se sugiere utilizar plantillas de diseño de interfaz de usuario de Bootstrap con el fin de garantizar un aspecto visual más atractivo y profesional facilitando la adaptabilidad a diferentes dispositivos.
- Se recomienda establecer el alcance y tamaño del proyecto, debido a que la metodología XP se centra en proyectos complejos y que requieren adaptabilidad a los cambios en los requisitos. Sin embargo, las fases se acoplan de una manera versátil a proyectos sin importar el tamaño con el fin de proveer un desarrollo más flexible, eficiente y centrado en el cliente.
- Se sugiere al Centro Odontológico “DENTALSIS” seguir con el desarrollo de diferentes módulos con nuevas funcionalidades como el pago en línea, la generación online de fichas odontológicas, creación de proformas, facturación en línea y la aplicación de un servidor en la nube para almacenar los datos de los pacientes.

BIBLIOGRAFÍA

- [1] Pere Escorsa Castells and Jaume Valls Pasola, Tecnología e innovación en la empresa. 2003.
- [2] C. G. MORALES GONZALEZ, “Nivel de satisfacción de los pacientes que asisten al servicio de urgencias, frente a la atención de enfermería en una institución de primer nivel de atención en salud, en Mistrató Risaralda,” PONTIFICIA UNIVERSIDAD JAVERIANA, SANTAFE DE BOGOTA D.C.
- [3] A. Peña, M. Lisseth Herrera, S. Carrera, and D. Sánchez, “Indicadores de tecnología de la información y comunicación,” 2021. [Online]. Available: www.ecuadorencifras.gob.ec
- [4] D. Martínez and M. Álvarez, “Responsive Web Design,” Apr. 2020. Accessed: Jun. 12, 2022. [Online]. Available: <https://www.lawebdelprogramador.com/pdf/16484-Manual-de-Responsive-Web-Design.html>
- [5] L. B. Rodriguez Cardenas and C. C. Santillan Aban, “Sistema informático web de control de citas médicas para el área de rehabilitación física de la Clínica San Juan de Dios.”
- [6] E. A. NEYRA VALDIVIA, “Implementación del Sistema Web para la Gestión de Citas Horarias en el Hospital María Auxiliadora,” LIMA, PERU, 2017.
- [7] M. Yanitza *et al.*, “Aplicación del Responsive Web Design en la creación e implementación del sitio Web del Centro de Histoterapia Placentaria,” La Habana - Cuba, 2018. [Online]. Available: <http://scielo.sld.cu>
- [8] P. A. Calle Játiva, “DESARROLLO DEL SISTEMA WEB APLICANDO TECNOLOGÍA RESPONSIVE WEB DESIGN PARA EL CONTROL DE INFORMACIÓN DE UN BROKER DE LA CIUDAD DE RIOBAMBA,” Riobamba - Ecuador, 2019.
- [9] J. E. Jurado, “IMPLEMENTACIÓN DE UN SISTEMA WEB DE ATENCIÓN MÉDICA UTILIZANDO YII FRAMEWORK EN EL CONSULTORIO MÉDICO GUTIÉRREZ,” Ambato, 2022.

- [10] L. T. Canillas, “ADMISIÓN Y GESTIÓN DE PACIENTES EN ATENCIÓN ESPECIALIZADA.”
- [11] “Documentación clínica 6 SUMARIO.”
- [12] R. L. Tocagon Imbaquingo and R. J. Pelaez Llamba, “SISTEMA WEB PARA LA AUTOMATIZACIÓN DE TURNOS E HISTORIAS CLÍNICAS DE LA FUNDACIÓN CLÍNICA MOSQUERA,” Quito - Ecuador, 2018.
- [13] C. H. URRUTIA MATHEUS, “CREACION DE SISTEMA DE AGENDAMIENTO DE CITAS MÉDICAS PRESENCIAL Y VIRTUAL PARA EL AREA DE CONSULTA EXTERNA,” Guayaquil, 2017.
- [14] J. C. Zambrano Goicochea, “Implementación de un Sistema Web para mejorar el proceso de Reserva de Citas en el Policlínico Dr. Nixon. Bagua Grande, Utcubamba, Amazonas, 2020.,” Bagua Grande Perú, 2020.
- [15] C. Hernández, D. Lyon, A. Pérez, G. Nebiolo, J. García, and L. Sosa, “Tipos de Sistemas de Información TPS, MIS, DSS y ESS.”
- [16] F. González, G. Gómez, D. León, and I. Suárez, “DESCRIPCIÓN GENERAL: SISTEMAS DE INFORMACIÓN PARA EJECUTIVOS (EIS),” 2004.
- [17] G. D. Caballero, “DE LOS PROCESOS Y CLASES DE DATOS A LOS SISTEMAS DE INFORMACIÓN. UN CASO DE ESTUDIO,” 2014.
- [18] E. J. LÓPEZ NAVARRETE, “IMPLEMENTACIÓN DE UN SISTEMA WEB PARA FORTALECER LOS PROCESOS DE GESTIÓN DE LIBROS EN LA BIBLIOTECA DE LA UNIVERSIDAD ESTATAL DEL SUR DE MANABÍ,” Jipijapa, 2017.
- [19] S. Luján, “PROGRAMACIÓN DE APLICACIONES WEB: HISTORIA, PRINCIPIOS BÁSICOS Y CLIENTES WEB.,” Alicante, 2002.
- [20] RJ Code Advance, “Arquitectura en Capas – Análisis completo + Tradicional vs Modernas, DDD, DIP (Cap 5).” <https://rjcodeadvance.com/patrones-de-software-arquitectura-en-capas-analisis-completo-ejemplo-ddd-parte-5/> (accessed Jun. 07, 2023).

- [21] E. Labrada Martínez and C. Salgado Ceballos, “DISEÑO WEB ADAPTATIVO O RESPONSIVO,” *Revista Digital Universitaria*, vol. 14, Jan. 2013, [Online]. Available: <http://www.revista.unam.mx/vol.14/num1/art07/index.html>
- [22] A. E. Veloz Arce, “De la interfaz del usuario al responsive web design,” Guayaquil, 2016. [Online]. Available: www.alistapart.com,
- [23] I. De Souza, “Entiende las diferencias entre Front-End y Back-end en el ambiente de los sitios web,” *rockcontent blog*, Mar. 02, 2020. <https://rockcontent.com/es/blog/front-end-y-back-end/> (accessed May 15, 2023).
- [24] S. Jiménez, “¿Qué es un framework de diseño y por qué las Administraciones Públicas lo necesitan?,” *Analitica Publica*, Feb. 07, 2022. <https://sergiojimenez.net/que-es-un-framework-de-diseno-y-por-que-las-administraciones-publicas-lo-necesitan/> (accessed May 15, 2023).
- [25] J. Ribas, “Tecnologías FrontEnd y BackEnd en el Desarrollo Web,” *Blog Disseny Producte*. <https://dissenyproducte.blogspot.com/2017/10/tecnologias-frontend-y-backend-en-el.html> (accessed May 15, 2023).
- [26] V. Morelli, “¿Qué Es Un Framework De Desarrollo Backend?,” *WheelHub - Blog*, Feb. 25, 2019. <https://wheelhub.es/blog/que-es-un-framework-de-desarrollo-backend/> (accessed May 15, 2023).
- [27] S. Darías, “Gestor de Base de datos: Qué es, Funcionalidades y Ejemplos,” *Intelequia*, Sep. 09, 2021. <https://intelequia.com/blog/post/gestor-de-base-de-datos-qu%C3%A9-es-funcionalidades-y-ejemplos> (accessed May 29, 2023).
- [28] B. García, “BASES DE DATOS,” Sep. 17, 2014. <https://cursos.aiu.edu/base%20de%20datos%20SOG/Sesi%C3%B3n%201.pdf> (accessed May 29, 2023).
- [29] S. Borges, “¿Qué es un Servidor local? Ventajas, desventajas e instalación,” *infranetworking*, Sep. 01, 2020. <https://blog.infranetworking.com/servidor-local/> (accessed May 29, 2023).

- [30] E. Maida and J. Pacienza, "Metodologías de desarrollo de software," PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA SANTA MARIA DE LOS BUENOS AIRES, 2015. Accessed: May 30, 2023. [Online]. Available:
<https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- [31] Santander Universidades, "Metodologías de desarrollo software," Dec. 21, 2020. <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html> (accessed May 30, 2023).
- [32] AdmItsqmet, "Las 8 metodologías de desarrollo de software," *Blog - ITSQMET*, Dec. 12, 2022. https://itsqmet.edu.ec/los-8-tipos-de-metodologia-de-desarrollo-de-software/#Metodologia_de_Espiral (accessed May 30, 2023).
- [33] S. Martínez, "Metodología iterativa o incremental en la gestión de proyectos," Dec. 14, 2014. <https://www.mundoerp.com/blog/metodologia-iterativa-o-incremental-gestion-proyectos/> (accessed May 30, 2023).
- [34] M. Castro, "Metodología RAD o DRA. El Desarrollo Rápido de Aplicaciones," Dec. 25, 2019. <https://www.incentro.com/es-ES/blog/metodologia-rad-desarrollo-rapido-aplicaciones> (accessed May 30, 2023).
- [35] J. Cendejas, "Modelos Y Metodologías Para El Desarrollo De Software." <https://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm> (accessed May 30, 2023).
- [36] M. Coppola, "Qué es Domain Driven Design (DDD), ventajas y proceso básico," Apr. 13, 2023. <https://blog.hubspot.es/website/que-es-ddd> (accessed May 30, 2023).
- [37] M. Humanes, "Metodología Lean: ¿qué es y cómo aplicarla en tu empresa?," Jul. 23, 2019. <https://www.ekon.es/blog/metodologia-lean-empresa/> (accessed May 30, 2023).
- [38] M. Lamarca, "Modelo RMM," Dec. 08, 2013. <http://www.hipertexto.info/documentos/rmm.htm> (accessed May 30, 2023).

- [39] M. Lamarca, “Modelo OOHDm,” Dec. 08, 2013.
<http://www.hipertexto.info/documentos/ooohdm.htm> (accessed May 31, 2023).
- [40] R. Soliz and F. Morales, “OOHDm (MÉTODO DE DISEÑO HIPERMEDIA OBJETO ORIENTADO) & NORMATIVA ISO 9126,” Mar. 2014.
<https://darjelingsilva.files.wordpress.com/2018/05/5-metd-ooohdm.pdf>
(accessed Jun. 01, 2023).
- [41] E. Bello, “Descubre qué es el Extreme Programming y sus características,” Apr. 28, 2021. <https://www.iebschool.com/blog/que-es-el-xp-programming-agile-scrum/> (accessed Jun. 01, 2023).
- [42] A. Raeburn, “La programación extrema (XP) produce resultados, pero ¿es la metodología adecuada para ti?,” *Asana*, Nov. 28, 2022.
<https://asana.com/es/resources/extreme-programming-xp> (accessed Jun. 01, 2023).
- [43] J. Kanjilal, “Introduction to Rational Unified Process (RUP) ,” Dec. 07, 2022.
<https://www.developer.com/project-management/rational-unified-process-rup/>
(accessed Jun. 01, 2023).
- [44] N. Guerrero, “¿Que Es El Proceso Unificado de Rational (RUP)? ,” May 25, 2018. <https://www.programaenlinea.net/proceso-unificado-rational-rup/>
(accessed Jun. 01, 2023).
- [45] J. Melgoza, “Como Crear una Master Page en Asp.Net - Tutorial,” Apr. 06, 2014.
- [46] J. Teran and R. Vizcaíno, “Breve introducción a MockFlow,” Jul. 31, 2020.
<https://masterenbuscadores.com/2020/07/31/breve-introduccion-a-mockflow/>
(accessed Jun. 15, 2023).
- [47] G. García, “PATRONES SINGLETON Y LAZY INIZIALIZATION EN C# Y JAVA,” Oct. 18, 2020. <https://gonzalogarciamr.com/2020/10/18/patterns-singleton-lazy/> (accessed Jun. 15, 2023).

- [48] S. Lizaraburu, “Patrón SINGLETON en C#/NET 6,” May 30, 2023.
<https://stevenlizaraburupezua.com/patron-singleton-implementacion-y-uso-en-c-net-6/> (accessed Jun. 15, 2023).
- [49] M. Alvarez, “AJAX en jQuery,” Oct. 05, 2021.
<https://desarrolloweb.com/articulos/entendiendo-ajax-jquery.html> (accessed Jun. 15, 2023).
- [50] A. Garro, “CAPÍTULO 7 AJAX,” Jan. 14, 2014.
<https://www.arkaitzgarro.com/jquery/capitulo-7.html> (accessed Jun. 15, 2023).

ANEXOS

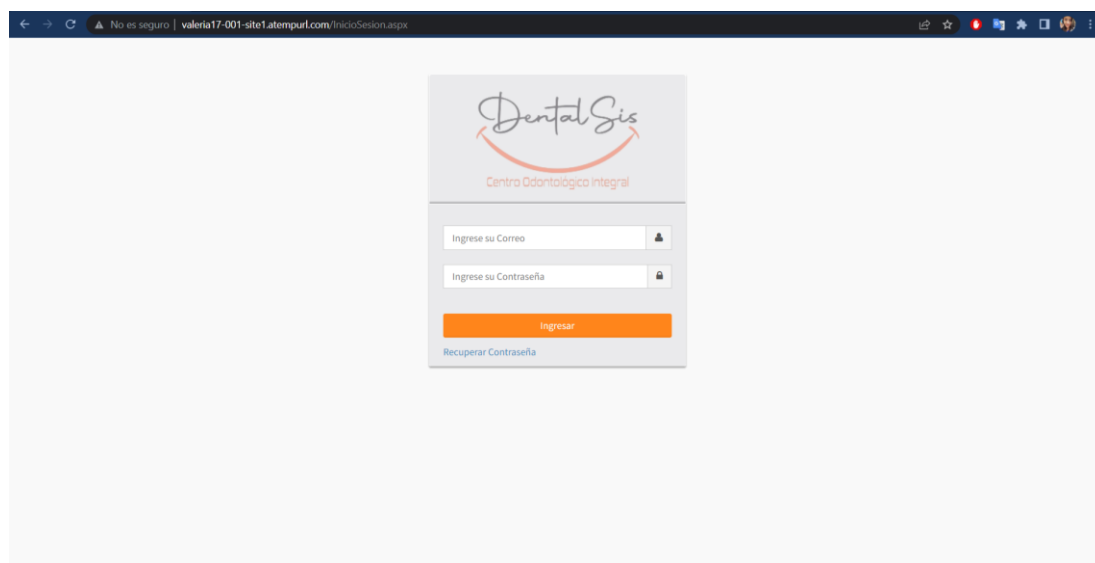
ANEXO 1. Manual de Usuario

El manual se encuentra estructurado de la siguiente forma:

1. Ingreso al Sistema
2. Restablecer Contraseña
3. Página Principal
4. Gestión Pacientes
5. Gestión Odontólogos
6. Gestión Horarios de Odontólogos
7. Reserva de Citas
8. Atención de Citas
9. Comprobante
10. Finalizar Sesión
11. Alertas

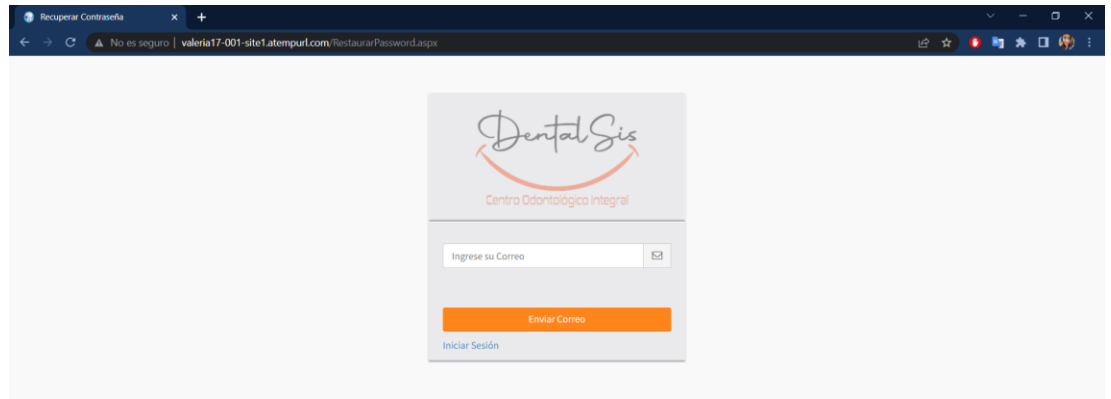
1. Ingreso al Sistema

En la interfaz de inicio de sesión se solicita al usuario ingresar el correo y clave ya establecidas previamente.



2. Restablecer Contraseña

Si el usuario no recuerda su contraseña, en la ventana de recuperar se debe digitar el correo con el cual está registrado y dar clic en enviar correo.



El sistema le mostrara un mensaje dependiendo del caso.

Si está registrado el correo



No está registrado el correo



Luego el usuario debe verificar el mensaje en su correo.

SISTEMA: Solicitud de recuperación de contraseña

dental.sis2611@gmail.com

para mí ▾

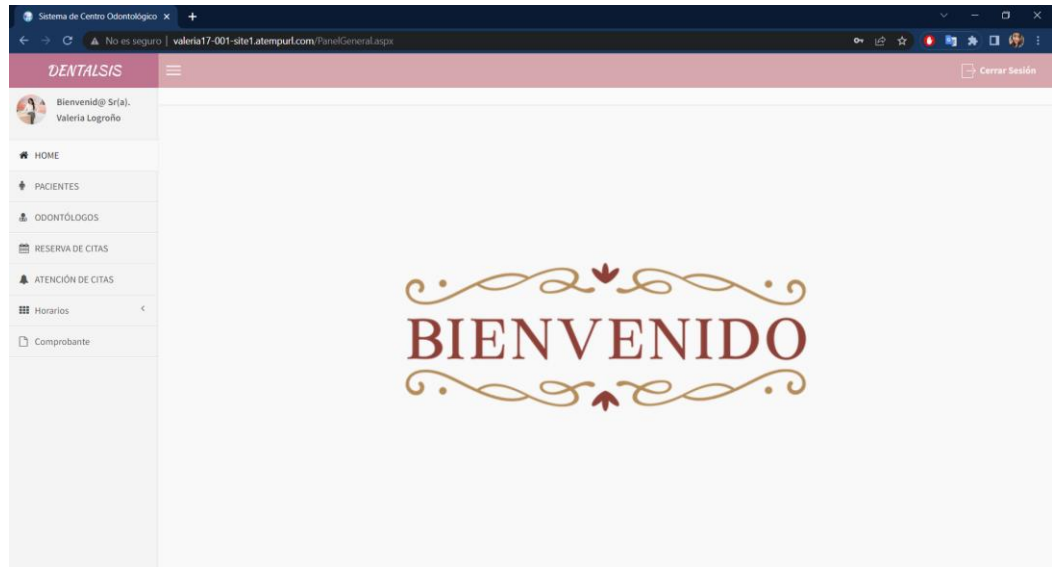
Hola, Valeria Logroño

Usted solicitó recuperar su contraseña.

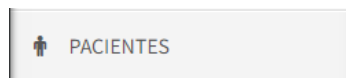
Su contraseña actual es: 12345

3. Página Principal

En la página principal se muestra un mensaje de bienvenida, los datos e imagen del usuario activo, la opción de cerrar sesión y los módulos de la aplicación web.



4. Gestión Pacientes



La opción Pacientes se divide en dos secciones:

- Registro de pacientes con datos esenciales para el odontólogo.

Nota: Todos los campos poseen validaciones ya sean de texto como numéricos.

Para agregar al paciente después de ingresar sus datos se debe dar clic en el botón

(Registrar)  .

- Listado de los pacientes registrados



LISTA DE PACIENTES

10 Registros por Página Buscar:

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	DIRECCIÓN	ACCIONES
1803747839	Jessica Alexandra	Logroño Padilla	Jessy@gmail.com	0984004014	Izamba	 
1762848546	MARÍA PAJULA	MORETA VILLACIS	mapau@gmail.com	0985116546	FICOA	 
1708036985	Oscar Mauricio	Macas Toapanta	mauMT@gmail.com	0978456321	Huachi Chico	 
1705251732	Adela María	Lino Suarez	adelaLinoa@gmail.com	0997763974	Atocha	 
1600259678	JUAN PABLO	ORTÍZ PARRALES	JuanPO2000@gmail.com	0987654321	ATAHUALPA	 
0501355853	Edmundo Filiberto	Logroño Castillo	edmundo16@gmail.com	0982365489	El Pisque	 

Mostrando 1 a 6 de 6 entradas ← Anterior 1 Siguiente →

En la tabla de los pacientes se ubica la columna de acciones, actualizar y eliminar.

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	DIRECCIÓN	ACCIONES
1705251732	Adela María	Lino Suarez	adelaLinoa@gmail.com	0997763974	Atocha	 

Además, se puede buscar a los pacientes por los datos que se muestra en la tabla.

Cédula

LISTA DE PACIENTES

10 Registros por Página Buscar:

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	DIRECCIÓN	ACCIONES
1705251732	Adela María	Lino Suarez	adelaLinoa@gmail.com	0997763974	Atocha	 
1708036985	Oscar Mauricio	Macas Toapanta	mauMT@gmail.com	0978456321	Huachi Chico	 

Mostrando 1 a 2 de 2 entradas (Filtrado de 6 entradas totales) ← Anterior 1 Siguiente →

Nombre

LISTA DE PACIENTES

10 Registros por Página Buscar:

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	DIRECCIÓN	ACCIONES
1600259678	JUAN PABLO	ORTÍZ PARRALES	JuanPO2000@gmail.com	0987654321	ATAHUALPA	 

Mostrando 1 a 1 de 1 entradas (Filtrado de 6 entradas totales) ← Anterior 1 Siguiente →

Apellido

LISTA DE PACIENTES

10 Registros por Página Buscar:

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	DIRECCIÓN	ACCIONES
0501355853	Edmundo Filiberto	Logroño Castillo	edmundo16@gmail.com	0982365489	El Pisque	 
1803747839	Jessica Alexandra	Logroño Padilla	Jessy@gmail.com	0984004014	Izamba	 

Mostrando 1 a 2 de 2 entradas (Filtrado de 6 entradas totales) ← Anterior 1 Siguiente →

Correo

LISTA DE PACIENTES

10 Registros por Página Buscar:

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	DIRECCIÓN	ACCIONES
1708036985	Oscar Mauricio	Macas Toapanta	mauMT@gmail.com	0978456321	Huachi Chico	 

Mostrando 1 a 1 de 1 entradas (Filtrado de 6 entradas totales) ← Anterior 1 Siguiente →

Teléfono

LISTA DE PACIENTES

10 Registros por Página Buscar: 0985116546

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	DIRECCIÓN	ACCIONES
1762848546	MARÍA PAULA	MORETA VILLACIS	mapau@gmail.com	0985116546	FICOA	 

Mostrando 1 a 1 de 1 entradas (Filtrado de 6 entradas totales) ← Anterior 1 Siguiente →


Dirección

LISTA DE PACIENTES

10 Registros por Página Buscar: FICOA

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	DIRECCIÓN	ACCIONES
1803747839	Jessica Alexandra	Logroño Padilla	Jessy@gmail.com	0984004014	FICOA	 
1762848546	MARÍA PAULA	MORETA VILLACIS	mapau@gmail.com	0985116546	FICOA	 

Mostrando 1 a 2 de 2 entradas (Filtrado de 6 entradas totales) ← Anterior 1 Siguiente →

Al momento de hacer clic en el botón (Actualizar)  se mostrará un modal con el nombre completo del paciente y los datos que se pueden editar

Actualizar Registro ✕

NOMBRES Y APELLIDOS

Oscar Mauricio Macas Toapanta

CORREO ELECTRÓNICO

mauMT@gmail.com

TELÉFONO

0978456321

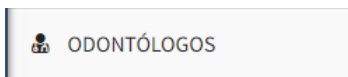
DIRECCIÓN

Huachi Chico

Actualizar Cancelar

Al momento de hacer clic en el botón (Eliminar)  se eliminará el paciente.

5. Gestión Odontólogos



La opción Odontólogos se divide en dos secciones:

- Registro de odontólogos con datos relevantes.

REGISTRO DE ODONTÓLOGOS

<p>CÉDULA DE IDENTIDAD</p> <input style="width: 95%;" type="text"/>	<p>APELLIDO MATERNO</p> <input style="width: 95%;" type="text"/>
<p>PRIMER NOMBRE</p> <input style="width: 95%;" type="text"/>	<p>CORREO ELECTRÓNICO</p> <input style="width: 95%;" type="text"/>
<p>SEGUNDO NOMBRE</p> <input style="width: 95%;" type="text"/>	<p>ESPECIALIDAD</p> <input style="width: 95%;" type="text" value="Implantología Oral"/>
<p>APELLIDO PATERNO</p> <input style="width: 95%;" type="text"/>	<p>TELÉFONO</p> <input style="width: 95%;" type="text"/>

Registrar
Cancelar

Nota: Todos los campos poseen validaciones ya sean de texto como numéricos. Para agregar al odontólogo después de ingresar sus datos se debe dar clic en el botón (Registrar) .

- Listado de los odontólogos registrados

LISTA DE ODONTÓLOGOS

10 Registros por Página Buscar:

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	ESPECIALIDAD	ACCIONES
1804392650	Pamela Magda	Toapanta Chávez	valeria1998.michelle@gmail.com	0983967363	Implantología Oral	
1704963245	Kelvin Alexander	Morocho Naranjo	kelvin.morocho123@gmail.com	0969148532	Endodoncia	

Mostrando 1 a 2 de 2 entradas ← Anterior 1 Siguiente →

En la tabla de los odontólogos se ubica la columna de acciones, actualizar y eliminar.

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	ESPECIALIDAD	ACCIONES
1804392650	Pamela Magda	Toapanta Chávez	valeria1998.michelle@gmail.com	0983967363	Implantología Oral	

Además, se puede buscar a los odontólogos por los datos que se muestra en la tabla.

Cédula

LISTA DE ODONTÓLOGOS

10 Registros por Página Buscar:

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	ESPECIALIDAD	ACCIONES
1804392650	Pamela Magda	Toapanta Chávez	valeria1998.michelle@gmail.com	0983967363	Implantología Oral	

Mostrando 1 a 1 de 1 entradas (Filtrado de 2 entradas totales) ← Anterior 1 Siguiente →

Nombre

LISTA DE ODONTÓLOGOS

10 Registros por Página Buscar:

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	ESPECIALIDAD	ACCIONES
1704963245	Kelvin Alexander	Morocho Naranjo	kelvin.morocho123@gmail.com	0969148532	Endodoncia	



Mostrando 1 a 1 de 1 entradas (Filtrado de 2 entradas totales) ← Anterior 1 Siguiente →

Apellido

LISTA DE ODONTÓLOGOS

10 Registros por Página

Buscar: Chávez

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	ESPECIALIDAD	ACCIONES
1804392650	Pamela Magda	Toapanta Chávez	valeria1998.michelle@gmail.com	0983967363	Implantología Oral	 

Mostrando 1 a 1 de 1 entradas (Filtrado de 2 entradas totales)



← Anterior 1 Siguiente →

Correo

LISTA DE ODONTÓLOGOS

10 Registros por Página

Buscar: kelvin.morocho123@gmail.com

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	ESPECIALIDAD	ACCIONES
1704963245	Kelvin Alexander	Morocho Naranjo	kelvin.morocho123@gmail.com	0969148532	Endodoncia	 

Mostrando 1 a 1 de 1 entradas (Filtrado de 2 entradas totales)



← Anterior 1 Siguiente →

Teléfono

LISTA DE ODONTÓLOGOS

10 Registros por Página

Buscar: 0969

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	ESPECIALIDAD	ACCIONES
1804392650	Pamela Magda	Toapanta Chávez	valeria1998.michelle@gmail.com	0983967363	Implantología Oral	 

Mostrando 1 a 1 de 1 entradas (Filtrado de 2 entradas totales)



← Anterior 1 Siguiente →

Especialidad

LISTA DE ODONTÓLOGOS


10 Registros por Página

Buscar: Endodoncia

CÉDULA	NOMBRES	APELLIDOS	EMAIL	TELÉFONO	ESPECIALIDAD	ACCIONES
1704963245	Kelvin Alexander	Morocho Naranjo	kelvin.morocho123@gmail.com	0969148532	Endodoncia	 

Mostrando 1 a 1 de 1 entradas (Filtrado de 2 entradas totales)

← Anterior 1 Siguiente →


Al momento de hacer clic en el botón (Actualizar)  se mostrará un modal con el nombre completo del odontólogo y los datos que se pueden editar

Actualizar Registro

NOMBRES Y APELLIDOS

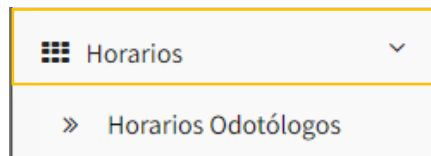
CORREO ELECTRÓNICO

TELÉFONO

Al momento de hacer clic en el botón (Eliminar)  se eliminará el odontólogo.

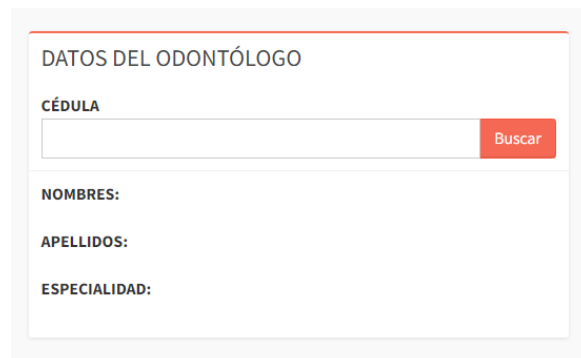
6. Gestión Horarios de Odontólogos

Para gestionar los horarios se debe dar clic en Horarios que desplegara la opción de Horarios odontólogos.

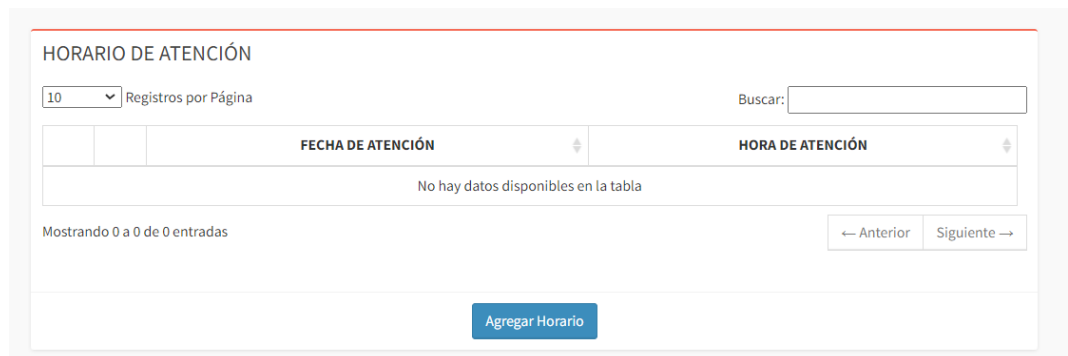


En la ventana de Gestionar horarios se visualiza dos secciones:


- Datos del odontólogo



- Horarios de Atención



El usuario debe ingresar la cédula de un odontólogo y dar clic en (Buscar)

 para que se visualice los horarios que posee.

GESTIÓN DE HORARIOS DE ODONTÓLOGOS

DATOS DEL ODONTÓLOGO

CÉDULA
1804392650 Buscar

NOMBRES: Pamela Magda

APELLIDOS: Toapanta Chávez

ESPECIALIDAD: Implantología Oral

HORARIO DE ATENCIÓN

10 Registros por Página Buscar:

		FECHA DE ATENCIÓN	HORA DE ATENCIÓN
		20/07/2023	17:00
		25/07/2023	11:00

Mostrando 1 a 2 de 2 entradas ← Anterior **1** Siguiente →

Agregar Horario

Al momento de dar clic en el botón (Agregar Horario) Agregar Horario se mostrará un modal para agregar un nuevo horario de atención.

AGREGAR HORARIO ATENCIÓN X

FECHA:

HORA DE INICIO: ⌚

↑
09
↓

:

↑
30
↓

Agregar
Cancelar

En la tabla de los horarios de atención las dos primeras columnas son actualizar y eliminar.

		FECHA DE ATENCIÓN	HORA DE ATENCIÓN
		20/07/2023	17:00

Además, se puede buscar los horarios por fecha u hora de atención.

Fecha

HORARIO DE ATENCIÓN

10 Registros por Página Buscar:

		FECHA DE ATENCIÓN	HORA DE ATENCIÓN
		25/07/2023	11:00

Mostrando 1 a 1 de 1 entradas (Filtrado de 2 entradas totales) ← Anterior **1** Siguiente →


Hora

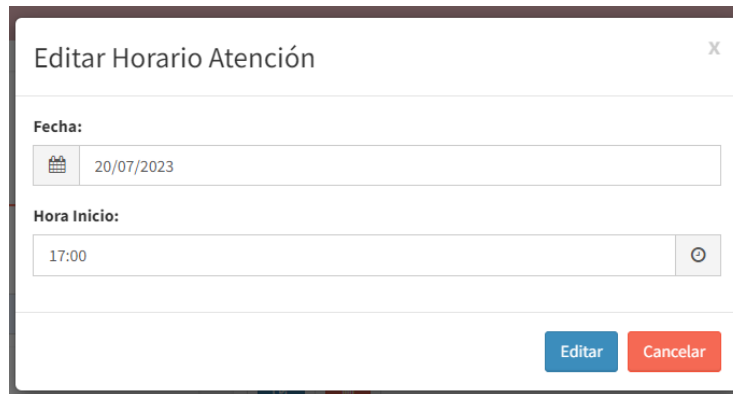
HORARIO DE ATENCIÓN

10 Registros por Página Buscar:

		FECHA DE ATENCIÓN	HORA DE ATENCIÓN
		20/07/2023	17:00

Mostrando 1 a 1 de 1 entradas (Filtrado de 2 entradas totales) ← Anterior **1** Siguiente →

Al momento de hacer clic en el botón (Actualizar)  se mostrará un modal con la fecha y hora del horario seleccionado.




Editar Horario Atención

Fecha: 20/07/2023

Hora Inicio: 17:00

Editar Cancelar

Al momento de hacer clic en el botón (Eliminar)  se eliminará el Horario de Atención.

7. Reserva de Citas

En la reserva de citas se encuentra dos secciones:

- Datos del paciente



DATOS DEL PACIENTE

CÉDULA DE IDENTIDAD

NOMBRES

APELLIDOS

TÉLFONO

EDAD

GÉNERO

- Horario de Atención



HORARIOS DE ATENCIÓN

FECHA

ESPECIALIDAD

Para reservar una cita el usuario deberá seguir los siguientes pasos:

- (1) Buscar al paciente por su cedula



DATOS DEL PACIENTE

CÉDULA DE IDENTIDAD 0501355853

NOMBRES Edmundo Filiberto

APELLIDOS Logroño Castillo

TÉLFONO 0982365489

EDAD 59

GÉNERO Masculino

(2) Ingresar la fecha

FECHA

 20/07/2023

(3) Seleccionar la especialidad y pulsar buscar

ESPECIALIDAD

Implantología Oral 

Buscar

(4) De la lista de horarios disponibles, seleccionar la casilla del que se desee y dar clic en el botón (Reservar Cita)

Reservar Cita

	Hora de Atención	Odontólogo
<input checked="" type="checkbox"/>	17:00	Toapanta Pamela

8. Atención de Citas

En la atención de citas se visualiza una lista de las citas pendientes de atender.

GESTIONAR ATENCIÓN ODONTOLÓGICA



ID CITA: 20
FECHA DE LA ATENCIÓN: 14/07/2023
NOMBRES: Edmundo Filiberto
APELLIDOS: Logroño Castillo
EDAD: 59
GÉNERO: Masculino

Realizar Atención **Cancelar**

HORA DE ATENCIÓN: 17:00



ID CITA: 21
FECHA DE LA ATENCIÓN: 14/07/2023
NOMBRES: Jessica Alexandra
APELLIDOS: Logroño Padilla
EDAD: 34
GÉNERO: Femenino

Realizar Atención **Cancelar**

HORA DE ATENCIÓN: 11:00

El usuario puede realizar la atención como cancelar cita.

Realizar Atención **Cancelar**

Al atender la cita se muestra una pagina en la cual se visualiza los datos del paciente, también se puede registrar un diagnóstico y observaciones.

DATOS DEL PACIENTE



NOMBRES: Edmundo Filiberto
APELLIDOS: Logroño Castillo
EDAD: 59
GÉNERO: Masculino
GÉNERO:

OBSERVACIONES

DIAGNÓSTICO

[Registrar](#)

9. Comprobante

El módulo de comprobante tiene el objetivo de crear un formato que se pueda imprimir para lo cual posee dos secciones:

- Datos del Paciente

DATOS DEL PACIENTE

CÉDULA DE IDENTIDAD

[BUSCAR](#)

NOMBRE Y APELLIDO

- Servicios

SERVICIOS

TRATAMIENTOS

PRECIO

[AGREGAR](#)

Para generar el comprobante se debe seguir los siguientes pasos:

(1) Buscar al paciente por cedula

DATOS DEL PACIENTE

CÉDULA DE IDENTIDAD

[BUSCAR](#)

NOMBRE Y APELLIDO

(2) Ingresar el tratamiento y precio correspondiente

SERVICIOS

TRATAMIENTOS

PRECIO

(3) Dar clic en el botón (Agregar)

[AGREGAR](#)

(4) Se visualiza una lista de los tratamientos y precios con el valor total

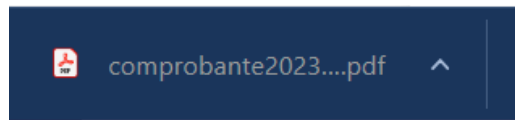
DESCRIPCIÓN	PRECIO
LIMPIEZA BUCAL	30
TOTAL: <input style="width: 100px;" type="text" value="\$ 30"/>	

[GENERAR PDF](#)

(5) Al dar clic en el botón (Generar PDF)

[GENERAR PDF](#)

(6) Se crea y descarga un archivo.pdf

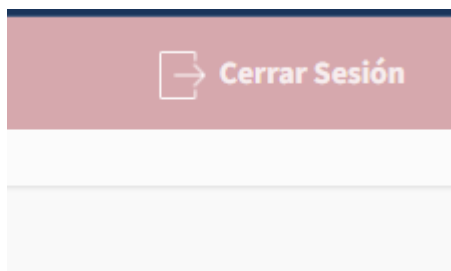


(7) Al abrir el Archivo, se visualiza dos copias del comprobante, de las cuales una se queda el centro odontológico y la otra se entrega al paciente.



10. Finalizar Sesión

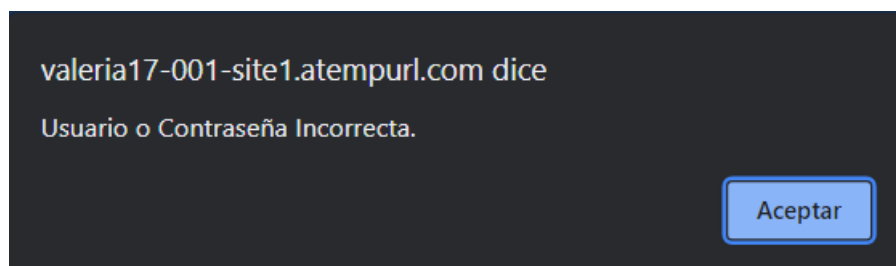
La opción de cerrar sesión se encuentra en la esquina superior derecha, y al hacerlo, será redirigido automáticamente a la página de inicio del sistema.



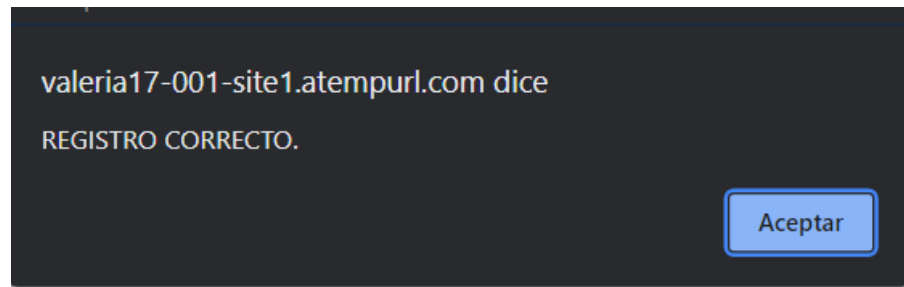
11. Alertas

Entre las alertas que nos proporciona la aplicación web son:

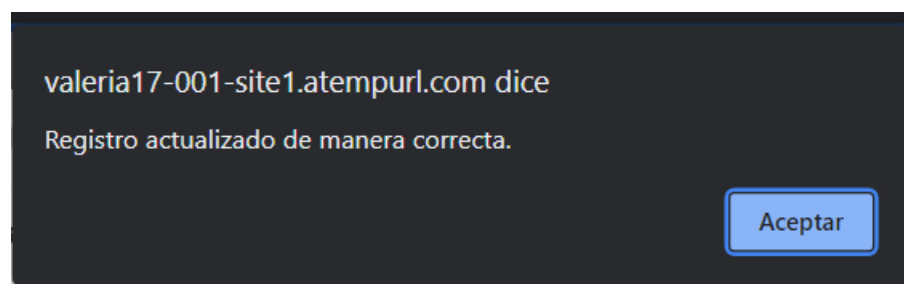
- Cuando se está tratando de ingresar con un correo no registrado o está mal la contraseña.



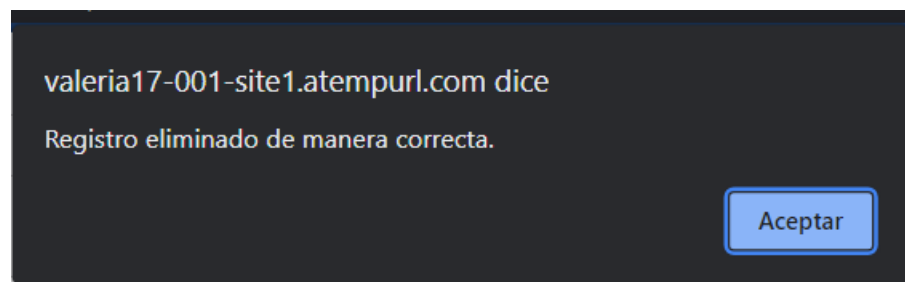
- Cuando se realiza el registro de un paciente u odontólogo



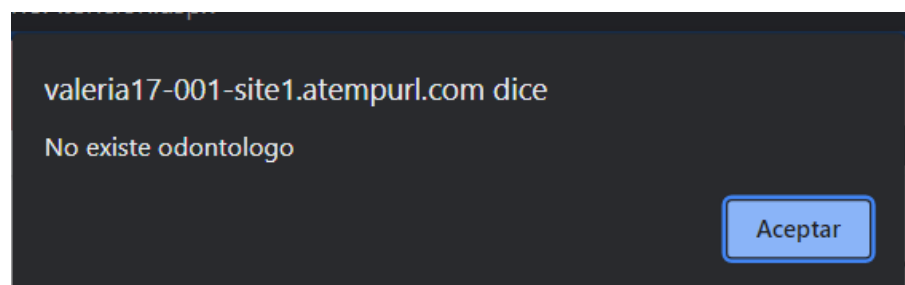
- Cuando se realiza la actualización de información de un paciente, odontólogo u horario de atención.



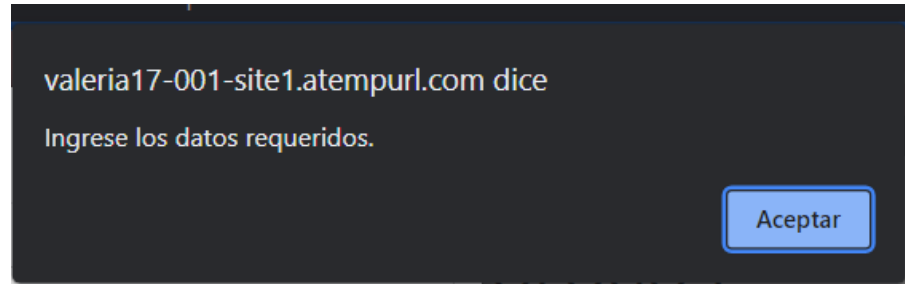
- Cuando se elimina un paciente, odontólogo u horario de atención.



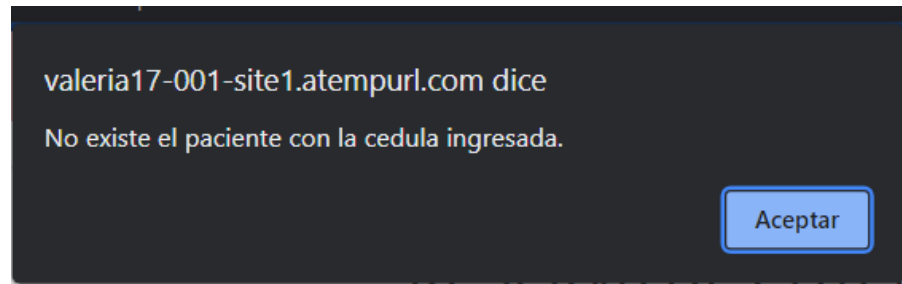
- Cuando no se efectúa la búsqueda de un odontólogo por su cedula.



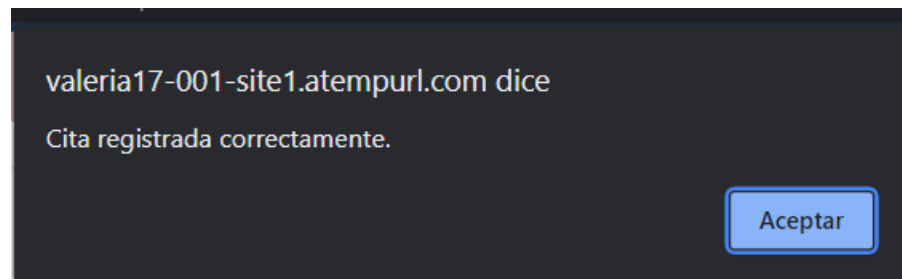
- Cuando no se ingresa datos para completar con el proceso deseado.



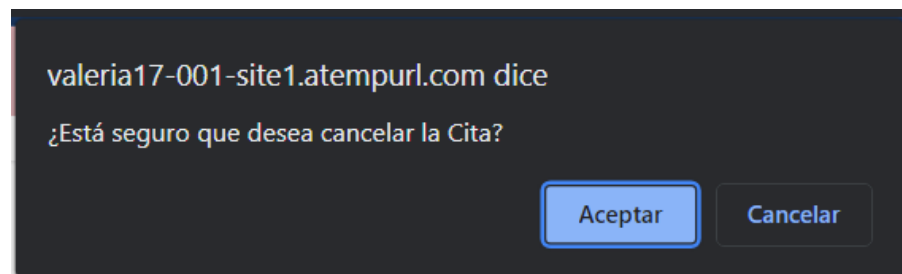
- Cuando se está buscando a un paciente no registrado.



- Cuando se registra una cita



- Cuando se cancela la atención de la cita



- Cuando no se tiene citas pendientes

