



UNIVERSIDAD TÉCNICA DE AMBATO

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

CARRERA DE SOFTWARE

Tema:

**APLICACIÓN WEB CENTRALIZADA PARA LA GESTIÓN Y ATENCIÓN
DE TURNOS DE LAS SUCURSALES DE LA EP-EMAPA-A**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero de Software

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de Software

AUTOR: Javier Alexander Jiménez Castro

TUTOR: Ing. Oscar Fernando Ibarra Torres

Ambato - Ecuador

febrero – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: APLICACIÓN WEB CENTRALIZADA PARA LA GESTIÓN Y ATENCIÓN DE TURNOS DE LAS SUCURSALES DE LA EP-EMAPA-A, desarrollado bajo la modalidad Proyecto de Investigación por el señor Javier Alexander Jiménez Castro, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado 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, febrero 2024.

Ing. Oscar Fernando Ibarra Torres
TUTOR

AUTORÍA

El presente trabajo de titulación con el tema APLICACIÓN WEB CENTRALIZADA PARA LA GESTIÓN Y ATENCIÓN DE TURNOS DE LAS SUCURSALES DE LA EP-EMAPA-A 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, febrero 2024.



Javier Alexander Jiménez Castro

C.C. 180437243-9

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, febrero 2024.



Javier Alexander Jiménez Castro

C.C. 180437243-9

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Javier Alexander Jiménez Castro, estudiante de la Carrera de Software de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado APLICACIÓN WEB CENTRALIZADA PARA LA GESTIÓN Y ATENCIÓN DE TURNOS DE LAS SUCURSALES DE LA EP-EMAPA-A, 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, febrero 2024.

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

Ing. Pablo Israel Morales Paredes
PROFESOR CALIFICADOR

Ing. Marco Vinicio Guachimboza Villalva
PROFESOR CALIFICADOR

DEDICATORIA

Este proyecto está dedicado a mis padres Jorge y María, cuyo apoyo y amor incondicional han sido fundamentales durante este viaje académico.

A mi hermano Andrés, cómplice de risas e inagotable fuente de inspiración y motivación en mi vida, tu apoyo ha sido la chispa que avivó mi determinación.

A mi familia, por ser un pilar en el cumplimiento de mis metas y una fuente de apoyo en mi vida, quienes han compartido mis triunfos y momentos difíciles.

Por último, a mis amigos quienes han sido parte fundamental de este camino y a quienes considero parte de mi familia por haber compartido experiencias tanto buenas como malas y aun así haber mantenido la amistad durante tantos años.

Javier Alexander Jiménez Castro

AGRADECIMIENTO

Quiero agradecer a mis padres por ser un pilar en mi vida, gracias por ayudarme a conseguir un logro más, su dedicación y amor han sido mi guía, son un ejemplo de perseverancia y sacrificio, este logro también es de ustedes.

A mi hermano Andrés por tu importancia en este proyecto y en mi vida, tu alegría y compañía iluminan incluso los momentos más desafiantes, nuestro lazo es la fuerza que impulsa mis logros.

A mi Trinidad, este logro lleva consigo el eco de tus palabras y enseñanzas como tributo a la fortaleza y bondad que siempre mostraste. Aunque ya no estés físicamente a mi lado, tu presencia sigue viva en cada paso que doy.

A mi Alberto, por ser un ejemplo de fortaleza, gracias por tus enseñanzas y tu cariño, este logro es un reconocimiento a la influencia que has tenido en mi vida.

Agradezco a mi tutor Ing. Fernando Ibarra por brindarme su orientación y apoyo durante el desarrollo de este proyecto.

Un agradecimiento especial al Ing. Ricardo Rivera, por ofrecerme su ayuda y el apoyo necesario para la culminación de este proyecto.

Por último, a mis amigos por su apoyo y amistad inquebrantable. Gracias por ser no solo amigos, sino una parte fundamental de este camino.

Javier Alexander Jiménez Castro

ÍNDICE GENERAL DE CONTENIDOS

PORTADA	i
APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS	xi
ÍNDICE DE FIGURAS	xv
ÍNDICE DE ANEXOS	xx
RESUMEN EJECUTIVO	xxi
ABSTRACT	xxii
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	3

1.3.1 Sistema de turnos	3
1.3.2 Sistema centralizado.....	4
1.3.3 Sistemas de gestión	4
1.3.4 Framework	5
1.3.5 Sitio Web.....	6
1.3.6 Base de datos.....	7
1.3.7 Servidor Web	8
1.3.8 Metodologías de desarrollo	8
1.4 Objetivos	14
1.4.1 Objetivo general.....	14
1.4.2 Objetivos específicos	14
CAPÍTULO II. METODOLOGÍA	15
2.1 Materiales.....	15
2.2 Métodos.....	15
2.2.1 Modalidad de la investigación	15
2.2.2 Población y muestra	16
2.2.3 Recolección de información.....	16
2.2.4 Procesamiento y análisis de datos	36
CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....	38
3.1 Análisis y discusión de los resultados	38

3.1.1 Diagrama del proceso con la gestión manual.....	38
3.1.2 Diagrama del proceso con la aplicación.....	39
3.1.3 Análisis y selección de la metodología de desarrollo	43
3.2 Desarrollo de la propuesta.....	46
3.2.1 Fase I: Recopilación de requerimientos	46
3.2.2 Fase II: Prototipado y retroalimentación del cliente	52
3.2.3 Fase III: Construcción	61
3.2.4 Fase IV: Pruebas	86
3.2.5 Fase V: Implementación	106
3.3 Resultados	109
3.3.1 Modelo TAM	110
3.3.2 Modelo UTAUT.....	118
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	125
4.1 Conclusiones	125
4.2 Recomendaciones.....	127
REFERENCIAS BIBLIOGRÁFICAS	129
ANEXOS	133

ÍNDICE DE TABLAS

Tabla 1. Resultados de la entrevista.....	16
Tabla 2. Resultados de las varianzas de la encuesta a los trabajadores	19
Tabla 3. Resultados de las varianzas de la encuesta a los clientes.....	20
Tabla 4. Rangos de confiabilidad [5]	20
Tabla 5. Tabla de frecuencias – Encuesta 1 - Pregunta 1.....	21
Tabla 6. Tabla de frecuencias – Encuesta1 – Pregunta 2.....	22
Tabla 7. Tabla de frecuencias – Encuesta 1 - Pregunta 3.....	23
Tabla 8. Tabla de frecuencias – Encuesta 1 – Pregunta 4.....	24
Tabla 9. Tabla de frecuencias – Encuesta 1 - Pregunta 5.....	25
Tabla 10. Tabla de frecuencias – Encuesta 1 – Pregunta 6.....	26
Tabla 11. Tabla de frecuencias – Encuesta 1 – Pregunta 7	27
Tabla 12. Tabla de frecuencias – Encuesta 2 – Pregunta 1	29
Tabla 13. Tabla de frecuencias – Encuesta 2 – Pregunta 2.....	30
Tabla 14. Tabla de frecuencias – Encuesta 2 – Pregunta 3.....	31
Tabla 15. Tabla de frecuencias – Encuesta 2 – Pregunta 4.....	32
Tabla 16. Tabla de frecuencias – Encuesta 2 – Pregunta 5.....	33
Tabla 17. Tabla de frecuencias – Encuesta 2 – Pregunta 6.....	34
Tabla 18. Tabla de frecuencias – Encuesta 2 – Pregunta 7	35
Tabla 19. Tabla comparativa – Tipos de metodologías	43
Tabla 20. Tabla comparativa – Metodologías de desarrollo ágiles.....	44

Tabla 21. Tabla comparativa – Herramientas de desarrollo	45
Tabla 22. Definición de roles y funciones	47
Tabla 23. Requerimientos técnicos	47
Tabla 24. Requerimientos iniciales	48
Tabla 25. Plan de entrega	50
Tabla 26. Requerimientos finales.....	58
Tabla 27. Prueba de aplicación – Requerimiento R001	86
Tabla 28. Prueba de aplicación – Requerimiento R002.....	86
Tabla 29. Prueba de aplicación – Requerimiento R003.....	87
Tabla 30. Prueba de aplicación – Requerimiento R004.....	88
Tabla 31. Prueba de aplicación – Requerimiento R005.....	88
Tabla 32. Prueba de aplicación – Requerimiento R006.....	89
Tabla 33. Prueba de aplicación – Requerimiento R007.....	89
Tabla 34. Prueba de aplicación – Requerimiento R008.....	90
Tabla 35. Prueba de aplicación – Requerimiento R009.....	91
Tabla 36. Prueba de aplicación – Requerimiento R010.....	99
Tabla 37. Prueba de aplicación – Requerimiento R011.....	99
Tabla 38. Prueba de aplicación – Requerimiento R012.....	101
Tabla 39. Prueba de aplicación – Requerimiento R013.....	101
Tabla 40. Prueba de aplicación – Requerimiento R014.....	102
Tabla 41. Pruebas de aceptación – Iteración 1	102

Tabla 42. Pruebas de aceptación – Iteración 2.....	103
Tabla 43. Pruebas de aceptación – Iteración 3.....	103
Tabla 44. Pruebas de aceptación – Iteración 4.....	104
Tabla 45. Pruebas de aceptación – Iteración 5.....	105
Tabla 46. Pruebas de usabilidad.....	105
Tabla 47. Cronograma de implementación.....	106
Tabla 48. Cuestionario de evaluación del modelo TAM.....	110
Tabla 49. Tabla de frecuencias – Modelo TAM – Percepción de utilidad.....	112
Tabla 50. Resultados modelo TAM – Percepción de utilidad.....	113
Tabla 51. Tabla de frecuencias – Modelo TAM – Facilidad de uso.....	114
Tabla 52. Resultados modelo TAM – Facilidad de uso.....	114
Tabla 53. Tabla de frecuencias – Modelo TAM – Actitud hacia el uso.....	115
Tabla 54. Resultados modelo TAM – Actitud hacia el uso.....	116
Tabla 55. Tabla de frecuencias – Modelo TAM – Intención de uso.....	117
Tabla 56. Resultados modelo TAM – Intención de uso.....	117
Tabla 57. Cuestionario de evaluación del modelo UTAUT.....	118
Tabla 58. Tabla de frecuencias – Modelo UTAUT – Expectativa de rendimiento..	119
Tabla 59. Resultados modelo UTAUT – Expectativa de rendimiento.....	120
Tabla 60. Tabla de frecuencias – Modelo UTAUT – Esfuerzo percibido.....	120
Tabla 61. Resultados modelo UTAUT – Esfuerzo percibido.....	121
Tabla 62. Tabla de frecuencias – Modelo UTAUT – Condiciones sociales.....	122

Tabla 63. Resultados modelo UTAUT – Condiciones sociales.....	122
Tabla 64. Tabla de frecuencias – Modelo UTAUT – Experiencia previa	123
Tabla 65. Resultados modelo UTAUT – Experiencia previa	124

ÍNDICE DE FIGURAS

Figura 1. Etapas modelo en cascada [1].....	9
Figura 2. Fases del modelo en espiral [2]	10
Figura 3. Ciclo de vida de RAD [3]	11
Figura 4. Ciclo de vida de Scrum [4]	12
Figura 5. Etapas de XP.....	13
Figura 6. Resultados Encuesta 1 – Pregunta 1	22
Figura 7. Resultados Encuesta 1 – Pregunta 2	23
Figura 8. Resultados Encuesta 1 – Pregunta 3	24
Figura 9. Resultados Encuesta 1 – Pregunta 4	25
Figura 10. Resultados Encuesta 1 – Pregunta 5	26
Figura 11. Resultados Encuesta 1 – Pregunta 6	27
Figura 12. Resultados Encuesta 1 – Pregunta 7	28
Figura 13. Resultados Encuesta 2 – Pregunta 1	29
Figura 14. Resultados Encuesta 2 – Pregunta 2	30
Figura 15. Resultados Encuesta 2 – Pregunta 3	31
Figura 16. Resultados Encuesta 2 – Pregunta 4	32
Figura 17. Resultados Encuesta 2 – Pregunta 5	33
Figura 18. Resultados Encuesta 2 – Pregunta 6	34
Figura 19. Resultados Encuesta 2 – Pregunta 7	35
Figura 20. Diagrama del proceso manual de atención de turnos	39

Figura 21. Diagrama del proceso de generación y atención de turnos con la aplicación	41
Figura 22. Diagrama del proceso de calificación de atención con la aplicación	42
Figura 23. Arquitectura de la aplicación	52
Figura 24. Diagrama Entidad – Relación de la Base de Datos	51
Figura 25. Prototipo de la pantalla principal	52
Figura 26. Prototipo de la pantalla de inicio de sesión	53
Figura 27. Prototipo de la pantalla de calificación.....	53
Figura 28. Prototipo de la pantalla de información.....	54
Figura 29. Prototipo de la pantalla de clientes	54
Figura 30. Prototipo de generación de turno	55
Figura 31. Prototipo de la pantalla del trabajador	55
Figura 32. Prototipo de atención de turno	56
Figura 33. Prototipo de la pantalla del administrador	57
Figura 34. Prototipo de la pantalla de gestión de datos	57
Figura 35. Comando de creación de proyecto de React en Visual Studio Code.....	62
Figura 36. Interfaz de creación de proyecto en Visual Studio	62
Figura 37. Comando de instalación de React Router	62
Figura 38. Comando de instalación de Axios	63
Figura 39. Comando de instalación de React Datepicker	63
Figura 40. Comando de instalación de jsPDF	63

Figura 41. Comando de instalación de Xslx	63
Figura 42. Comando de instalación de React Paginate	64
Figura 43. Página principal de la aplicación	64
Figura 44. Código C# para listar las sucursales	65
Figura 45. Código JavaScript para el consumo de la API de listar sucursales	65
Figura 46. Página para sacar turnos	66
Figura 47. Código C# para listar tipos de consulta	66
Figura 48. Código JavaScript para el consumo de la API de listar tipos de consulta	66
Figura 49. Ventana para confirmar el turno	67
Figura 50. Ventana de confirmación de generación de turno	67
Figura 51. Código C# para la generación de un turno	68
Figura 52. Código JavaScript para el consumo de la API de nuevo turno.....	68
Figura 53. Código JavaScript para la impresión del ticket	69
Figura 54. Pantalla de información	69
Figura 55. Código C# para listar los turnos por sucursal y estado.....	70
Figura 56. Código JavaScript para el consumo de la API para cargar turnos en espera	70
Figura 57. Código JavaScript para el consumo de la API para cargar turnos que se están atendiendo	70
Figura 58. Selección de ventanilla (Pantalla de calificación)	71
Figura 59. Encuesta de calificación (Pantalla de calificación)	71
Figura 60. Calificación general (Pantalla de calificación).....	72

Figura 61. Código C# para actualizar la calificación.....	72
Figura 62. Código JavaScript para el consumo de la API para actualizar la calificación	73
Figura 63. Pantalla de inicio de sesión.....	73
Figura 64. Código C# para devolver un usuario	74
Figura 65. Código JavaScript para el consumo de la API de inicio de sesión.....	75
Figura 66. Código C# para la creación del token de usuario	75
Figura 67. Código C# para la validación del token.....	76
Figura 68. Código JavaScript para la validación del token.....	76
Figura 69. Pantalla de trabajador	77
Figura 70. Ventana de atención de turno	77
Figura 71. Código JavaScript para cargar la información del turno	78
Figura 72. Código C# para actualizar la atención	79
Figura 73. Código C# para actualizar el estado del turno	79
Figura 74. Código JavaScript para el consumo de las API de atención y turno	80
Figura 75. Pantalla de administrador	81
Figura 76. Pantalla de administración de atenciones	81
Figura 77. Código C# para listar las atenciones.....	82
Figura 78. Código JavaScript para el consumo de la API para listar las atenciones .	82
Figura 79. Página de administración de turnos	83
Figura 80. Código C# para listar todos los turnos.....	83

Figura 81. Código JavaScript para el consumo de la API para cargar los turnos	84
Figura 82. Pantalla de administración de roles	84
Figura 83. Código C# para listar los roles	85
Figura 84. Código JavaScript para el consumo de la API para cargar los roles	85
Figura 85. Comando de construcción de paquete de aplicación	107
Figura 86. Página IIS Servicios.....	108
Figura 87. Archivos Servicios.....	108
Figura 88. Página IIS Aplicación	109
Figura 89. Archivos Aplicación	109

ÍNDICE DE ANEXOS

Anexo A. Cuestionario aplicado al personal de la empresa.....	133
Anexo B. Cuestionario aplicado a los clientes de la empresa.....	134
Anexo C. Guía de entrevista	135
Anexo D. Material extra.....	136
Anexo E. Diccionario de datos de la base de datos.....	159
Anexo F. Manual de usuario	163

RESUMEN EJECUTIVO

El presente proyecto surge en respuesta a la necesidad de iniciar en la EP-EMAPA-A un proceso de mejora y automatización del proceso de generación y atención de turnos en las sucursales, realizado de manera manual, ineficiente e ineficaz, generando limitaciones tanto para los clientes como para los trabajadores. El propósito de la presente investigación fue implementar una aplicación centralizada para la gestión y atención de turnos de las sucursales de la empresa. En este contexto, se utilizaron técnicas de recolección de datos como entrevistas y encuestas para identificar las necesidades y requerimientos para el desarrollo del proyecto. Se utilizó la metodología RAD (Rapid Application Development) con el objetivo de obtener una retroalimentación constante del cliente. Para el desarrollo se utilizó SQL Server como gestor de base de datos y React JS para la aplicación. Se emplearon los modelos TAM y UTAUT para evaluar la aceptación y adopción de la aplicación. Esta propuesta optimizó la generación, asignación y atención de los turnos, además de mejorar la generación de informes con lo que la empresa obtiene información completa del proceso. Además, se abrió la posibilidad a explorar la implementación de un sistema de soporte de decisiones para mejorar el análisis y toma de decisiones.

Palabras clave: Gestión de turnos, TAM, UTAUT, Rapid Application Development (RAD), React, aplicación web, SQL server.

ABSTRACT

The present project arises in response to the need to initiate a process of improvement and automation of the turn generation and service process at EP-EMAPA-A, currently carried out manually, inefficiently, and ineffectively, leading to limitations for both clients and workers. The purpose of this research was to implement a centralized application for managing and attending to turns in the company's branches. In this context, data collection techniques such as interviews and surveys were used to identify the needs and requirements for project development. The RAD (Rapid Application Development) methodology was employed to achieve constant client feedback. SQL Server was used as the database management system, and React JS was employed for the application development. The TAM and UTAUT models were utilized to assess the acceptance and adoption of the application. This proposal optimized the generation, allocation, and handling of turns, while also enhancing the reporting process, providing the company with comprehensive information. Additionally, it opened the possibility of exploring the implementation of a decision support system to improve analysis and decision-making processes.

Keywords: Turn management, TAM, UTAUT, Rapid Application Development (RAD), React, web application, SQL server.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

APLICACIÓN WEB CENTRALIZADA PARA LA GESTIÓN Y ATENCIÓN DE TURNOS DE LAS SUCURSALES DE LA EP-EMAPA-A.

1.1.1 Planteamiento del problema

Las Tecnologías de la Información y Comunicación (TIC) han dejado una profunda huella en Latinoamérica, transformando de manera significativa el panorama socioeconómico de la región. En las últimas décadas, el crecimiento exponencial de la adopción de las TIC ha revolucionado la forma en que los ciudadanos interactúan, trabajan y acceden a la información. Las TIC han impulsado el desarrollo de sectores clave, como el comercio electrónico, la educación en línea y la automatización de procesos empresariales [1].

El impacto de las Tecnologías de la Información y Comunicación (TIC) en Ecuador ha sido significativo, generando una transformación profunda en diversos ámbitos de la sociedad y la economía. Según datos estadísticos, la inversión en TIC ha sido una prioridad tanto para el gobierno como para el sector privado, representado por un porcentaje creciente de las empresas que realizan inversión en TIC. Esta inversión ha impulsado la modernización de los servicios que ofrecen estas empresas, el desarrollo de comercio electrónico y la educación en línea [2].

En la actualidad, las Tecnologías de la Información y Comunicación (TIC) han tenido un impacto significativo en diversos sectores de la sociedad, transformando la manera en que interactuamos y accedemos a servicios. Dentro del sector público, en particular, las TIC han jugado un papel fundamental en la gestión y atención de turnos, optimizando los procesos de asignación y proporcionando a los usuarios una experiencia más ágil y satisfactoria [3].

En el cantón Ambato, en la Empresa Pública Municipal de Agua Potable y Alcantarillado (EP-EMAPA-A), la gestión de los turnos en las sucursales de la empresa se realiza de manera independiente y descentralizada. Esto genera diversos inconvenientes y limitaciones para los usuarios y trabajadores encargados de atender los turnos, como la dificultad para el manejo de la información y como consecuencia la desinformación de los clientes. Además, debido a la descentralización del proceso y la ausencia de un sistema actualizado, no existe disponibilidad de información sobre el proceso de todas las sucursales de manera global.

1.2 Antecedentes investigativos

Después de haber realizado una investigación sobre proyectos y tesis en diferentes buscadores especializados para este tipo de documentos, como Dialnet, RENATI, etc., se encontraron tesis y trabajos similares referentes al diseño de un sistema de gestión de turnos.

Según R. Cespedes [4] en su tesis ‘Sistema de información para mejorar la gestión de turnos en el área de operaciones de Latina TV’, menciona que:

- “A través del uso preciso de las herramientas de la Tecnología de la Información y el Desarrollo de Software, al incorporarlas como parte del proceso de gestión administrativa, contribuirá al desarrollo de la empresa, agilizando gran parte de las tareas involucradas con tendencia a ser repetitivas, eliminando la dependencia manual y logrando una mayor rendimiento y autonomía de gestión de la información y turnos”.

Según J. Molina [5] en su tesis ‘Desarrollo e implantación del sistema de gestión de turnos para consulta externa del Hospital de especialidades Eugenio Espejo’, menciona que:

- “Con el desarrollo e implantación de un sistema de turnos, se facilitará la labor al departamento de Estadística, al centralizar la administración de turnos, mejorar el proceso de distribución, optimizar el manejo de la información y acabar con el malestar existente por parte de cientos de pacientes de consulta externa que no obtienen su turno de manera rápida y eficiente”.

Según J. Bazán [6] en su tesis ‘Desarrollo de aplicación Web para gestión de turnos en la escuela de educación básica particular Monseñor Roberto María del Pozo’, menciona que:

- “El desarrollo de una aplicación que gestione los turnos de la escuela permitirá realizar una transición y transformación que optimizará el proceso de generación y distribución, terminando con las atenciones ineficientes con largos tiempos de espera y mejorando las gestiones y atenciones de los diferentes departamentos de la institución”.

Según B. Chávez y A. Adelina [7] en su trabajo de suficiencia profesional ‘Mejoras al sistema de gestión de turnos en una empresa de telecomunicaciones’, mencionan que:

- “A partir de las mejoras realizadas, el sistema de gestión de turnos permitió recuperar la confianza en la información de los reportes obtenidos, así como en la gestión de los tickets de atención y un comportamiento estable en los servidores que alojaban el sistema, consiguiendo una mejor percepción de los usuarios con respecto al sistema y poniendo en evidencia las ventajas de un sistema de gestión de turnos que funcione correctamente”.

1.3 Fundamentación teórica

1.3.1 Sistema de turnos

Un sistema de turnos es una herramienta que se utiliza para organizar y administrar de una manera eficiente el flujo de personas que requieren de servicios o actividades en un determinado lugar como una empresa, institución o entidad pública. Tienen una gran importancia para cualquier sector que atiende a un gran número de personas. Son utilizados en diversos sectores, como la salud, el sector público y los servicios, con el objetivo principal de evitar largos tiempos de espera y así mejorar la calidad de atención al cliente [8].

1.3.2 Sistema centralizado

Se denomina sistema centralizado al sistema en el cual todas las funciones, datos y recursos están concentrados en una ubicación central. La centralización implica que todas las decisiones y operaciones que se lleven a cabo se realizan desde dicha ubicación y las otras partes del sistema dependen de ella para adquirir los recursos y funciones necesarias, ofreciendo uniformidad, coherencia y eficiencia en las operaciones [9].

Así mismo la centralización puede implicar la integración de múltiples programas o aplicaciones en una única plataforma centralizada, con el objetivo de eliminar duplicidades y redundancias en la funcionalidad, y evitar el mantenimiento de múltiples programas separados.

1.3.3 Sistemas de gestión

Los sistemas de gestión son herramientas utilizadas para administrar y controlar distintos aspectos dentro de una organización, empresa o entidad, proporcionando un enfoque estructurado y eficiente en la gestión de actividades, recursos, información y operaciones en varias áreas [10].

Los sistemas abarcan diferentes áreas funcionales, como la calidad, el inventario, los recursos humanos, las finanzas, los clientes, entre otros, teniendo que manejar grandes cantidades de datos que ofrecen información importante para que las entidades puedan tomar decisiones y automatizar procesos.

Dentro de la variedad de sistemas de gestión se encuentran los siguientes: SGD (Sistemas de Gestión de Documentos), utilizados para administrar y controlar el flujo de una gran cantidad de documentos dentro de una organización [11]; ERP (Sistemas de Planificación de Recursos Empresariales), que permiten a las entidades administrar el flujo de datos entre procesos [12]; y CRM (Sistemas de Gestión de Relación con el Cliente), herramientas para administrar y mejorar las interacciones con los clientes [13].

1.3.4 Framework

Un framework de desarrollo es una estructura o conjunto de herramientas, librerías y patrones de diseño predefinidos que brindan una base consistente para la creación de aplicaciones de software. Los frameworks permiten a los desarrolladores enfocarse en la lógica de negocio de la aplicación al proporcionarles herramientas y estructuras que simplifican y agilizan el proceso de desarrollo [14].

a. Angular

Angular es un framework de desarrollo de aplicaciones Web de código abierto escrito en TypeScript, desarrollado por Google y lanzada por primera vez en 2016. Se ha convertido en uno de los frameworks más populares y utilizados en el desarrollo de aplicaciones web debido a su potencia y versatilidad [15].

Está basado en componentes para la creación de aplicaciones Web SPA (Single Page Application), cuenta con un enfoque bidireccional de datos, permitiendo que cualquier cambio que se realice en los datos, se verá reflejado de manera automática en la interfaz de usuario que se carga una sola vez. Proporciona un sistema de enrutamiento que permite navegar entre diferentes componentes de la aplicación.

Ofrece la capacidad de realizar pruebas unitarias e integradas, a partir de las herramientas y utilidades que facilitan la escritura y ejecución de pruebas, detectando y corrigiendo errores de manera eficiente, garantizando la calidad del código.

b. React

React es una biblioteca de JavaScript que se conoce comúnmente como un framework para la construcción de interfaces de usuario, es de código abierto, fue desarrollado por Facebook y se lanzó por primera vez en 2013. Desde su lanzamiento se ha convertido en uno de los frameworks más populares y utilizados para el desarrollo web [16].

Utiliza un enfoque basado en componentes, priorizando la creación de componentes reutilizables, los cuales representan diferentes partes de la interfaz de usuario de la

aplicación y se pueden combinar para construir interfaces de usuarios más complejas, lo que facilita su creación, mantenimiento y reutilización de código.

Utiliza un algoritmo llamado “Virtual DOM” que realiza actualizaciones de manera selectiva, minimizando la manipulación directa del DOM, lo que mejora el rendimiento de la aplicación al reducir la cantidad de operaciones de actualización realizadas por el DOM.

c. Laravel

Laravel es un framework de desarrollo Web de código abierto basado en PHP, desarrollado por Taylor Otwell y lanzado por primera vez en 2011. Se ha convertido en un framework bastante popular para el desarrollo web debido a que proporciona una manera rápida, eficiente y estructurada de construir aplicaciones [17].

Utiliza una arquitectura MVC (Modelo-Vista-Controlador), lo que facilita separar la lógica de negocio de la presentación. Permite la gestión de bases de datos con su sistema de migraciones, que permite a los desarrolladores definir y controlar la estructura de la base de datos a través de código y migrar esa estructura para crear una base de datos completa.

Integra un sistema de autenticación completo que incluye funciones predefinidas para el registro, inicio de sesión, restablecimiento de contraseñas y protección de rutas, personalizable y configurable dependiendo de las necesidades de la aplicación.

1.3.5 Sitio Web

Un sitio Web es una colección de páginas Web interconectadas a través de una dirección URL (Uniform Resource Locator) y que son accesibles a través de Internet. Cada página Web es un conjunto de archivos electrónicos que contienen información como imágenes, vídeos, enlaces y otros elementos que se muestran de manera visual y organizada [18].

Para acceder a una página Web se utilizan navegadores Web, que permiten visualizar e interactuar con su contenido, creado a partir de lenguajes de marcado como HTML (HyperText Markup Language). Las páginas Web tienen una serie de características

que deben cumplir para que el usuario tenga una buena experiencia al interactuar con ellas, características como:

- Usabilidad: La página Web debe ser fácil de usar y navegar con elementos bien organizados que permitan al usuario encontrar la información de manera rápida.
- Accesibilidad: La página Web debe ser accesible para todos los usuarios y seguir pautas de accesibilidad web, como el uso de etiquetas adecuado y opciones de navegación alternativas.
- Velocidad de carga: La página Web debe cargar rápidamente y evitar tiempos de espera prolongados.

1.3.6 Base de datos

Las Bases de datos son sistemas organizados en los cuáles se almacena una colección de datos estructurados y relacionados entre sí para permitir el almacenamiento eficiente, la recuperación, la gestión y la manipulación de los datos [19].

En las Bases de datos, los datos se organizan en tablas, que contienen filas, que representan un registro único y columnas que representan un campo específico o atributo de ese conjunto de datos. Esto permite realizar consultas, filtraciones y ordenamiento de datos y generación de informes.

Los SGBD (Sistemas de Gestión de Bases de Datos) son herramientas diseñadas para administrar y gestionar las bases de datos, ocupan un lenguaje de definición de datos (SQL en el caso de bases de datos relacionales), para definir la estructura, restricciones y tipos de datos. Entre los diferentes SGBD se encuentran los siguientes:

- MySQL.
- Oracle.
- Microsoft SQL Server.
- PostgreSQL.

- SQLite.

1.3.7 Servidor Web

Un servidor Web es un programa o computadora que brinda servicios de alojamiento y entrega de contenido Web a través de internet. Actúa como intermediario entre el cliente y el sitio Web, su función es recibir solicitudes de clientes (navegadores Web) y proporcionar los recursos solicitados que están guardados en el servidor a través del protocolo HTTP (HyperText Transfer Protocol) o HTTPS (HyperText Transfer Protocol Secure) [20].

Entre los servidores Web más importantes se encuentran:

- Apache HTTP Server.
- Nginx.
- Microsoft Internet Information Services (IIS).
- Google Web Server.

1.3.8 Metodologías de desarrollo

Las metodologías de desarrollo son enfoques organizados empleados en el ámbito tecnológico y de desarrollo de software que permiten la gestión de proyectos de manera eficiente. Brindan una serie de buenas prácticas y establecen un marco de trabajo de roles y responsabilidades que ayuda a los equipos a trabajar de manera más efectiva y estructurada, fomentando la comunicación y colaboración entre los miembros [21].

a. Metodologías clásicas

- *Modelo en cascada*

El modelo en cascada es un enfoque de desarrollo de software basado en una secuencia lineal y jerárquica de etapas bien definidas a la hora de llevar a cabo un proyecto [22].

Se compone de varias etapas como:

- **Análisis de requisitos:** A través de consultas con el usuario se analizan las necesidades, restricciones y metas del sistema.
- **Diseño:** Se establece la arquitectura del sistema y se describen partes indispensables del producto final.
- **Implementación:** Se crea el producto de software o sistema realizando todas las tareas necesarias para su funcionamiento en un entorno de producción.
- **Integración y verificación:** Se integran todas las partes del sistema y se prueban como un sistema completo para verificar que se cumplen todos los requisitos.
- **Mantenimiento:** Se realizan correcciones del sistema, que no fueron descubiertos en etapas anteriores, se mejora y se adapta a los nuevos requisitos.

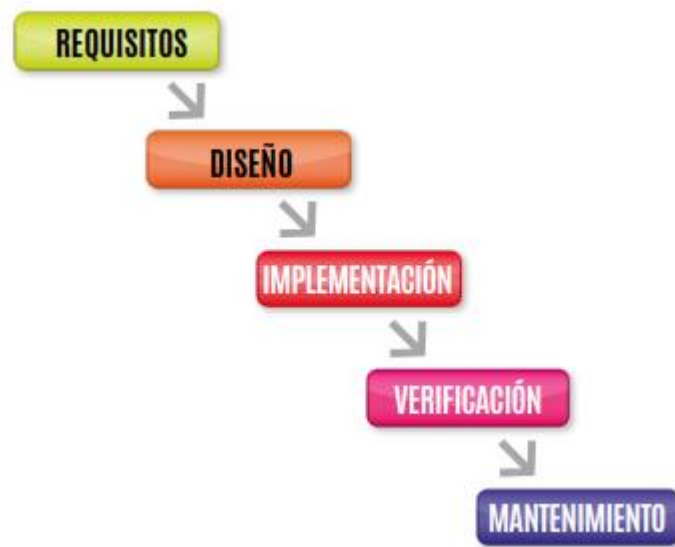


Figura 1. Etapas modelo en cascada [22]

Al ser una secuencia lineal se centra en completar una etapa antes de pasar a la siguiente, y cualquier cambio en una etapa posterior puede conllevar a la necesidad de retroceder y realizar los cambios necesarios en las etapas que lo requieran. Este enfoque se utiliza en proyectos donde los requisitos están bien definidos y no son cambiantes durante el desarrollo.

- **Modelo en espiral**

El modelo en espiral es un enfoque de desarrollo de software que se basa en la gestión de riesgos a través de la planificación y desarrollo de ciclos iterativos para conseguir una mejor adaptabilidad y flexibilidad durante el proyecto [23]. La estructura del modelo en espiral se compone de 4 fases cíclicas:

- **Planificación:** Se establecen los objetivos y requerimientos del proyecto, además de las metas del ciclo.
- **Análisis de riesgo:** Se analizan y evalúan los riesgos del proyecto y se establecen estrategias para gestionarlos.
- **Ingeniería:** Se lleva a cabo el desarrollo del software en base a los objetivos y requerimientos establecidos.
- **Evaluación:** Se evalúa el software para verificar si cumple con los objetivos y se planifica la siguiente iteración.

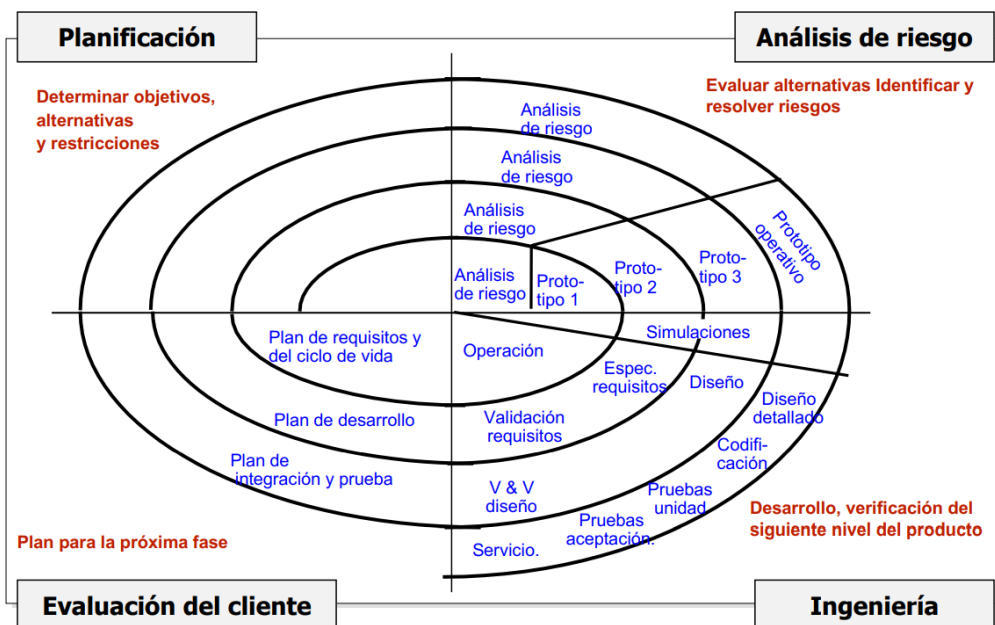


Figura 2. Fases del modelo en espiral [23]

Se utiliza en proyectos de software que tienden a ser cambiantes, con requerimientos no definidos completamente y que son propensos a cambios significativos. No sigue

un proceso lineal, lo que permite a los equipos de desarrollo adaptarse a cambios durante el desarrollo del proyecto.

b. Metodologías ágiles

- **RAD (Rapid Application Development)**

La metodología RAD es un enfoque de desarrollo de software caracterizado por su énfasis en la entrega rápida y el prototipado iterativo [24]. El ciclo de vida de RAD es:

- **Recopilación de requisitos:** Se obtienen los requisitos esenciales del sistema a través de entrevistas o encuestas.
- **Prototipado:** Se desarrolla un prototipo funcional del sistema en base a los requisitos para obtener una retroalimentación del usuario.
- **Construcción:** Se construye el sistema de manera iterativa, con construcción y prueba en bloques pequeños.
- **Pruebas:** Se realizan pruebas del sistema completo para verificar que cumple con los requisitos finales.

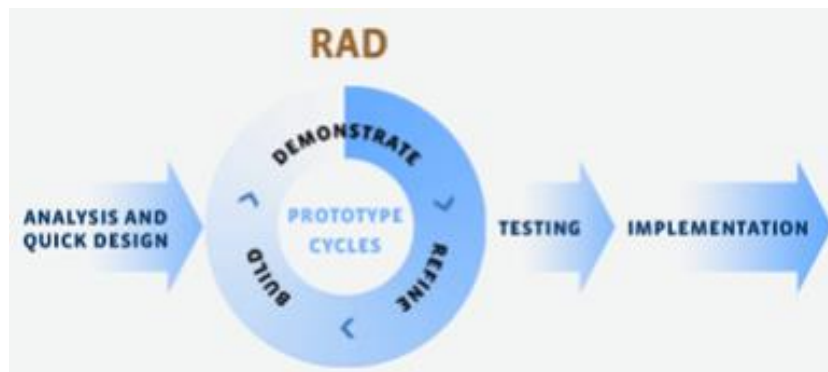


Figura 3. Ciclo de vida de RAD [24]

Se enfoca en la participación continua de los usuarios desde las primeras etapas y en la construcción rápida de prototipos funcionales que permitan a los desarrolladores y usuarios la evaluación y el ajuste del sistema. Esta metodología se utiliza para proyectos con requisitos que no están completamente definidos o son muy propensos a cambios, buscando mejorar la adaptabilidad y satisfacción del cliente.

- *Scrum*

La metodología Scrum es un enfoque ágil para la gestión de proyectos basado en principios de flexibilidad y colaboración a la hora de llevar a cabo proyectos complejos y cambiantes [25]. El ciclo de vida de Scrum es:

- **Planeación del Sprint:** El equipo selecciona las tareas del Backlog para el próximo sprint y establece los objetivos y duración del sprint.
- **Reunión de equipo:** Se realiza una reunión para que el equipo comparta actualizaciones sobre el progreso de cada uno.
- **Refinamiento de Backlog:** El equipo revisa y refina el Backlog del producto para asegurarse de que las tareas estén bien definidas para futuros sprints.
- **Revisión del Sprint:** El equipo demuestra el trabajo completado durante el sprint al Product Owner para obtener una retroalimentación.
- **Retrospectiva:** Se realiza una reunión en la que el equipo discute sobre el sprint anterior para identificar mejoras.



Figura 4. Ciclo de vida de Scrum [25]

La estructura de Scrum incluye roles, como el Product Owner, que representa al cliente, el Scrum Master, que ayuda al equipo a seguir la metodología y el

Development Team, que lleva a cabo el desarrollo. La flexibilidad de Scrum hace que sea una metodología utilizada para proyectos dinámicos.

- **XP (Extreme Programming)**

La metodología XP es un enfoque ágil de desarrollo de software centrada en la entrega de software de calidad de una manera rápida [26]. XP define etapas como:

- **Planificación:** Se identifican las historias de usuario para describir las funcionalidades del sistema, se priorizan y se establece un plan en función de la prioridad.
- **Diseño:** Se crea un diseño simple y eficiente para las funcionalidades seleccionadas.
- **Codificación:** Se implementa el código del sistema en función del diseño creado.
- **Pruebas:** Se realizan pruebas unitarias de componentes para asegurar la calidad y detectar errores del sistema.
- **Lanzamiento:** Se entregan versiones funcionales del sistema en ciclos cortos y regulares para obtener una retroalimentación del usuario.

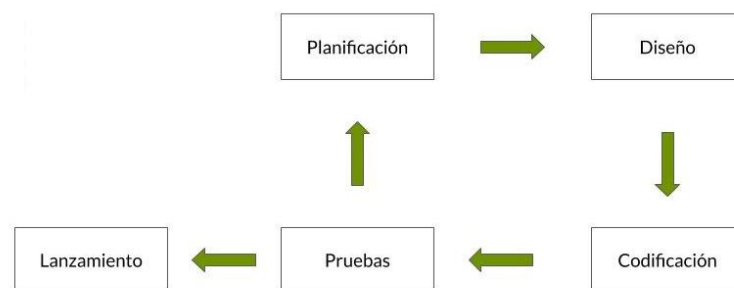


Figura 5. Etapas de XP

Se centra en la colaboración continua de los usuarios desde las etapas iniciales, obteniendo una retroalimentación constante y ajustando el sistema en base a ello. Este enfoque se aplica en proyectos con requisitos que tienden a cambiar o evolucionar.

1.4 Objetivos

1.4.1 Objetivo general

Implantar una aplicación Web para la centralización de la gestión y atención de los turnos en las sucursales de la EP-EMAPA-A.

1.4.2 Objetivos específicos

- Analizar el proceso de gestión y atención de turnos en las sucursales de EP-EMAPA-A.
- Determinar las herramientas y tecnologías necesarias para el desarrollo de la aplicación.
- Desarrollar una aplicación Web para la centralización de la gestión y atención de los turnos en las sucursales de EP-EMAPA-A.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Este trabajo de investigación utilizó una entrevista en conjunto con los representantes del departamento de TICS y de la Unidad de Atención al Cliente de la EMAPA, como método para entender los procesos mediante los cuales se manejan la gestión y la atención de los turnos dentro de la institución, específicamente dentro de la sucursal “La Matriz”.

Adicionalmente se realizó un cuestionario a los empleados encargados de la gestión y atención de los turnos de la EMAPA para obtener una vista general de los trabajadores con respecto a la manera en la que se maneja la información.

2.2 Métodos

2.2.1 Modalidad de la investigación

a. Modalidad de campo

La modalidad de campo se trata de un método de investigación que implica una recopilación de datos de manera directa en un lugar de estudio o entorno específico. En este caso se trabajó con los representantes y empleados de la institución mediante entrevistas y encuestas sobre los requisitos relacionados con la gestión y atención de turnos.

b. Modalidad bibliográfica

La modalidad bibliográfica se trata de un método de investigación basado en la revisión y análisis sobre la literatura existente acerca de un campo de estudio en concreto. En este caso la investigación bibliográfica se llevó a cabo para examinar las fuentes científicas encontradas en libros, tesis y artículos científicos relacionados a las variables de estudio. Debido a esto se identifica las mejoras prácticas y tendencias

sobre la gestión de turnos y la atención de los mismo basadas en opiniones de personas entendidas del tema.

2.2.2 Población y muestra

a. Población

La población de estudio del presente proyecto se divide en 2:

- Los trabajadores de la Empresa que se encargan de atender los turnos y generar reportes, que serán aproximadamente 12 personas.
- Las personas que requieren los turnos para que sean atendidos, que serán aproximadamente 40 personas al día.

b. Muestra

La muestra representativa será el conjunto de total de personas que forman parte de la Empresa siendo los trabajadores que ejercen sus labores en la actualidad y las personas que son atendidas en la entidad:

El tamaño de la muestra representativa es de 52 personas.

2.2.3 Recolección de información

Resultados de la entrevista a los representantes de la EP-EMAPA-A.

Tabla 1. Resultados de la entrevista

Estimados representantes			
La información recolectada se usará con fines académicos.			
Instrucciones: Responder honestamente las preguntas			
Entrevistador: Javier Jiménez			
N.º	Pregunta	Respuesta	Observación
1	¿Quiénes intervienen actualmente en la gestión y atención de turnos?	El personal de balcón de servicios es el encargado de atender a los usuarios que previamente solicitaron su turno acercándose al personal de información.	El personal de balcón de servicios no cuenta con una herramienta actualizada para atender al usuario.

N.º	Pregunta	Respuesta	Observación
			La atención se realiza por orden de llegada sin ningún ticket.
2	¿Qué problemas se han presentado con el proceso actual?	El sistema actual instalado ya no se utiliza porque se encuentra desactualizado y no cumple con las necesidades de los clientes ni de la empresa.	Con el proceso manual no se pueden realizar ciertas actividades que serían un gran aporte a la empresa (Reportes, calificaciones, mediciones de tiempo)
3	¿En qué aspectos creen que mejoraría la situación la aplicación web de gestión y atención de turnos?	Permitiría controlar y medir el tiempo de atención a los usuarios, saber cómo es el procedimiento en todas las agencias en las que tenemos balcón de servicios y obtener información de manera global y específica.	La aplicación brindaría información importante a la empresa y mejorará el proceso actual.
4	¿Qué características debe tener la aplicación web de gestión y atención de turnos?	Debe permitir a los usuarios sacar turnos para los distintos servicios que presta la empresa. Debe funcionar para todas las agencias en las que tengamos un balcón de servicios. Proporcionar información sobre el proceso de manera global de todas las agencias o de una agencia en específico.	Debe ser una aplicación desarrollada con determinadas tecnologías utilizadas por la empresa.
5	¿Cuál es el objetivo de implementar la aplicación web?	Facilitar al usuario la generación y atención de su turno. Facilitar las labores de los trabajadores encargados de la atención en las agencias. Obtener información sobre el proceso que será importante para la empresa.	Se busca de igual manera una aplicación que funcione en todas las agencias.

N.º	Pregunta	Respuesta	Observación
6	¿Qué tipo de consultas se van a atender con la aplicación web?	Consulta de valores de planillas, valores a cancelar, actualización de datos, obtención de servicios, consultas generales, actualización de servicios.	
7	¿Qué características debe cumplir la aplicación web para cubrir con este tipo de consultas?	La aplicación debe permitir añadir o desactivar un tipo de consulta según las valoraciones de la empresa y en consecuencia permitir a los usuarios generar turnos solo para los tipos de consultas que estén activos.	Se busca diferencias claramente el tipo de consulta para el que se sacó un turno en específico. Los datos no se deben eliminar completamente, si no que se debe manejar a través de un estado.
8	¿Cómo piensan que mejoraría la atención al cliente la implementación de la aplicación web?	Agilizaría tanto los procesos de generación de turnos, así como el de atención al cliente al brindar al trabajador una herramienta acorde con los requerimientos de la empresa.	Mejoraría varios aspectos tanto del lado de la atención al cliente y de la empresa, como del cliente.

Conclusión:

En base a las respuestas de la entrevista, se concluye que el sistema actual ya no se utiliza debido a que está desactualizado y no cumple con las necesidades de la empresa, lo que obliga a la empresa a manejar todo el proceso de manera manual, demorando el proceso de generación y atención de turnos. Por lo cual, se evidencia que el desarrollo de una aplicación web satisfaría las necesidades de la empresa y mejoraría todo el proceso.

Cálculo del Alfa de Cronbach

El Alfa de Cronbach se utilizó para evaluar la confiabilidad del cuestionario realizado, ya que este cálculo permitió evaluar si los resultados obtenidos no están sesgados. Si el resultado es mayor o igual al 0,7 indica que el cuestionario tiene una mayor fiabilidad.

Para calcular el alfa de Cronbach se obtiene la varianza de cada pregunta para esto se aplicó la siguiente fórmula:

$$VAR = \sum \frac{[(X - \mu)^2]}{N}$$

X → representa el valor individual del conjunto de datos.

μ → es la media de los valores del conjunto de datos.

N → cantidad total de datos.

- **Encuesta realizada al personal de la EMAPA**

Tabla 2. Resultados de las varianzas de la encuesta a los trabajadores

Pregunta	Varianza
¿Cómo califica la gestión actual de la información de los turnos?	0,75
¿Con qué frecuencia interactúa con los clientes que solicitan turnos para ser atendidos?	0,6875
¿Qué métodos o sistemas utiliza actualmente para administrar la información de los turnos?	0,2431
¿Ha tenido dificultades para manejar la información de los turnos?	1,3056
¿Con qué frecuencia los clientes han expresado alguna queja o comentario sobre el proceso de solicitud de turnos?	0,8542
¿Se siente satisfecho con el proceso de administración de turnos de la institución?	0,5
¿Cree que una aplicación de gestión y atención de turnos puede mejorar el proceso actual?	0,1389

- **Encuesta realizada a las personas que son atendidas en la EMAPA**

NOTA: Las personas a las que se les realizó la encuesta se escogieron de manera aleatoria en diferentes días y en el periodo de una semana.

Tabla 3. Resultados de las varianzas de la encuesta a los clientes

Pregunta	Varianza
¿Cómo califica la calidad de la atención?	0,7744
¿Con qué frecuencia utiliza los servicios de atención al cliente de la entidad?	0,81
¿Qué nivel de satisfacción tiene con respecto al tiempo para generar su turno?	1,0594
¿Qué nivel de satisfacción tiene con respecto al tiempo de espera para recibir atención?	1,0944
¿Cuánto tiempo ha esperado para que se atienda su turno?	0,4594
¿En qué medida cree que el proceso de generación de turnos es accesible y cómodo?	0,79
¿Qué cantidad de información sobre la atención de su turno cree que se proporciona?	0,4975

Para calcular el Alfa de Cronbach se aplica la siguiente fórmula:

$$\alpha = \frac{K}{K - 1} * \left[1 - \frac{\sum Vi}{Vt} \right]$$

Donde:

K → número de ítems

$\sum Vi$ → sumatoria total de la varianza de cada ítem

Vt → varianza total

Para el cálculo del nivel de confiabilidad al utilizar el coeficiente alfa de Cronbach se tienen en cuenta el nivel de confiabilidad en base al rango [27]:

Tabla 4. Rangos de confiabilidad [27]

Rango	Confiabilidad
0,30 a menos	Confiabilidad nula
0,31 a 0,50	Confiabilidad baja
0,51 a 0,70	Confiable
0,71 a 0,90	Muy confiable

Rango	Confiabilidad
0,91 a 0,99	Excelente confiabilidad
1	Confiabilidad perfecta

a. Nivel de confiabilidad de la encuesta al personal

K → número de ítems = 7

$\sum Vi$ → sumatoria total de la varianza de cada ítem = 4,4793

Vt → varianza total = 12,9097

$$\alpha = \frac{7}{7-1} * \left[1 - \frac{4,4793}{12,9097}\right]$$

$$\alpha = 0,7618664$$

El valor del alfa de Cronbach para la encuesta realizada a los trabajadores de la EMAPA-A es mayor a 0,7, por lo que basándose en los rangos de la *Tabla 4* se puede concluir que existe consistencia en el cuestionario y las preguntas están correlacionadas entre sí.

- **Resultados de la encuesta aplicada al personal**

1. ¿Cómo califica la gestión actual de la información de los turnos?

Tabla 5. Tabla de frecuencias – Encuesta 1 - Pregunta 1

Indicador	Muy mala	Mala	Regular	Buena	Muy buena	Suma total
Frecuencia	2	3	6	1	0	12
Porcentaje	16,67%	25,00%	50,00%	8,33%	0,00%	100,00%

¿Cómo califica la gestión actual de la información de los turnos?

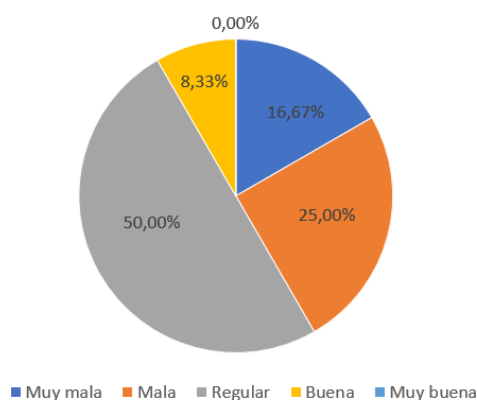


Figura 6. Resultados Encuesta 1 – Pregunta 1

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 6 indican que solo una minoría del personal representado con un 8,33% califica la gestión de la información como buena, mientras que un 16,67% la califica como muy mala, seguido de un 25% como mala y la mayoría del personal representado con un 50% la catalogan como regular, evidenciando una falta de uniformidad con respecto a la calidad de la gestión por la falta de un método eficaz.

2. ¿Con qué frecuencia interactúa con los clientes que solicitan turnos para ser atendidos?

Tabla 6. Tabla de frecuencias – Encuesta 1 – Pregunta 2

Indicador	Con mucha frecuencia	Con frecuencia	Nunca	Ocasionalmente	Raramente	Suma total
Frecuencia	1	3	6	2	0	12
Porcentaje	8,33%	25,00%	50,00%	16,67%	0,00%	100,00%

¿Con qué frecuencia interactúa con los clientes que solicitan turnos para ser atendidos?

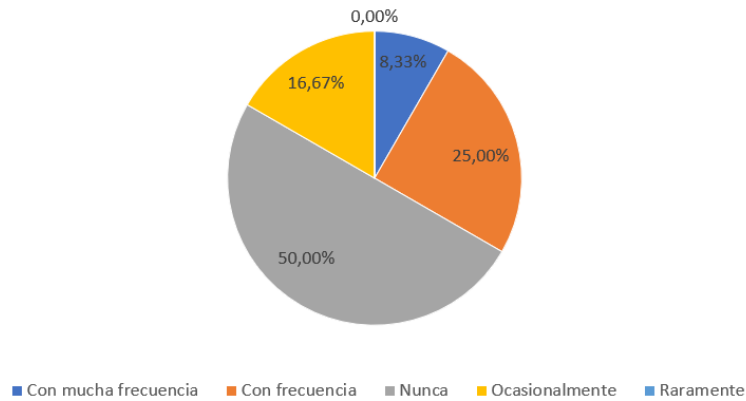


Figura 7. Resultados Encuesta 1 – Pregunta 2

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 7 indican que solo una minoría representada con el 8,33% interactúa con mucha frecuencia con los clientes que solicitan turnos, seguido de un 16,67% que interactúa ocasionalmente, un 25% con frecuencia y un 50% que nunca interactúa, lo que demuestra la falta de automatización en la generación de turnos debido a la falta de un sistema.

3. ¿Qué métodos o sistemas utiliza actualmente para administrar la información de los turnos?

Tabla 7. Tabla de frecuencias – Encuesta 1 - Pregunta 3

Indicador	Aplicación móvil	Hojas de Excel	Hojas físicas	Sistema especializado	Suma total
Frecuencia	0	7	5	0	12
Porcentaje	0,00%	58,33%	41,17%	0,00%	100,00%

¿Qué métodos o sistemas utiliza actualmente para administrar la información de los turnos?

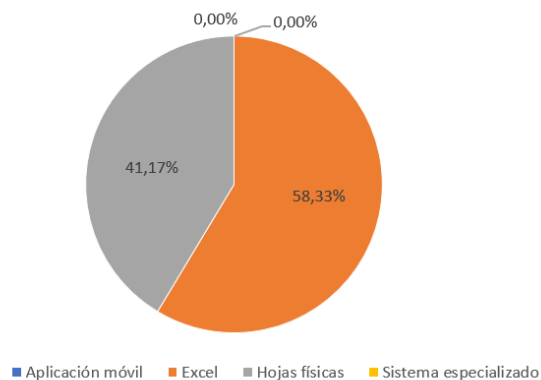


Figura 8. Resultados Encuesta 1 – Pregunta 3

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 8 indican que los métodos utilizados son con un 41,17% las hojas físicas, seguido del método más utilizado con un 58,33% siendo hoja de Excel, descartando métodos como el uso de una aplicación móvil o un sistema especializado representados en ambos casos con un 0% y evidenciando la falta de un sistema especialmente enfocado al proceso.

4. ¿Ha tenido dificultades para manejar la información de los turnos?

Tabla 8. Tabla de frecuencias – Encuesta 1 – Pregunta 4

Indicador	Con mucha frecuencia	Con frecuencia	Nunca	Ocasionalmente	Raramente	Suma total
Frecuencia	4	5	0	3	0	12
Porcentaje	33,33%	41,17%	0,00%	25,00%	0,00%	100,00%

¿Ha tenido dificultades para manejar la información de los turnos?

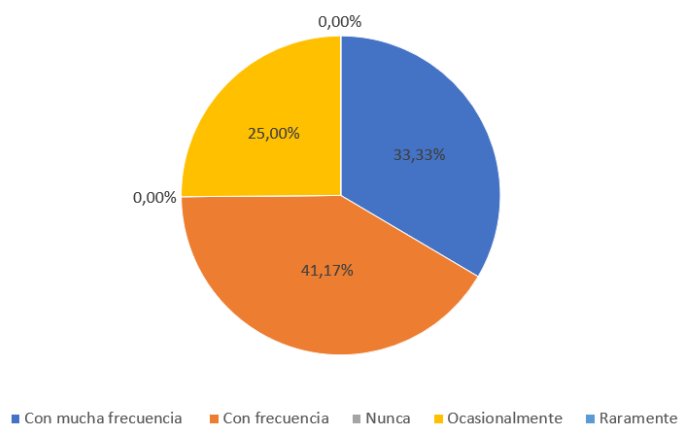


Figura 9. Resultados Encuesta 1 – Pregunta 4

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 9 indican que un 25% del personal ha tenido dificultades para manejar la información de manera ocasional, seguido de un 33,33% que las ha tenido con mucha frecuencia y por último un 41,17% con frecuencia, reflejando una clara falencia al concluir que todo el personal ha tenido dificultades al menos de manera ocasional y la mayoría con frecuencia debido a la falta de un método efectivo para el manejo de la información.

5. ¿Con qué frecuencia los clientes han expresado alguna queja o comentario sobre el proceso de solicitud de turnos?

Tabla 9. Tabla de frecuencias – Encuesta 1 - Pregunta 5

Indicador	Con mucha frecuencia	Con frecuencia	Nunca	Ocasionalmente	Raramente	Suma total
Frecuencia	0	2	1	7	2	12
Porcentaje	0,00%	16,67%	8,33%	58,33%	16,67%	100,00%

¿Con qué frecuencia los clientes han expresado alguna queja o comentario sobre el proceso de solicitud de turnos?

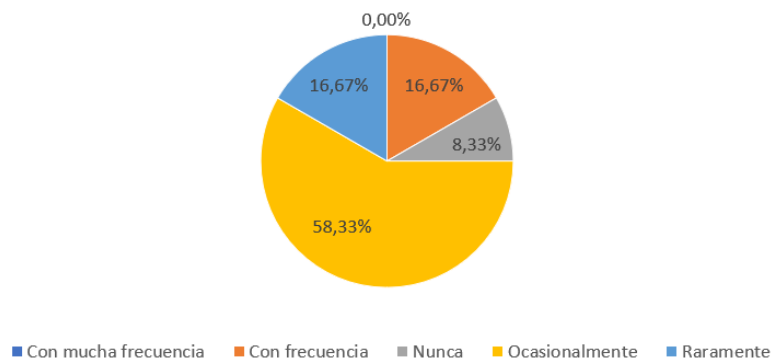


Figura 10. Resultados Encuesta 1 – Pregunta 5

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 10 indican que solo un 8,33% del personal nunca ha recibido alguna queja por parte de los clientes, seguido de un 16,67% que raramente ha recibido quejas, otro 16,67% que ha recibido quejas con frecuencia y un 58,33% que las ha recibido de manera ocasional.

Los datos reflejan que solo una minoría del personal nunca ha recibido quejas sobre el proceso de generación y atención de turnos, mientras que la mayoría ha recibido quejas al menos una vez, evidenciando cierto nivel de molestia por parte de los clientes debido a la falta de una herramienta automatizada.

6. ¿Se siente satisfecho con el proceso de administración de turnos de la institución?

Tabla 10. Tabla de frecuencias – Encuesta 1 – Pregunta 6

Indicador	Muy satisfecho	Satisfecho	Neutral	Insatisfecho	Muy insatisfecho	Suma total
Frecuencia	0	3	6	3	0	12
Porcentaje	0,00%	25,00%	50,00%	25,00%	0,00%	100,00%

¿Se siente satisfecho con el proceso de administración de turnos de la institución?

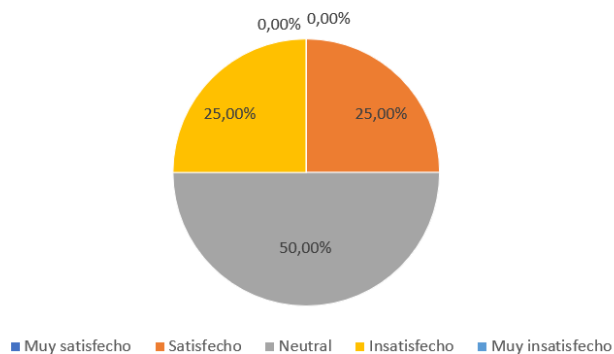


Figura 11. Resultados Encuesta 1 – Pregunta 6

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 11 indican que ningún porcentaje del personal se siente muy satisfecho o muy insatisfecho con el proceso, seguido de un 25% que se siente satisfecho, otro 25% que se siente insatisfecho y un 50% que se mantiene neutral, mostrando una clara falta de uniformidad con respecto al proceso desde el punto de vista de los trabajadores.

7. ¿Cree que una aplicación de gestión y atención de turnos puede mejorar el proceso actual?

Tabla 11. Tabla de frecuencias – Encuesta 1 – Pregunta 7

Indicador	No	No estoy seguro/a	Si	Suma total
Frecuencia	0	2	10	12
Porcentaje	0,00%	16,67%	83,33%	100,00%

¿Cree que una aplicación de gestión y atención de turnos puede mejorar el proceso actual?

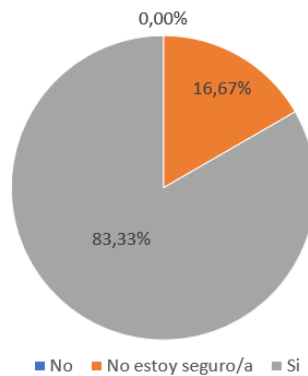


Figura 12. Resultados Encuesta 1 – Pregunta 7

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 12 indican que no hay ningún porcentaje del personal que no crea que el proceso mejoraría con el desarrollo de una aplicación, seguido de un 16,67% que no están seguro/as y seguido por la mayoría representada con un 83,33% que dicen que si mejoraría el proceso con el desarrollo de una aplicación.

b. Nivel de confiabilidad de la encuesta a clientes

$K \rightarrow$ número de ítems = 7

$\sum Vi \rightarrow$ sumatoria total de la varianza de cada ítem = 5,485

$Vt \rightarrow$ varianza total = 13,8275

$$\alpha = \frac{7}{7-1} * \left[1 - \frac{5,485}{13,8275} \right]$$

$$\alpha = 0,703881154$$

El valor del alfa de Cronbach para la encuesta realizada a los clientes de la EMAPA-A es mayor a 0,7, por lo que basándose en los rangos de la Tabla 4 se puede concluir

que existe consistencia en el cuestionario y las preguntas están correlacionadas entre sí.

- **Resultados de la encuesta aplicada a los clientes**

1. **¿Cómo califica la calidad de la atención?**

Tabla 12. Tabla de frecuencias – Encuesta 2 – Pregunta 1

Indicador	Muy mala	Mala	Regular	Buena	Muy buena	Suma total
Frecuencia	2	8	21	7	2	40
Porcentaje	5,00%	20,00%	52,50%	17,50%	5,00%	100,00%

¿Cómo califica la calidad de la atención?

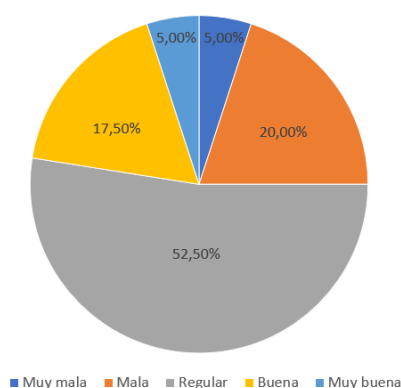


Figura 13. Resultados Encuesta 2 – Pregunta 1

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 13 indican que un 5% de los clientes opinan que la calidad de la atención es muy mala, seguido de otro 5% que opina que es muy buena, un 17,5% que opina que es buena, un 20% que indica lo contrario y por último un 52,5% que califica la calidad de la atención como regular, evidenciando una disconformidad con respecto a la calidad de la atención por la falta de una herramienta automatizada.

2. ¿Con qué frecuencia utiliza los servicios de atención al cliente de la entidad?

Tabla 13. Tabla de frecuencias – Encuesta 2 – Pregunta 2

Indicador	Con mucha frecuencia	Con frecuencia	Ocasionalmente	Raramente	Suma total
Frecuencia	3	12	15	10	40
Porcentaje	7,50%	30,00%	37,50%	25,00%	100,00%

¿Con qué frecuencia utiliza los servicios de atención al cliente de la entidad?

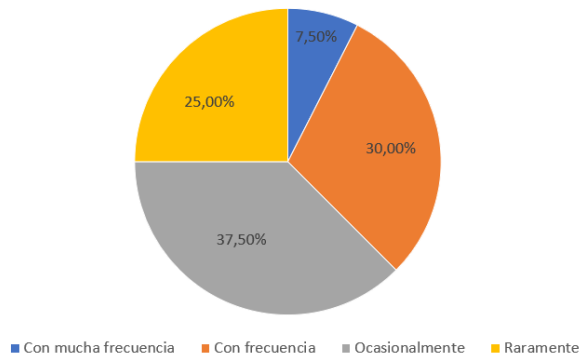


Figura 14. Resultados Encuesta 2 – Pregunta 2

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 14 indican que solo una minoría representada con un 7,5% acude a los servicios de atención al cliente con mucha frecuencia, seguido de un 25% que acude raramente, un 30% con frecuencia y un 37,5% que acude de manera ocasional.

3. ¿Qué nivel de satisfacción tiene con respecto al tiempo para generar su turno?

Tabla 14. Tabla de frecuencias – Encuesta 2 – Pregunta 3

Indicador	Muy satisfecho	Satisfecho	Neutral	Insatisfecho	Muy insatisfecho	Suma total
Frecuencia	2	11	9	16	2	40
Porcentaje	5,00%	27,50%	22,50%	40,00%	5,00%	100,00%

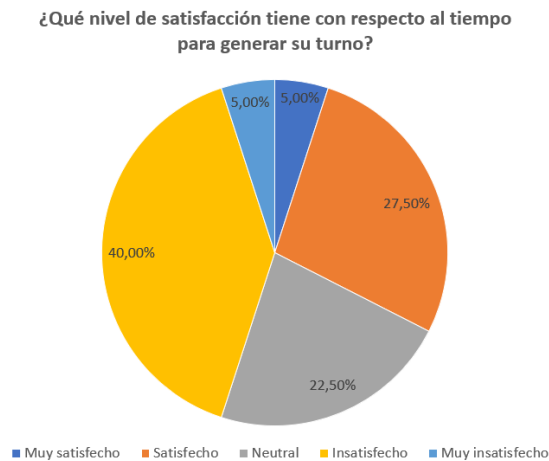


Figura 15. Resultados Encuesta 2 – Pregunta 3

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 15 indican que el 5% de los clientes está muy satisfecho con el tiempo que se tarda en generar un turno, mientras que otro 5% está muy insatisfecho, un 22,5% se mantiene en una opinión neutral, un 27,5% está satisfecho y la mayoría, un 40% se muestra insatisfecho con el tiempo de generación del turno, mostrando que la falta de un sistema provoca un cierto nivel de insatisfacción a los clientes con los tiempos del proceso.

4. ¿Qué nivel de satisfacción tiene con respecto al tiempo de espera para recibir atención?

Tabla 15. Tabla de frecuencias – Encuesta 2 – Pregunta 4

Indicador	Muy satisfecho	Satisfecho	Neutral	Insatisfecho	Muy insatisfecho	Suma total
Frecuencia	2	9	13	12	4	40
Porcentaje	5,00%	22,50%	32,50%	30,00%	10,00%	100,00%

¿Qué nivel de satisfacción tiene con respecto al tiempo de espera para recibir atención?

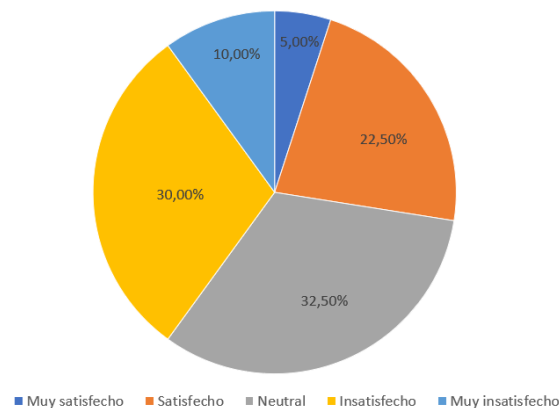


Figura 16. Resultados Encuesta 2 – Pregunta 4

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 16 indican que el 5% de los clientes se muestran muy satisfechos con el tiempo de espera para recibir atención, un 10% se mantiene muy insatisfecho, seguido de un 22,5% que se muestra satisfecho, un 30% que se encuentra insatisfecho y un 32,5% que mantiene una opinión neutral, evidenciando nuevamente un cierto nivel de insatisfacción con los tiempos debido a la falta de un sistema especializado.

5. ¿Cuánto tiempo espera normalmente hasta que se atiende su turno?

Tabla 16. Tabla de frecuencias – Encuesta 2 – Pregunta 5

Indicador	Entre 1 y 5 minutos	Entre 5 y 15 minutos	Más de 15 minutos	Suma total
Frecuencia	7	21	12	40
Porcentaje	17,50%	52,50%	30,00%	100,00%

¿Cuánto tiempo espera normalmente hasta que se atiende su turno?

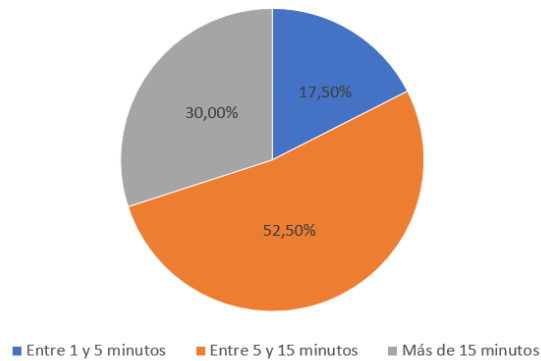


Figura 17. Resultados Encuesta 2 – Pregunta 5

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 17 indican que la minoría de los clientes representados con un 17,5% espera de 1 a 5 minutos para ser atendidos, seguido de un 30% que espera más de 15 minutos y un 52,5% que se mantiene en espera de 5 a 15 minutos.

6. ¿En qué medida cree que el proceso de generación de turnos es accesible y cómodo?

Tabla 17. Tabla de frecuencias – Encuesta 2 – Pregunta 6

Indicador	Muy cómodo	Cómodo	Neutral	Incómodo	Muy incómodo	Suma total
Frecuencia	1	10	14	14	1	40
Porcentaje	2,50%	25,00%	35,00%	35,00%	2,50%	100,00%

¿En qué medida cree que el proceso de generación de turnos es accesible y cómodo?

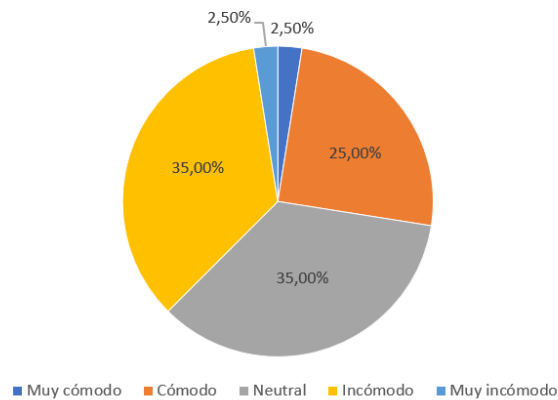


Figura 18. Resultados Encuesta 2 – Pregunta 6

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 18 indican que solo un 2,5% cree que el proceso es muy cómodo, otro 2,5% cree por el contrario que es muy incómodo, seguido de un 25% que opina que es cómodo, un 35% que se muestra neutral y otro 35% que opina que es incómodo el proceso de generación de turnos, evidenciando la dificultad del proceso para los clientes.

7. ¿Qué cantidad de información sobre la atención de su turno cree que se proporciona?

Tabla 18. Tabla de frecuencias – Encuesta 2 – Pregunta 7

Indicador	Mucha	Normal	Ninguna	Poca	Muy poca	Suma total
Frecuencia	0	1	2	11	26	40
Porcentaje	0,00%	2,50%	5,00%	27,50%	65,00%	100,00%

¿Qué cantidad de información sobre la atención de su turno cree que se proporciona?

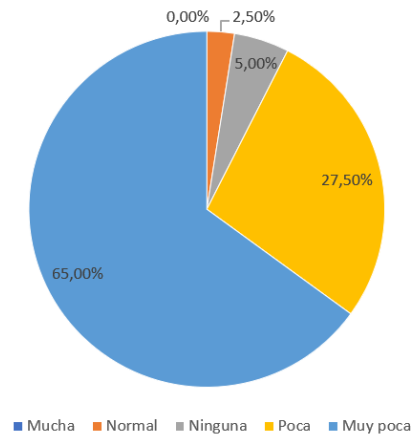


Figura 19. Resultados Encuesta 2 – Pregunta 7

Análisis e interpretación de datos

Los resultados obtenidos y representados en la Figura 19 indican que un 2,5% opina que la información que se brinda al cliente sobre la atención de su turno es normal, seguido de un 5% que opina que no se brinda ninguna información, un 27,5% cree que es poca y por último un 65% opina que la información brindada es muy poca, por lo que se puede concluir que la información relevante recopilada y que se muestra al cliente durante el proceso es prácticamente nula.

2.2.4 Procesamiento y análisis de datos

De acuerdo con la entrevista y las encuestas aplicadas tanto al personal como a los clientes de la EMAPA, se pudo concluir que:

- El proceso mediante el cual se maneja actualmente la información no es totalmente efectivo, habiendo tenido un sistema que ya no se utiliza debido a que está desactualizado y conllevando al manejo manual, dando como resultado una falta de unificación y automatización del proceso.
- La aplicación debe ser web debido a que la empresa maneja varias aplicaciones web para brindar sus servicios.
- Las tecnologías a utilizar deben ser:
 - Servicios WCF para el desarrollo de los servicios, debido a que la empresa maneja servicios WCF y ASP.NET y tiene un servidor IIS dedicado a hospedar estos servicios.
 - SQL Server como motor de base de datos, debido a que la empresa ya maneja varias bases de datos con SQL Server y ya tiene un servidor que se dedica a almacenar distintas bases.
- La aplicación debe tener los siguientes apartados:
 - Pantalla de generación de turnos.
 - Pantalla de información.
 - Pantalla de calificación.
 - Pantalla de atención de turnos para los trabajadores.
 - Pantalla de administración para los usuarios administradores.
- La aplicación debe permitir además la generación de reportes según varios parámetros en formatos PDF o Excel.

- El personal de la empresa encuestado considera en su mayoría que el desarrollo de una aplicación web ayudaría a solucionar diversos aspectos en los que tienen falencias los métodos actuales.
- Los clientes encuestados no se muestran totalmente satisfechos con los tiempos de generación y atención de los turnos, además de que indican que la información brindada acerca de la atención de su turno es en su mayoría muy poca o nula.

A través del análisis de la información recolectada se pudieron identificar los requisitos fundamentales para cubrir las necesidades de la empresa y llevar a cabo el desarrollo de la aplicación web centralizada para la gestión y atención de turnos.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

3.1.1 Diagrama del proceso con la gestión manual

a. Proceso de generación y atención de turnos

El proceso de generación y atención de turnos de manera manual es el siguiente:

1. El cliente:
 - Pregunta al personal de información.
2. El personal de información:
 - Atiende la petición del cliente.
 - Proporciona la información requerida al cliente.
3. El cliente:
 - Espera a ser atendido.
4. El personal de atención al cliente:
 - Llama a los clientes en orden de llegada.
5. El cliente:
 - Se dirige a la ventanilla en la que fue llamado.
6. El personal de atención:
 - Atiende al cliente.

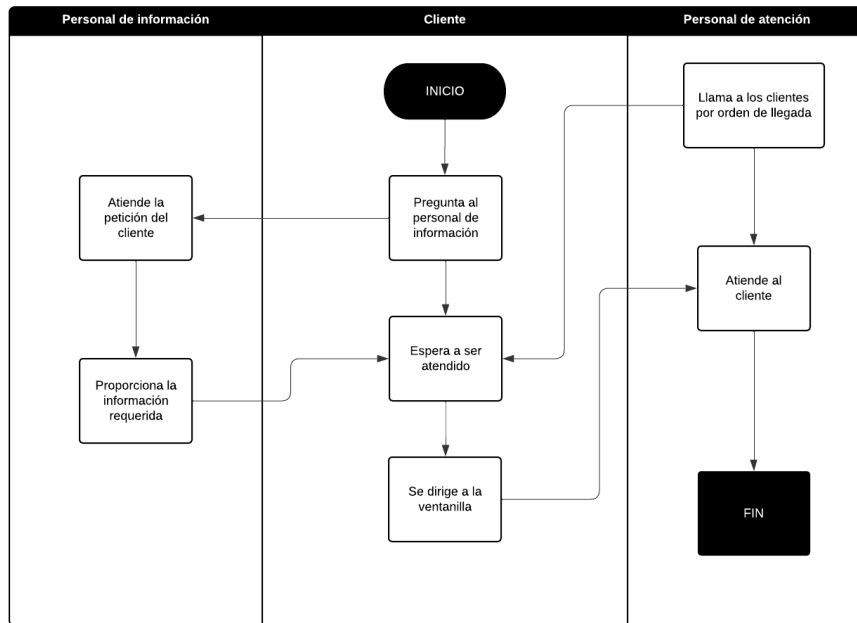


Figura 20. Diagrama del proceso manual de atención de turnos

b. Proceso de calificación de atención

La empresa no utiliza ningún tipo de sistema por lo que no tiene ningún método efectivo para calificar las atenciones, el único método de calificación que manejan es mediante llamadas aleatorias a clientes para realizarles preguntas a través de una encuesta de satisfacción. Este método limita el alcance y la capacidad de obtener una visión completa y precisa de la calidad de servicio.

3.1.2 Diagrama del proceso con la aplicación

a. Proceso de generación y atención de turnos

El proceso de generación y atención de turnos con la aplicación es el siguiente:

1. El cliente:
 - Selecciona el motivo del turno.
 - Confirma la generación del turno.
2. La aplicación:

- Imprime y guarda el turno generado.
3. El trabajador:
- Ingresa a la aplicación.
 - Selecciona la sucursal y la ventanilla.
 - Inicia sesión.
 - Selecciona el turno que va a atender.
4. La aplicación:
- Actualiza la información sobre las atenciones.
 - Muestra la información sobre las atenciones a los clientes.
5. El cliente:
- Visualiza la información sobre las atenciones.
 - Se acerca a la ventanilla en la que le van a atender.
6. El trabajador:
- Toma las observaciones y datos de la atención.
 - Finaliza la atención.
7. La aplicación:
- Guarda las observaciones de la atención.

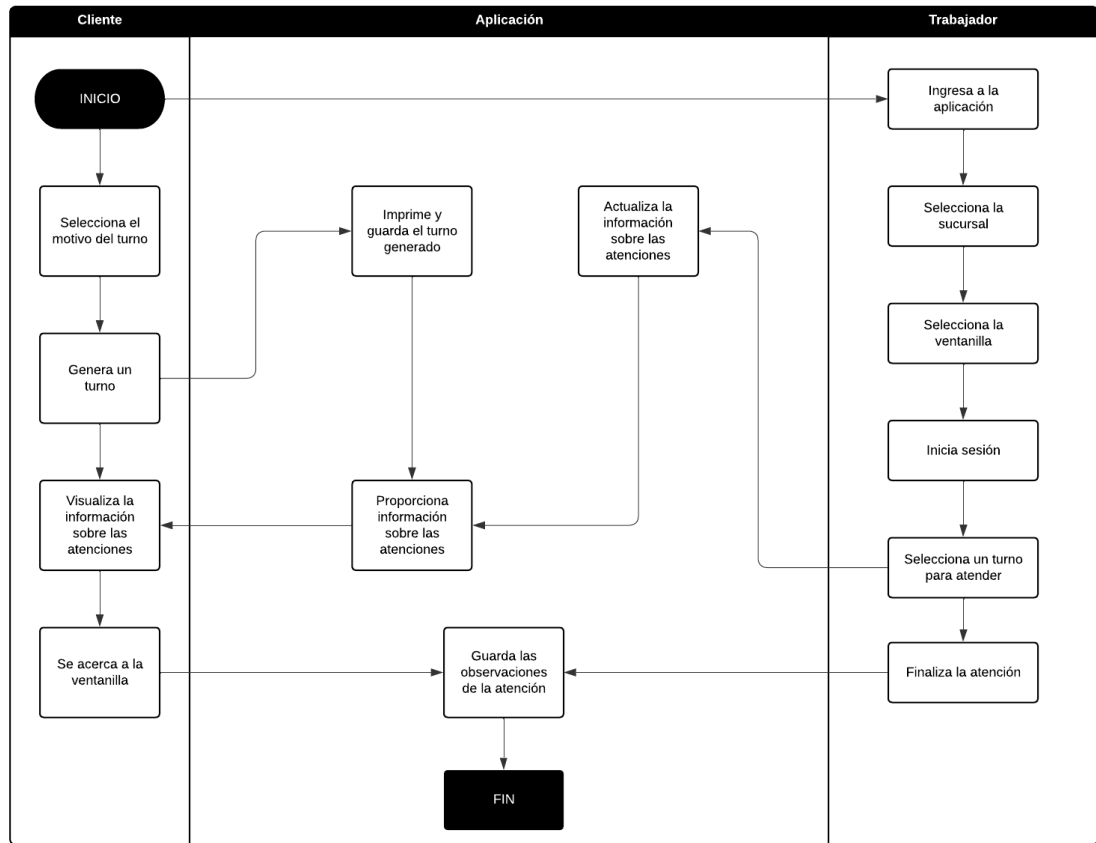


Figura 21. Diagrama del proceso de generación y atención de turnos con la aplicación

b. Proceso de calificación de atención

El proceso de calificación de la atención con la aplicación es el siguiente:

1. El trabajador:
 - Ingresa a la aplicación.
 - Selecciona la sucursal y la ventanilla.
 - Finaliza una atención.
2. La aplicación:
 - Crea una calificación y guarda una calificación vacía para la atención finalizada.

- Muestra el número de turno atendido.
- Muestra las preguntas de la encuesta.

3. El cliente:

- Visualiza su número de turno y las preguntas.
- Contesta las preguntas de la encuesta.
- Envía las respuestas.

4. La aplicación:

- Actualiza los valores de calificación de la atención.

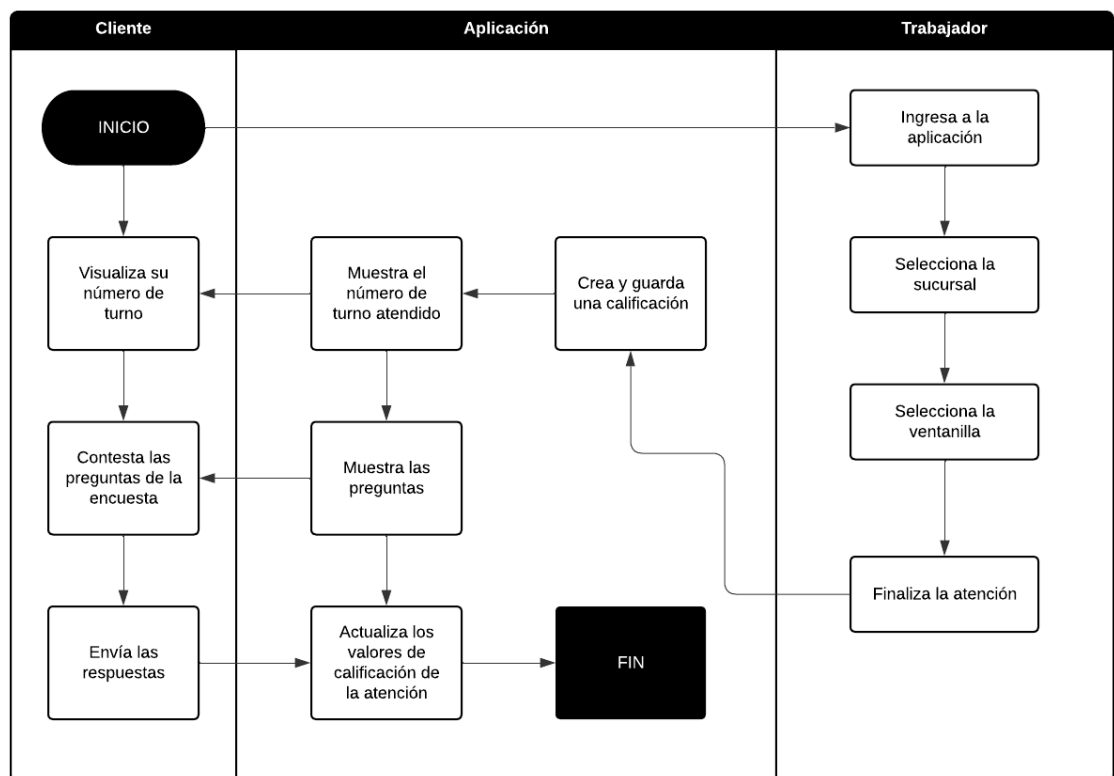


Figura 22. Diagrama del proceso de calificación de atención con la aplicación

3.1.3 Análisis y selección de la metodología de desarrollo

a. Análisis y selección de tipo de metodología

A continuación, se presenta una tabla comparativa destacando las características más importantes de los tipos de metodologías [28]:

Tabla 19. Tabla comparativa – Tipos de metodologías

Criterio	Metodologías clásicas	Metodologías ágiles
Enfoque	Secuencial	Iterativo e incremental
Planificación	Detallada desde el inicio	Flexible y adaptable
Requerimientos	Requerimientos bien definidos y estables	Requerimientos cambiantes y evolutivos
Comunicación	Formal y limitada	Constante y colaborativa
Retroalimentación	Limitada durante el desarrollo	Continua
Entregas	Al finalizar el proyecto	Periódicas
Adaptabilidad	Poco adaptativa a cambios durante el desarrollo	Altamente adaptativa a cambios durante el desarrollo
Clientes	Involucramiento limitado	Participación activa y continua
Riesgos	Poca capacidad de gestión de riesgos	Enfocado en la gestión de riesgos

En base a las características de la Tabla 19, se decide utilizar una metodología ágil debido a que los requerimientos del proyecto son cambiantes y actualizables en base a entregas periódicas y la participación continua del cliente, garantizando que el proyecto satisfaga las necesidades de la entidad.

b. Análisis y selección de metodología

A continuación, se presenta una tabla comparativa que destaca las características y principios significativos de 3 marcos de trabajo ágiles empleados en el proceso de desarrollo de software [24] [30]:

Tabla 20. Tabla comparativa – Metodologías de desarrollo ágiles

Criterio	RAD	Scrum	XP
Enfoque	Entrega rápida de prototipos funcionales	Colaboración y entrega incremental	Calidad de código y adaptabilidad
Ciclos de desarrollo	Ciclos de desarrollo cortos y rápidos	Ciclos de desarrollo de 2 a 4 semanas (Sprints)	Ciclos de desarrollo cortos enfocados en la calidad
Participación del cliente	Participación activa y continua del cliente	Involucra al cliente en las revisiones finales de cada sprint	Colaboración continua con el cliente
Comunicación	Fuerte enfoque en la colaboración y comunicación con el cliente	Colaboración y comunicación constante con el cliente	Colaboración y comunicación cercana con el cliente
Entregas	Entregas funcionales rápidas y periódicas	Entregas regulares al final de cada Sprint	Entregas continuas e incrementales
Flexibilidad	Altamente adaptable a cambios	Adaptable a cambios con un enfoque estructurado	Adaptable a cambios
Tipo de proyectos	Adecuado para proyectos pequeños o medianos con requerimientos cambiantes	Adecuado para proyectos medianos o grandes con entregas regulares	Adecuado para proyectos pequeños o medianos enfocados en la calidad
Ventajas	Costo y tiempo	Colaboración y documentación	Calidad y comunicación

Finalizado el análisis de las características de las metodologías expuestas en la Tabla 20, se decidió utilizar la metodología RAD, debido a que se destaca por la capacidad de realizar entregas rápidas de prototipos funcionales, lo que encaja con el modo de trabajo de este proyecto, al tener que trabajar con 2 clientes y ser un proyecto en el que los requerimientos se van a actualizar continuamente a través de la retroalimentación periódica.

Tanto Scrum como XP tienen sus propias ventajas, pero teniendo en cuenta varios aspectos como el tipo de proyecto, en este caso, un proyecto de mediana escala con requerimientos cambiantes, con un periodo de desarrollo no tan extenso y teniendo que adaptarse de manera continua a los cambios resultantes de las revisiones, se considera a RAD como la mejor opción para cumplir con los plazos y las necesidades del cliente.

c. Análisis y selección de framework

A continuación, se presenta una tabla comparativa que destaca las características y puntos importantes de tres frameworks para el desarrollo de aplicaciones web [15] [16] [17]:

Tabla 21. Tabla comparativa – Herramientas de desarrollo

Criterios	Laravel	Angular	React JS
Lenguaje de programación	PHP	TypeScript	JavaScript
Rendimiento	Buen rendimiento	Buen rendimiento	Excelente rendimiento
UI y experiencia de usuario	No se enfoca en la creación de interfaces de usuario	Permite crear interfaces de usuario personalizadas e interactivas	Destaca por su alta capacidad para crear interfaces de usuario interactivas y dinámicas
Ecosistema	Maduro y en crecimiento Buena cantidad de recursos disponibles	Amplio y complejo Amplia gama de bibliotecas	Amplio y maduro Gran cantidad de bibliotecas

Criterios	Laravel	Angular	React JS
Curva de aprendizaje	Curva de aprendizaje moderada	Curva de aprendizaje moderada	Curva de aprendizaje accesible debido a su enfoque en JavaScript
Flexibilidad	Flexibilidad en el desarrollo web	Estructurado, con tendencia a seguir un conjunto de reglas y convenciones predefinidas	Altamente flexible, se integra fácilmente con otras bibliotecas y frameworks
Rendimiento en Aplicaciones de Tiempo Real	No está diseñado específicamente para aplicaciones en tiempo real	Se adapta bien a aplicaciones en tiempo real	Excelente rendimiento a través de su manipulación eficiente del DOM

Al finalizar el análisis en base a las características de la Tabla 21, se concluyó que React JS es la mejor opción debido a que maneja un lenguaje de programación ampliamente utilizado, ofrece un rendimiento excelente a través de una manipulación eficiente del DOM (Document Object Model) para crear interfaces interactivas y dinámicas, lo que se ajusta a las necesidades de este proyecto para mostrar la información actualizada de manera rápida, siendo una de las mejores opciones para el rendimiento de aplicaciones en tiempo real, además de tener una amplia gama de bibliotecas compatibles y una curva de aprendizaje accesible.

3.2 Desarrollo de la propuesta

3.2.1 Fase I: Recopilación de requerimientos

En la fase de recopilación de requisitos en conjunto con los representantes se levantaron los requerimientos necesarios para iniciar con el proyecto.

a. Definición de roles

Para realizar el proyecto se contó con la colaboración de tres personas, las cuales se detallan en la siguiente tabla.

Tabla 22. Definición de roles y funciones

Responsable	Rol	Función
Javier Jiménez	Desarrollador	Encargado de programar el sistema de acuerdo con los requerimientos establecidos.
Ing. Fernando Ibarra	Verificador	Encargado de monitorear y dar seguimiento al desarrollo de la aplicación.
Ing. Ricardo Rivera	Cliente/Usuario	Representante del departamento de TICS de EMAPA, en conjunto se realizará la retroalimentación y se entregará el producto final.
Responsable	Rol	Función
Ing. Eduardo Gordillo	Cliente/Usuario	Representante del departamento de la Unidad de Atención al Cliente de EMAPA, en conjunto se realizará la retroalimentación y se entregará el producto final.

b. Requerimientos técnicos

Los requerimientos de herramientas que fueron determinados por la institución, en base a la infraestructura y la experiencia tecnológica existente.

Tabla 23. Requerimientos técnicos

Nombre de la tecnología	Versión	Descripción
Sql Server	15	Gestor de base de datos
.NET	5	Plataforma de desarrollo de servicios

c. Listado de requerimientos

Los requerimientos de la aplicación web para la gestión y atención de turnos se establecieron en etapas, en base a las entregas periódicas de los avances acorde con la metodología RAD y en conjunto con la información recopilada con las encuestas al personal y los clientes de la institución.

- **Etapas inicial:** Se realiza una reunión con el representante de la empresa para establecer los requerimientos iniciales de la aplicación.

- **Etapa de entregas periódicas:** Se realizan reuniones periódicas en las que se revisa el prototipo presentado y en las que se puede actualizar los requerimientos de la aplicación.
- **Etapa final:** Se realiza una reunión final en la que se revisa que todos los requerimientos hayan sido cumplidos y se aprueba la finalización de la fase de desarrollo de la aplicación.

Durante la recopilación de requerimientos se interactuó con:

La siguiente tabla consta de los siguientes campos:

- **ID:** identificador del requerimiento, consta de la siguiente estructura: (R00 - Número).
- **Descripción:** describe el requerimiento.
- **Prioridad:** Se representa con la siguiente escala: Alta, Media, Baja.
- **Riesgo:** Se representa con la siguiente escala: Alta, Media, Baja.

Requerimientos iniciales

Se realizó una reunión con los representantes de la empresa en la que se establecieron los requerimientos iniciales e indispensables de la aplicación representados en la Tabla 24.

Tabla 24. Requerimientos iniciales

ID	Requerimiento	Descripción	Prioridad	Riesgo
R001	Listar sucursales	La aplicación debe listar las sucursales que se encuentran disponibles	Alta	Alto
R002	Selección de sucursal	La aplicación debe permitir seleccionar la sucursal en la que va a funcionar de manera obligatoria	Alta	Alto

ID	Requerimiento	Descripción	Prioridad	Riesgo
R003	Generación de turnos	La aplicación debe tener una pantalla para la generación de turnos del usuario de manera rápida, únicamente seleccionando el motivo de la consulta, con lo que se le generará un número de turno impreso en un ticket	Alta	Alto
R004	Selección de ventanilla	Antes de iniciar sesión se deben listar las ventanillas que están disponibles en la sucursal seleccionada para que el trabajador pueda seleccionar la ventanilla en la que va a trabajar	Alta	Medio
R005	Autenticación del usuario	El ingreso a la aplicación web debe ser a través del nombre de usuario y la clave asignada Se debe manejar roles de usuario (Administrador, trabajador)	Alta	Alto
R006	Listar turnos en espera de atención	Como usuario con rol de trabajador debo poder visualizar todos los turnos generados en ese día que estén a la espera de atención para poder brindarles atención	Alta	Alto
R007	Atender turno	Como usuario con rol trabajador, debo poder seleccionar un turno de la lista, añadir una observación y guardar la atención	Alta	Alto
R008	Visualización de atenciones	Como usuario con rol administrador debo poder visualizar el listado de atenciones con la información del usuario que realizó la atención, las fechas y horas de generación de turno, inicio y finalización de la atención, la sucursal en la que se llevó a cabo y la observación de cada atención	Alta	Alto
R009	Gestión de datos	La aplicación debe permitir crear, actualizar, eliminar y visualizar los datos almacenados	Alta	Medio

ID	Requerimiento	Descripción	Prioridad	Riesgo
R010	Pantalla de información	La aplicación debe contar con una pantalla en la que se muestre la información de los turnos de la sucursal en la que esté funcionando	Alta	Medio
R011	Pantalla de calificación	La aplicación debe contar con una pantalla en la que se pueda calificar una atención una vez que ha finalizado	Media	Medio
R012	Filtrado de datos	Como usuario con rol administrador debo poder realizar un filtrado de los datos de cada tabla, en base a distintos parámetros	Media	Medio
R013	Generación de reportes	Como usuario con rol administrador debo poder exportar los datos de una tabla (filtrada o no) a un archivo Excel	Media	Medio
R014	Cerrar sesión	La aplicación debe permitir al usuario cerrar la sesión que tiene una activa	Alta	Medio

d. Plan de entregas

En base a los requerimientos definidos, se elaboró un plan de entrega para organizar el desarrollo del proyecto:

Tabla 25. Plan de entrega

Iteración	Código	Requerimientos	Tiempo	
			Horas	Días
Iteración 1	R001	Listado de sucursales	1	1
	R002	Selección de sucursal	1	1
	R003	Generación de turnos	5	2
Iteración 2	R004	Selección de ventanilla	2	1
	R005	Autenticación del usuario	4	1
	R006	Listado de turnos en espera de atención	2	1

Iteración	Código	Requerimientos	Tiempo	
			Horas	Días
Iteración 3	R007	Atención de turno	5	2
	R008	Visualización de atenciones	2	1
	R009	Gestión de datos	7	3
Iteración 4	R010	Pantalla de información	4	2
	R011	Pantalla de calificación	5	2
	R012	Filtrado de datos	6	1
Iteración 5	R013	Generación de reportes	3	2
	R014	Cerrar sesión	2	1

e. Arquitectura de la aplicación

La arquitectura de la aplicación es una multicapa, para facilitar la gestión, mantenimiento y escalabilidad de la aplicación y que consta de 6 capas, las cuales se detallan a continuación:

Entidades: Capa en la que se definen las estructuras de los objetos que representan las entidades de la base de datos que participan en la aplicación.

Base de datos: Capa que representa y que se encarga de la creación, lectura, actualización y eliminación de toda la información almacenada dentro de la base de datos.

Datos: Capa que se ocupa de la manipulación de los datos interactuando de manera directa con la capa de base de datos y en la que se definen operaciones específicas (acceso a datos, consultas, inserciones, actualizaciones y eliminaciones).

Lógica de negocio: Capa en la que se definen operaciones y reglas que no están directamente relacionadas con la manipulación de los datos, encargándose de gestionar las operaciones entre las capas de datos y entidades.

Servicios: Capa en la que se definen los servicios que representan la funcionalidad de la aplicación, brindando un mecanismo de comunicación con la capa de datos.

Cliente: Capa que contiene a la aplicación web que interactúa directamente con el servidor enviando solicitudes y recibiendo respuestas.

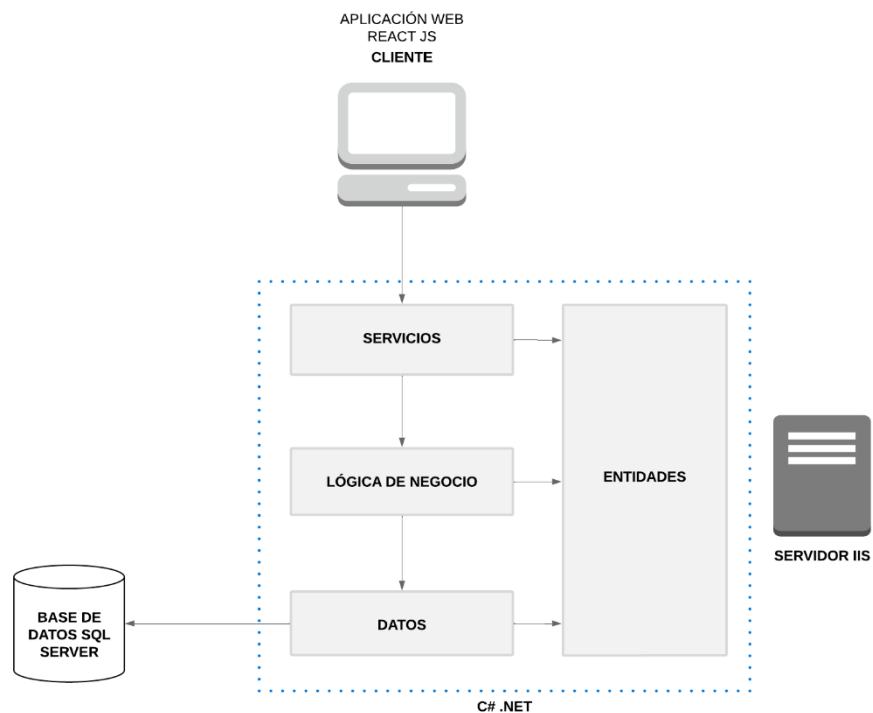


Figura 23. Arquitectura de la aplicación

f. Diseño de la base de datos

Para el almacenamiento y lectura de datos se realizó un diseño de base de datos relacional con la finalidad de obtener una estructura sólida y mantener la integridad en los datos, debido a la importancia de que la información que se maneje dentro de la aplicación para la gestión y atención de turnos sea lo más precisa y consistente posible.

La base de datos fue implementada en el gestor de bases de datos SQL Server y actualizada en base a los diferentes cambios planteados durante las entregas periódicas para que el diseño se ajuste completamente a los requerimientos de la empresa. Al

finalizar todo el proceso el diseño final de la base de datos se representó a través de un diagrama entidad–relación como se observa en la Figura 24.

Se obtuvieron un total de 10 tablas, que almacenan toda la información necesaria para la generación y atención de los turnos. Los datos almacenados en cada una de estas tablas están brevemente descritos dentro de un diccionario de datos que se encuentra en el Anexo E, para una mejor comprensión de la base de datos.

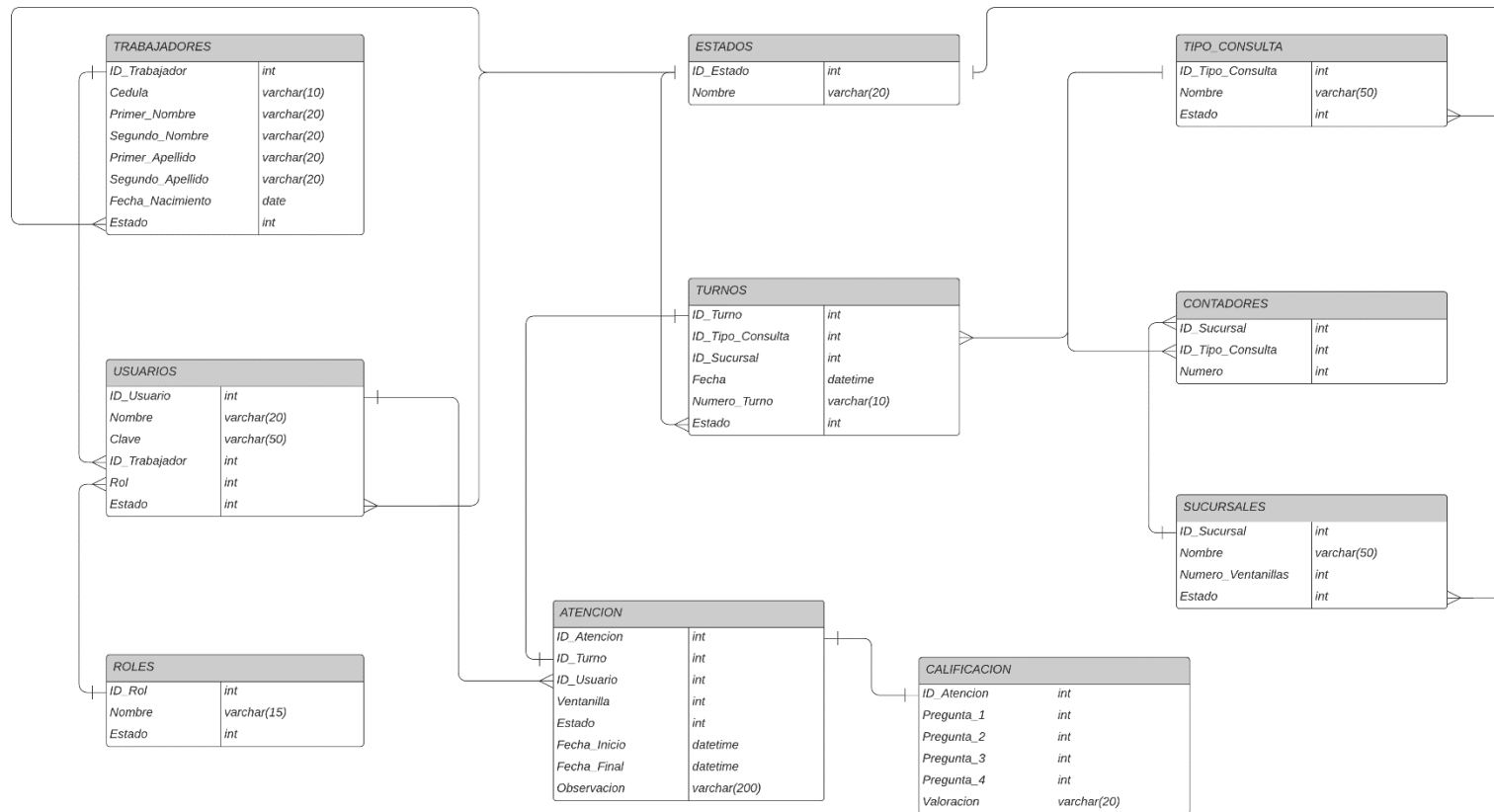


Figura 24. Diagrama Entidad – Relación de la Base de Datos

3.2.2 Fase II: Prototipado y retroalimentación del cliente

En esta fase de la metodología se elaboró el prototipo de la aplicación y se presentó al usuario con el objetivo de obtener una retroalimentación con la que realizar los cambios necesarios para obtener la aprobación y comenzar con el desarrollo de la aplicación.

Para la elaboración del prototipo se utilizó Figma, una herramienta de diseño de interfaz y experiencia de usuario basada en la nube que ofrece una interfaz intuitiva para la elaboración de interfaces interactivas con diseños versátiles [31].

- **Pantalla principal**



Figura 25. Prototipo de la pantalla principal

En la Figura 25 se muestra una representación de la pantalla principal de la aplicación, pantalla que se muestra al ingresar, en la que se debe seleccionar la sucursal donde va a funcionar para poder acceder a cualquiera de las otras pantallas.

- **Inicio de sesión**

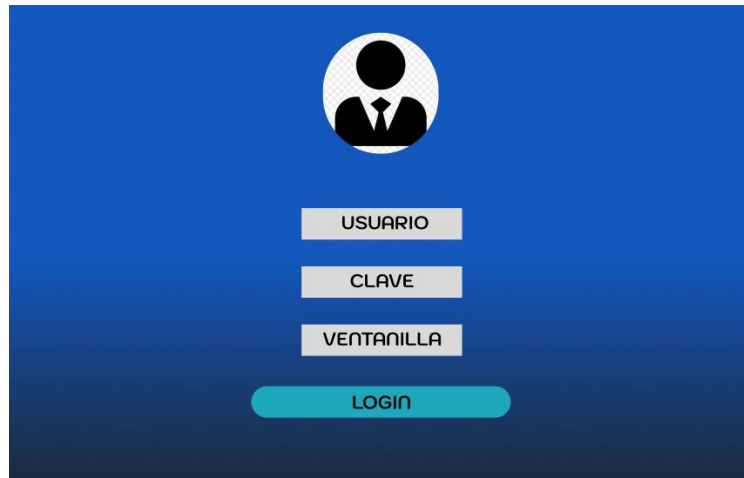


Figura 26. Prototipo de la pantalla de inicio de sesión

En la Figura 26 se muestra el prototipo de la pantalla para iniciar sesión en la que el usuario debe ingresar solo su nombre de usuario y clave en caso de ser administrador o también seleccionar el número de ventanilla en el que va a trabajar en caso de ser trabajador.

- **Pantalla de calificación**

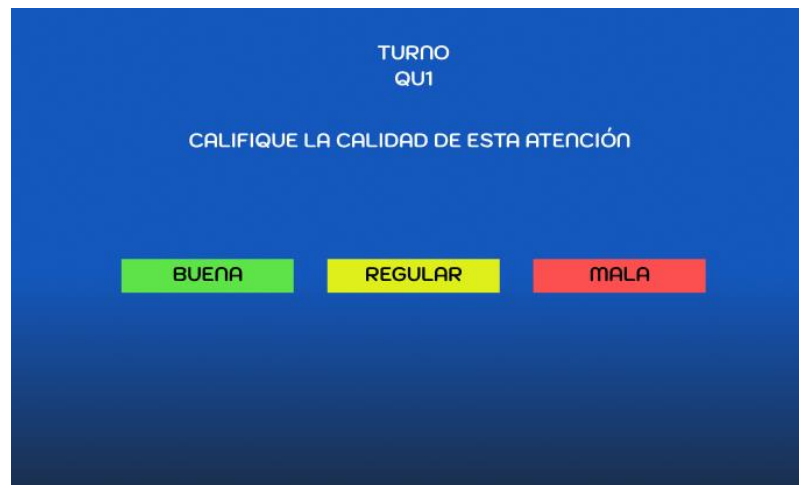


Figura 27. Prototipo de la pantalla de calificación

En la Figura 27 se muestra el prototipo de la pantalla de calificación en la que el cliente puede realizar una valoración sobre la atención recibida.

- **Pantalla de información**



Figura 28. Prototipo de la pantalla de información

En la Figura 28 se observa el prototipo de la pantalla de información en la que el cliente puede visualizar información sobre los turnos que están siendo atendidos y en qué ventanillas, además los turnos que están en espera y un video en el lado derecho de la pantalla.

- **Pantalla de clientes**



Figura 29. Prototipo de la pantalla de clientes

En la Figura 29 se muestra el prototipo de la pantalla en la que los clientes pueden visualizar los diferentes motivos para sacar un turno.

- **Generación de turno**

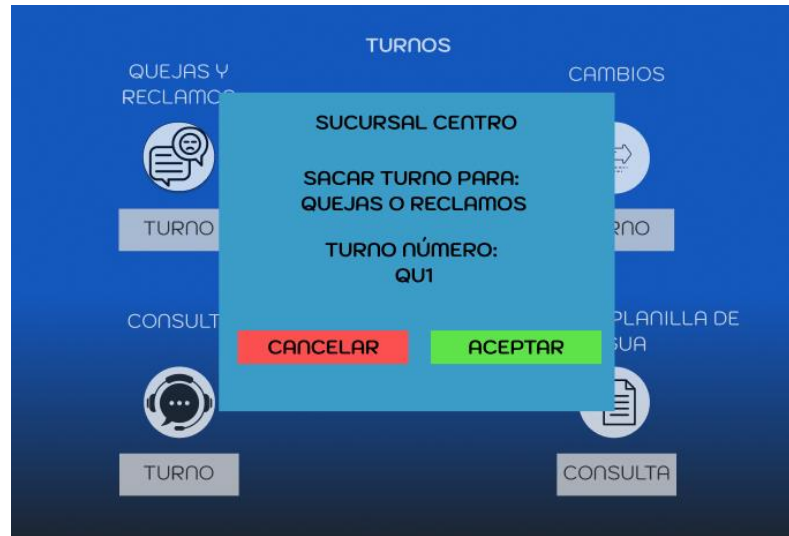


Figura 30. Prototipo de generación de turno

En la Figura 30 se puede observar una ventana que aparece al seleccionar un motivo para sacar el turno como se muestra en la Figura 29, en la que el cliente observa la sucursal, el motivo y el número de turno que va a generar si selecciona “Aceptar”.

En caso de seleccionar la opción “Cancelar”, la ventana se cierra, el turno no se genera y se vuelve a la pantalla de clientes representada en la Figura 29.

- **Pantalla de trabajador**

ATENCIÓN DE TURNOS			LOGOUT
TURNO	MOTIVO	HORA	
QU1	Queja	10:30	ATENDER
---	---	---	
---	---	---	
---	---	---	
---	---	---	
---	---	---	
---	---	---	
---	---	---	
---	---	---	
---	---	---	

Figura 31. Prototipo de la pantalla del trabajador

En la Figura 31 se muestra el prototipo de la pantalla que visualiza el trabajador en la que se listan todos los turnos que están esperando atención y en la que el trabajador puede seleccionar un turno para atender.

- **Atención de turno**

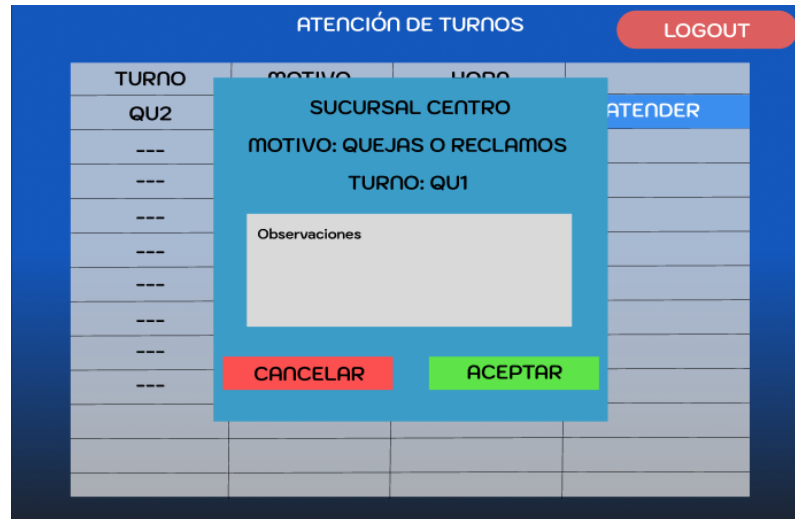


Figura 32. Prototipo de atención de turno

En la Figura 32 se muestra una ventana que aparece al seleccionar la opción “Atender” en la pantalla del trabajador representada en la Figura 31, donde se muestra la sucursal, el motivo y el número del turno seleccionado para atender, además de una caja para ingresar observaciones de la atención.

Al seleccionar la opción “Aceptar”, la atención finaliza y el turno desaparece de la lista de turnos en espera. Al contrario, si se selecciona la opción “Cancelar”, la ventana se cierra y el turno vuelve a aparecer en la lista.

- **Pantalla de administrador**



Figura 33. Prototipo de la pantalla del administrador

En la Figura 33 se observa el prototipo de la pantalla que se muestra al iniciar sesión con un usuario de administrador, en la que aparecen los distintos datos que se pueden gestionar.

- **Gestión de datos**



Figura 34. Prototipo de la pantalla de gestión de datos

En la Figura 34 se visualiza una representación de la pantalla para gestionar los datos, a la que se ingresa al seleccionar una opción en la pantalla del administrador representada en la Figura 33.

En esta pantalla se listan los datos con unos botones para editar o desactivar un dato en específico, uno para añadir un dato y además unos para exportar los datos en un archivo Excel o PDF.

- **Requerimientos finales**

Al finalizar:

- **Etapa de entregas periódicas:** Una vez cumplidos los requerimientos iniciales y teniendo una base de la aplicación se realizaron reuniones periódicas en conjunto con los representantes de la empresa para discutir los requerimientos ya establecidos y actualizar aquellos requerimientos necesarios para cumplir con las necesidades tanto en la parte del cliente como en la parte de los trabajadores y la institución.
- **Etapa final:** Se realizó una reunión una vez se cumplió con los últimos requerimientos actualizados en la reunión anterior, donde al discutir sobre los mismos se llegó a la conclusión de que los requerimientos ya establecidos cumplieron con las necesidades totales de los clientes y de la institución, estando de acuerdo los representantes de la empresa.

En conjunto con el prototipo presentado los requerimientos finales actualizados de la aplicación fueron los siguientes:

Tabla 26. Requerimientos finales

ID	Requerimiento	Descripción	Prioridad	Riesgo
R001	Listar sucursales	La aplicación debe listar las sucursales que se encuentran disponibles	Alta	Alto
R002	Selección sucursal	de La aplicación debe permitir seleccionar la sucursal en la que va a funcionar de manera obligatoria	Alta	Alto
R003	Generación turnos	de La aplicación debe tener una pantalla para la generación de turnos del usuario de manera rápida, únicamente seleccionando el motivo de la consulta, con lo que se le generará un número de turno impreso en un ticket	Alta	Alto

ID	Requerimiento	Descripción	Prioridad	Riesgo
R004	Selección de ventanilla	Antes de iniciar sesión se deben listar las ventanillas que están disponibles en la sucursal seleccionada para que el trabajador pueda seleccionar la ventanilla en la que va a trabajar	Alta	Medio
R005	Autenticación del usuario	<p>El ingreso a la aplicación web debe ser a través del nombre de usuario y la clave asignada</p> <p>Se debe manejar roles de usuario (Administrador, trabajador)</p> <p>En caso de ingresar las credenciales de administrador, la selección de ventanilla no debe ser obligatoria</p> <p>En caso de ingresar las credenciales de trabajador, la selección de ventanilla debe ser obligatoria</p>	Alta	Alto
R006	Listar turnos en espera de atención	<p>Como usuario con rol de trabajador debo poder visualizar todos los turnos generados en ese día que estén a la espera de atención para poder brindarles atención</p> <p>La información que debo visualizar como trabajador debe ser:</p> <ul style="list-style-type: none"> • Sucursal • Motivo de consulta • Numero de turno • Estado del turno • Hora a la que se sacó el turno 	Alta	Alto
R007	Atender turno	Como usuario con rol trabajador, debo poder seleccionar un turno de la lista, añadir una observación y guardar la atención	Alta	Alto

Como trabajador debo poder cancelar la atención en cualquier momento, de modo que, al cancelar la atención, el turno vuelva a aparecer en la lista que está en espera de atención

ID	Requerimiento	Descripción	Prioridad	Riesgo
R008	Visualización de atenciones	<p>Como usuario con rol administrador debo poder visualizar el listado de atenciones de todas las sucursales</p> <p>La información que se muestra de cada atención debe ser:</p> <ul style="list-style-type: none"> • Número de turno atendido • Nombre de usuario que atendió el turno • Nombre completo del trabajador que atendió el turno • Fecha en la que se llevó a cabo todo el proceso (Generación de turno y atención) • Hora en la que se generó el turno • Hora en la que inició la atención • Hora en la que finalizó la atención • Sucursal • Observación de la atención • Calificación 	Alta	Alto
R009	Gestión de datos	La aplicación debe permitir crear, actualizar, eliminar y visualizar los datos almacenados	Alta	Medio
R010	Pantalla de información	<p>La aplicación debe contar con una pantalla en la que se muestre la información de los turnos de la sucursal en la que esté funcionando</p> <p>La información mostrada en la pantalla debe ser:</p> <ul style="list-style-type: none"> • Turnos o turno que se está atendiendo 	Alta	Medio

- El número de ventanilla en la que se está atendiendo cada turno
- Lista de turnos que están en espera en orden según la hora a la que fueron creados

En una parte de la pantalla se debe mostrar un vídeo que se reproducirá en bucle

ID	Requerimiento	Descripción	Prioridad	Riesgo
R011	Pantalla de calificación	<p>La aplicación debe contar con una pantalla en la que se pueda calificar una atención una vez que ha finalizado</p> <p>El modo de calificación debe ser mediante una encuesta de satisfacción y al final una valoración de la atención en general del modo:</p> <ul style="list-style-type: none"> • Buena • Regular • Mala <p>Para el ingreso a la pantalla de calificación, se debe seleccionar la ventanilla de la cual se van a calificar las atenciones</p>	Media	Medio
R012	Filtrado de datos	Como usuario con rol administrador debo poder realizar un filtrado de los datos de cada tabla, en base a distintos parámetros	Media	Medio
R013	Generación de reportes	Como usuario con rol administrador debo poder exportar los datos de una tabla (filtrada o no) a un archivo Excel o PDF	Media	Medio
R014	Cerrar sesión	La aplicación debe permitir al usuario cerrar la sesión que tiene una activa	Alta	Medio

3.2.3 Fase III: Construcción

a. Herramientas de desarrollo

Para el desarrollo del proyecto se utilizaron:

- Visual Studio Code: Para el desarrollo de la interfaz del usuario de la aplicación web (Frontend)
- Visual Studio: Para el desarrollo de la lógica y los servicios de la aplicación web (Backend)

b. Creación del proyecto

- Para la creación del proyecto en Visual Studio Code se utilizó el comando:

```
PS E:\Projects> npx create-react-app sistema_turnos
```

Figura 35. Comando de creación de proyecto de React en Visual Studio Code

- Para la creación del proyecto en Visual Studio se utilizó la interfaz:

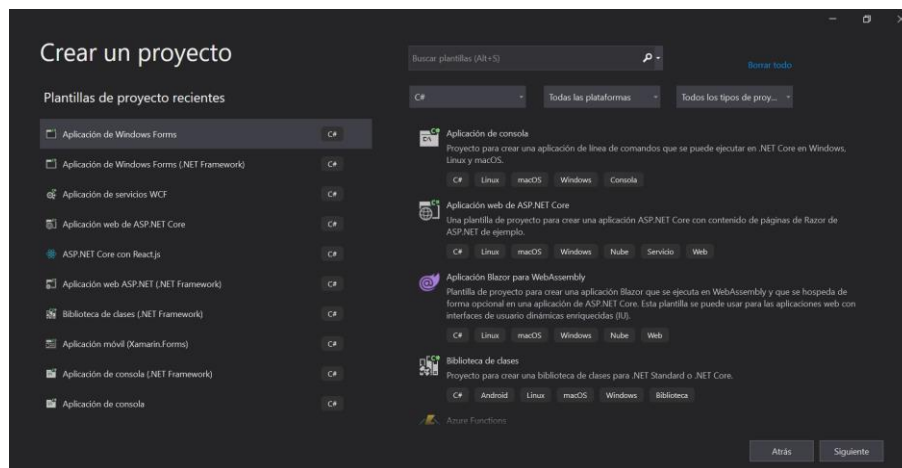


Figura 36. Interfaz de creación de proyecto en Visual Studio

c. Librerías utilizadas

- **React Router**

Es una biblioteca de React para facilitar el enrutamiento y navegación dentro de aplicaciones web [32]. Para la instalación de la biblioteca se utilizó el comando:

```
PS E:\Projects\sistema_turnos> npm install react-router
```

Figura 37. Comando de instalación de React Router

- **Axios**

Es una biblioteca de JavaScript para manejar solicitudes HTTP desde el navegador y realizar operaciones GET, POST, PUT y DELETE hacia el servidor [33]. Para la instalación de la biblioteca se utilizó el comando:

```
PS E:\Projects\sistema_turnos> npm install axios
```

Figura 38. Comando de instalación de Axios

- **React Datepicker**

Es una biblioteca de componentes de React para facilitar la selección y manejo de fechas dentro de aplicaciones web [34]. Para la instalación de la biblioteca se utilizó el comando:

```
PS E:\Projects\sistema_turnos> npm install react-datepicker
```

Figura 39. Comando de instalación de React Datepicker

- **jsPDF**

Es una biblioteca de JavaScript para generar documentos PDF a partir de la manipulación de elementos HTML de aplicaciones web [35]. Para la instalación de la biblioteca se utilizó el comando:

```
PS E:\Projects\sistema_turnos> npm install jsPDF
```

Figura 40. Comando de instalación de jsPDF

- **Xlsx**

Es una biblioteca de JavaScript para leer y generar documentos en formato Excel (.xlsx) a partir de datos de aplicaciones web [36]. Para la instalación de la biblioteca se utilizó el comando:

```
PS E:\Projects\sistema_turnos> npm install xlsx
```

Figura 41. Comando de instalación de Xlsx

- **React Paginate**

Es una biblioteca de React para la creación de paginaciones y la división de grandes cantidades de datos de aplicaciones web [37]. Para la instalación de la biblioteca se utilizó el comando:

```
PS E:\Projects\sistema_turnos> npm install react-paginate
```

Figura 42. Comando de instalación de React Paginate

d. Aplicación

- **Página principal**



Figura 43. Página principal de la aplicación

En la Figura 43 se muestra la página principal de la aplicación a partir de la que se accede a toda la funcionalidad. En la página principal se cargan las sucursales de la base de datos para que se seleccione, si no se selecciona una sucursal no se puede acceder a ninguna otra ventana.

```

public static List<Sucursal> DevolverListaSucursales()
{
    try
    {
        List<Sucursal> lista = new List<Sucursal>();
        SqlConnection conexion = new SqlConnection(Properties.Settings.Default.ConexionDB);
        conexion.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = @"SELECT [ID_Sucursal], [Nombre], [Numero_Ventanillas], [Estado]
FROM dbo.[SUCURSALES]";

        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                Sucursal sucursal = new Sucursal();
                sucursal.ID_Sucursal = Convert.ToInt32(dr["ID_Sucursal"].ToString());
                sucursal.Nombre = dr["Nombre"].ToString();
                sucursal.Estado = Convert.ToInt32(dr["Estado"].ToString());
                sucursal.Numero_Ventanillas = Convert.ToInt32(dr["Numero_Ventanillas"].ToString());

                lista.Add(sucursal);
            }
        }

        conexion.Close();

        return lista;
    }
    catch (Exception ex)
    {
        throw;
    }
}

```

Figura 44. Código C# para listar las sucursales

```

useEffect(() => {
    axios.get(apiUrl)
        .then(response => {
            setSucursales(response.data);
        }).catch(error => {
            console.error("Error fetching data:", error);
        });
}, []);

```

Figura 45. Código JavaScript para el consumo de la API de listar sucursales

En la Figura 45 se muestra el código con el que se realiza una petición GET al servidor para ejecutar el código que se observa en la Figura 44, que crea abre una conexión a la base de datos para obtener la lista de sucursales.

- **Página para sacar turnos**



Figura 46. Página para sacar turnos

En la Figura 46 se muestra la página en la que se listan los tipos de consulta de la base de datos y donde los clientes pueden sacar un turno dependiendo de su selección.

```

public static List<TipoConsulta> DevolverListaTiposConsulta()
{
    try
    {
        List<TipoConsulta> lista = new List<TipoConsulta>();
        SqlConnection conexion = new SqlConnection(Properties.Settings.Default.ConexionDB);
        conexion.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = @"SELECT [ID_Tipo_Consulta], [Nombre], [Estado]
                            FROM dbo.[TIPOS_CONSULTA]";

        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                TipoConsulta tipo = new TipoConsulta();
                tipo.ID_Tipo_Consulta = Convert.ToInt32(dr["ID_Tipo_Consulta"].ToString());
                tipo.Nombre = dr["Nombre"].ToString();
                tipo.Estado = Convert.ToInt32(dr["Estado"].ToString());

                lista.Add(tipo);
            }
        }

        conexion.Close();

        return lista;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

Figura 47. Código C# para listar tipos de consulta

```

useEffect(() => {
    axios
        .get(apiUrlConsultas)
        .then((response) => {
            setTipos(response.data);
        })
        .catch((error) => {
            console.error("Error fetching data:", error);
        });
}, []);

```

Figura 48. Código JavaScript para el consumo de la API de listar tipos de consulta

En la Figura 48 se muestra el código con el que se realiza una petición GET al servidor para ejecutar el código que se muestra en la Figura 47, que abre una conexión a la base de datos para obtener los tipos de consulta.



Figura 49. Ventana para confirmar el turno



Figura 50. Ventana de confirmación de generación de turno

En la Figura 49 se observa la ventana que se abre al seleccionar una opción, en la que se muestra la información sobre el turno que se va a generar. La ventana se cierra en caso de seleccionar cancelar y en caso de confirmar el turno, se imprime y se muestra la ventana de la Figura 50.


```

public static Turno NuevoTurno(Turno turno)
{
    try
    {
        SqlConnection conexion = new SqlConnection(Properties.Settings.Default.ConexionDB);
        conexion.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = @"INSERT INTO [dbo].[TURNOS]
        ([ID_Tipo_Consulta], [ID_Sucursal], [Fecha], [Numero_Turno], [Estado])
        VALUES
        (@id_Tipo_Consulta, @id_Sucursal, @fecha, @numero_turno, @estado);
        SELECT SCOPE_IDENTITY();";

        cmd.Parameters.AddWithValue("id_Tipo_Consulta", turno.ID_Tipo_Consulta);
        cmd.Parameters.AddWithValue("id_Sucursal", turno.ID_Sucursal);
        cmd.Parameters.AddWithValue("fecha", turno.Fecha);
        cmd.Parameters.AddWithValue("numero_turno", turno.Numero_Turno);
        cmd.Parameters.AddWithValue("estado", turno.Estado);

        object id_Turno = cmd.ExecuteScalar();
        turno.ID_Turno = Convert.ToInt32(id_Turno);

        conexion.Close();

        return turno;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

Figura 51. Código C# para la generación de un turno

```

const sacarTurno = (e) => {
    const nuevoTurno = {
        id_Tipo_Consulta: turno.id_Consulta,
        id_Sucursal: turno.id_Sucursal,
        fecha: turno.fecha,
        numero_turno: turno.numeroTurno,
        estado: turno.estado,
    };

    const urlNuevoTurno = turnos.crearNuevoTurno();

    axios
        .post(urlNuevoTurno, nuevoTurno, head)
        .then(() => {
            incrementarContador(turno.id_Consulta, turno.id_Sucursal, turno.num);
            closeModal();
            abrirModalTurno();
            imprimirTicket();
        })
        .catch((error) => {
            console.error("Error fetching data:", error);
        });
};

```

Figura 52. Código JavaScript para el consumo de la API de nuevo turno

```

function imprimirTicket() {
  const contenidoTicket = `
    <div style="text-align: center;">
      <h1>EMAPA</h1>
      <p>${turno.sucursal}</p>
      <hr>
      <p>TURNO</p>
      <p>${turno.tipo}</p>
      <p>${turno.numeroTurno}</p>
    </div>
  `;

  const windowObj = window.open("", "", "width=100,height=100");

  windowObj.document.open();
  windowObj.document.write(
    <html>
      <head>
        <title>Ticket</title>
      </head>
      <body>
        ${contenidoTicket}
      </body>
    </html>
  );
  windowObj.document.close();
  windowObj.print();
  windowObj.close();
}

```

Figura 53. Código JavaScript para la impresión del ticket

En la Figura 52 se muestra el código que realiza una petición POST al servidor para ejecutar el código que se observa en la Figura 51, que abre una conexión a la base de datos para guardar un nuevo turno. En la Figura 53 se muestra el código utilizado para imprimir el ticket.

- **Pantalla de información**

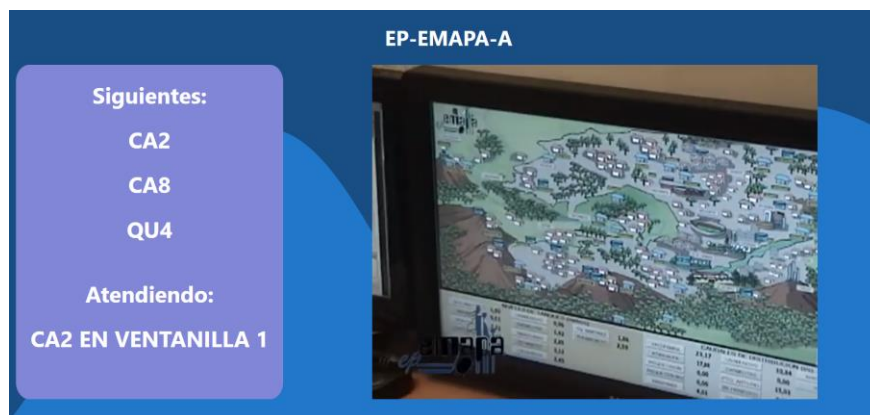


Figura 54. Pantalla de información

La Figura 54 muestra la pantalla de información en la que se muestran en orden los turnos que están en espera de ser atendidos, los turnos y en que ventanilla están siendo atendidos, además de un video que se reproduce en bucle.

```

public static List<Turno> DevolverTurnosSucursalEstado(int IdSucursal, int idEstado)
{
    try
    {
        List<Turno> lista = new List<Turno>();

        SqlConnection conexion = new SqlConnection(Properties.Settings.Default.ConexionDB);
        conexion.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = @"SELECT [ID_Turno], [ID_Tipo_Consulta], [ID_Sucursal], [Fecha], [Numero_Turno], [Estado]
                            FROM dbo.[TURNOS]
                            WHERE [ID_Sucursal] = @id AND [Estado] = @idEstado";

        cmd.Parameters.AddWithValue("@id", idSucursal);
        cmd.Parameters.AddWithValue("@idEstado", idEstado);

        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                Turno turno = new Turno();
                turno.ID_Turno = Convert.ToInt32(dr["ID_Turno"].ToString());
                turno.ID_Tipo_Consulta = Convert.ToInt32(dr["ID_Tipo_Consulta"].ToString());
                turno.ID_Sucursal = Convert.ToInt32(dr["ID_Sucursal"].ToString());
                turno.Fecha = Convert.ToDateTime(dr["Fecha"].ToString());
                turno.Numero_Turno = dr["Numero_Turno"].ToString();
                turno.Estado = Convert.ToInt32(dr["Estado"].ToString());

                lista.Add(turno);
            }
        }

        conexion.Close();

        return lista;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

Figura 55. Código C# para listar los turnos por sucursal y estado

```

useEffect(() => {
    axios
        .get(apiUrlTurnosEspera)
        .then((response) => {
            setTurnos(response.data);
        })
        .catch((error) => {
            console.error("Error fetching data:", error);
        });
}, [turnos, selectedValue, apiUrlTurnosEspera])

```

Figura 56. Código JavaScript para el consumo de la API para cargar turnos en espera

```

useEffect(() => {
    axios
        .get(apiUrlTurnosAtendiendo)
        .then((response) => {
            setTurnosAtendiendo(response.data);
        })
        .catch((error) => {
            console.error("Error fetching data:", error);
        });
}, [turnosAtendiendo, selectedValue, apiUrlTurnosAtendiendo])

```

Figura 57. Código JavaScript para el consumo de la API para cargar turnos que se están atendiendo

En la Figura 56 y Figura 57 se muestra el código que realiza una petición GET al servidor para ejecutar con diferentes parámetros el código de la Figura 55, que abre

una conexión a la base de datos para devolver la lista de turnos de una sucursal y con un estado en específico.

- **Pantalla de calificación**

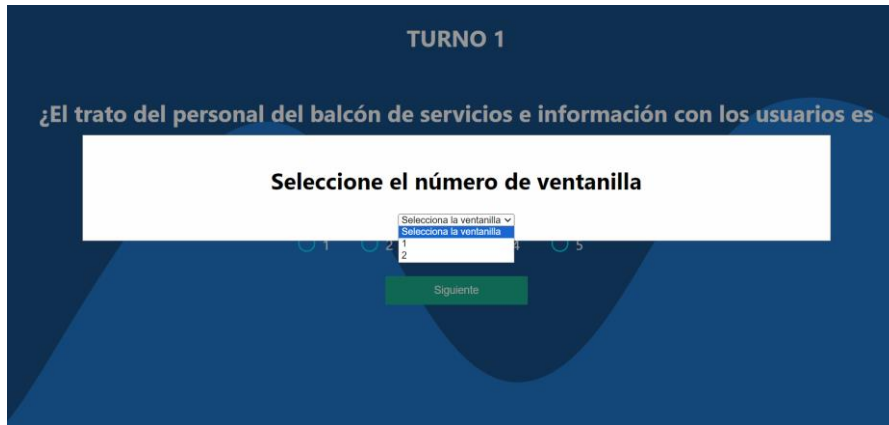


Figura 58. Selección de ventanilla (Pantalla de calificación)

En la Figura 58 se observa el inicio de la pantalla de calificación, en la que el usuario debe seleccionar de manera obligatoria el número de ventanilla de la que se van a calificar las atenciones.

NOTA: La encuesta se compone de 4 preguntas.

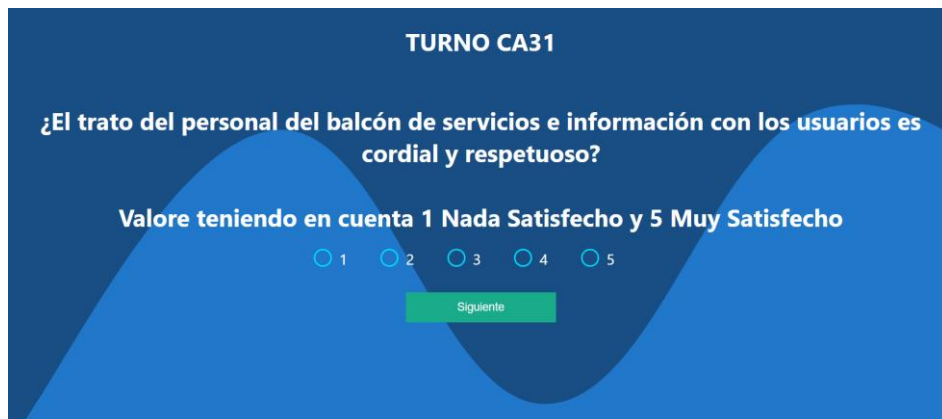


Figura 59. Encuesta de calificación (Pantalla de calificación)

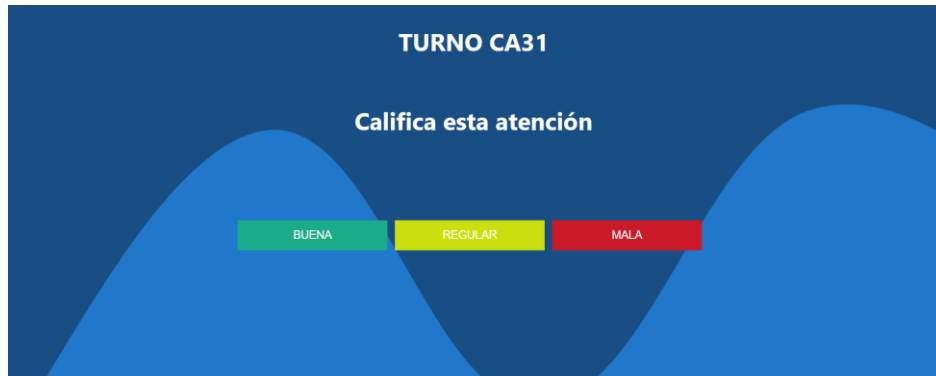


Figura 60. Calificación general (Pantalla de calificación)

La Figura 59 muestra una de las preguntas de la encuesta de calificación, en la que el usuario debe obligatoriamente seleccionar una opción para poder pasar a la siguiente, al finalizar las preguntas de la encuesta se muestra la pantalla que se observa en la Figura 60, en la que se debe seleccionar una opción de valoración general de la atención.

```

public static Calificacion ActualizarCalificacion(Calificacion calificacion)
{
    try
    {
        SqlConnection conexion = new SqlConnection(Properties.Settings.Default.ConexionDB);
        conexion.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = @"UPDATE [dbo].[CALIFICACION]
            SET [Pregunta_1] = @pregunta_1,
                [Pregunta_2] = @pregunta_2, [Pregunta_3] = @pregunta_3,
                [Pregunta_4] = @pregunta_4, [Valoracion] = @valoracion
            WHERE [ID_Atencion] = @id_Atencion";

        cmd.Parameters.AddWithValue("@id_Atencion", calificacion.ID_Atencion);
        cmd.Parameters.AddWithValue("@pregunta_1", calificacion.Pregunta_1);
        cmd.Parameters.AddWithValue("@pregunta_2", calificacion.Pregunta_2);
        cmd.Parameters.AddWithValue("@pregunta_3", calificacion.Pregunta_3);
        cmd.Parameters.AddWithValue("@pregunta_4", calificacion.Pregunta_4);
        cmd.Parameters.AddWithValue("@valoracion", calificacion.Valoracion);

        cmd.ExecuteNonQuery();

        conexion.Close();

        return calificacion;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

Figura 61. Código C# para actualizar la calificación

```

const calificar = (e) => {
  const calificacion = e.target.textContent.toUpperCase();
  const atencion = obtenerAtencion();

  if (atencion.ID_Atencion === atencionCalificada.ID_Atencion) {
    alert("Esta atencion ya fue calificada");
    reinicio();
  } else {
    const nuevaAtencion = {
      id_Atencion: atencion.ID_Atencion,
      pregunta_1: pregunta1,
      pregunta_2: pregunta2,
      pregunta_3: pregunta3,
      pregunta_4: pregunta4,
      valoracion: calificacion,
    };

    const urlActualizarCalificacion = calificacionesAPI.actualizarCalificacionPorID();
    if (atencion.ID_Atencion !== undefined) {
      axios
        .put(urlActualizarCalificacion, nuevaAtencion, head)
        .then((response) => {
          setAtencionCalificada(response.data.ActualizarCalificacionResult);
          reinicio();
        })
        .catch((error) => {
          console.error("Error fetching data:", error);
        });
    }
  }
};

```

Figura 62. Código JavaScript para el consumo de la API para actualizar la calificación

En la Figura 62 se observa el código que realiza una petición POST al servidor para ejecutar el código que se muestra en la Figura 61, que abre una conexión a la base de datos para actualizar los valores de calificación de una atención.

- **Pantalla para iniciar sesión**



Figura 63. Pantalla de inicio de sesión

En la Figura 63 se muestra la pantalla de inicio de sesión en la que el usuario debe ingresar sus credenciales (nombre de usuario y clave) para entrar a la pantalla de trabajador o a la de administrador. En caso de ingresar las credenciales de administrador, no es obligatorio seleccionar una ventanilla, esto solo es obligatorio para los usuarios trabajadores.

```

public static Usuario DevolverUsuario(string nombre, string clave)
{
    try
    {
        Usuario usuario = new Usuario();

        SqlConnection conexion = new SqlConnection(Properties.Settings.Default.ConexionDB);
        conexion.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = @"SELECT us.[ID_Usuario], us.[Nombre], us.[Clave], us.[ID_Trabajador], us.[ID_Rol], us.[Estado], r.[Nombre] as Rol
                            FROM dbo.[USUARIOS] as us, dbo.[ROLES] as r
                            WHERE us.[Nombre] = @nombre AND us.[Clave] = @clave AND us.[Estado] = 1 AND us.[ID_Trabajador] IN (SELECT [ID_Trabajador] FROM dbo.[TRABAJADORES]
                            WHERE [Estado] = 1) AND us.[ID_ROL] = r.[ID_Rol]";

        cmd.Parameters.AddWithValue("nombre", nombre);
        cmd.Parameters.AddWithValue("clave", clave);

        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                usuario.ID_Usuario = Convert.ToInt32(dr["ID_Usuario"].ToString());
                usuario.ID_Rol = Convert.ToInt32(dr["ID_Rol"].ToString());
                string rol = dr["Rol"].ToString();
                string token = tokenService.GenerateJwtToken(usuario.Nombre, rol);
                usuario.Token = token;
            }
        }

        conexion.Close();

        return usuario;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

Figura 64. Código C# para devolver un usuario

```

//Login
const Logeo = async () => {
  const url = usuariosAPI.logeo(nombre, clave);
  const response = await axios.get(url);

  const user = response.data;

  if(user.ID_Rol !== "" && user.ID_Rol != null) {
    if(user.ID_Rol === 1) {
      setUsuario(user);
      localStorage.setItem("user", user.ID_Usuario);
      localStorage.setItem("token", user.Token);
      localStorage.setItem("rol", user.ID_Rol);
      Redireccion(user.ID_Rol);
    } else if(user.ID_Rol !== 1 && ventanilla === 0) {
      alert("Selecciona la ventanilla");
    } else {
      setUsuario(user);
      localStorage.setItem("user", user.ID_Usuario);
      localStorage.setItem("token", user.Token);
      localStorage.setItem("rol", user.ID_Rol);
      localStorage.setItem("ventanilla", ventanilla);
      Redireccion(user.ID_Rol);
    }
  } else {
    alert('Usuario o contraseña incorrectos');
  }
}

```

Figura 65. Código JavaScript para el consumo de la API de inicio de sesión

La Figura 65 muestra el código que realiza una petición GET al servidor para ejecutar el código de la Figura 64 y dependiendo del rol del usuario devuelto realiza la redirección a la pantalla de administrador o de trabajador.

- **Control de sesión**

```

public string GenerateJwtToken(string username, string rol)
{
  var securityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(Secret_Key));
  var credentials = new SigningCredentials(securityKey, SecurityAlgorithms.HmacSha256);

  var claims = new[]
  {
    new Claim(JwtRegisteredClaimNames.Sub, username),
    new Claim(JwtRegisteredClaimNames.Jti, Guid.NewGuid().ToString()),
    new Claim(JwtRegisteredClaimNames.Iat, DateTimeOffset.UtcNow.ToUnixTimeSeconds().ToString(), ClaimValueTypes.Integer64),
    new Claim(ClaimTypes.Role, rol)
  };

  var token = new JwtSecurityToken(
    issuer: Emissor,
    audience: Receptor,
    claims: claims,
    expires: DateTime.UtcNow.AddHours(10),
    signingCredentials: credentials
  );

  var tokenString = new JwtSecurityTokenHandler().WriteToken(token);

  return tokenString;
}

```

Figura 66. Código C# para la creación del token de usuario

Al iniciar sesión se crea un JWT (JSON Web Token) con el nombre y el rol del usuario con el código que se muestra en la Figura 66. Para verificar que el token

es válido y corresponde al rol para el que se creó se utiliza el código que se presenta a continuación:

```
public bool ValidateToken(string token, string requiredRole)
{
    try
    {
        var tokenHandler = new JwtSecurityTokenHandler();
        var validationParameters = new TokenValidationParameters
        {
            ValidateIssuerSigningKey = true,
            IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(Secret_Key)),
            ValidateIssuer = false,
            ValidateAudience = false,
            ClockSkew = TimeSpan.Zero
        };

        ClaimsPrincipal claimsPrincipal = tokenHandler.ValidateToken(token, validationParameters, out SecurityToken validatedToken);

        if (claimsPrincipal != null && claimsPrincipal.Identity is ClaimsIdentity claimsIdentity)
        {
            ClaimsPrincipal principal = tokenHandler.ValidateToken(token, validationParameters, out validatedToken);
            if (principal != null)
            {
                if (claimsIdentity.HasClaim(ClaimTypes.Role, requiredRole))
                {
                    return true;
                }
            }
        }

        return false;
    }
    catch (Exception ex)
    {
        return false;
    }
}
```

Figura 67. Código C# para la validación del token

```
#!/ COMPROBACIÓN DE TOKEN Y ROL
useEffect(() => {
    axios
        .get(apiUrlValidacionToken, head)
        .then((response) => {
            if(!response.data) {
                navigate('/');
            }
        })
        .catch((error) => {
            console.error("Error:", error);
        });
}, [])
```

Figura 68. Código JavaScript para la validación del token

En la Figura 68 se muestra el código que realiza la validación del token de usuario para el rol requerido para ingresar a una pantalla. Esta comprobación se realiza en todas las pantallas para las que se requiere iniciar sesión.

- **Pantalla de trabajador**



Figura 69. Pantalla de trabajador

En la Figura 69 se observa la pantalla de trabajador, en el que se muestran los turnos que están en espera de atención para que el usuario pueda atenderlos. El turno que se seleccione para atender desaparece de la lista.



Figura 70. Ventana de atención de turno

En la Figura 70 se muestra la ventana que aparece al seleccionar un turno para atender, en ella se observa una en la que se visualiza la información sobre el turno y una sección para ingresar observaciones. Al seleccionar cancelar, la ventana se cierra y el turno vuelve a aparecer en la lista.

```

const showModal = (e) => {
  let idConsulta = e.target.id;

  let parrafos = document.getElementsByTagName("p");

  let tipoConsulta = "";

  for (const parrafo of parrafos) {
    if (parrafo.id === idConsulta) {
      tipoConsulta = parrafo.textContent;
    }
  }

  getContador(idConsulta, sucursal.ID_Sucursal)
  .then((contador) => {
    let n = contador.Numero;

    let numero = tipoConsulta.substring(0, 2) + n;

    let modal = document.getElementById("myModal");

    if (sucursal !== null) {
      //!FECHA
      let fecha = new Date();

      const milisegundos = fecha.getTime();
      const fechaFormateada = `\\Date(${milisegundos})\\`;

      let ticket = {
        id_Consulta: idConsulta,
        id_Sucursal: selectedValue,
        fecha: fechaFormateada,
        estado: 3,
        sucursal: sucursal.Nombre,
        tipo: tipoConsulta,
        num: n,
        numeroTurno: numero,
      };
      setTurno(ticket);
      modal.style.display = "block";
    }
  })
  .catch((error) => {
    console.error(error);
  });
};

```

Figura 71. Código JavaScript para cargar la información del turno

En la Figura 71 se muestra el código que se utiliza para abrir la ventana y cargar la información del turno seleccionado.

```

public static Atencion ActualizarAtencion(Atencion atencion)
{
    try
    {
        SqlConnection conexion = new SqlConnection(Properties.Settings.Default.ConexionDB);
        conexion.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = @"UPDATE [dbo].[ATENCION]
                            SET [Ventanilla] = @ventanilla,
                                [Estado] = @estado, [Fecha_Inicio] = @fecha_Inicio,
                                [Fecha_Final] = @fecha_Final,
                                [Observacion] = @observacion
                            WHERE [ID_Atencion] = @id";

        cmd.Parameters.AddWithValue("id", atencion.ID_Atencion);
        cmd.Parameters.AddWithValue("ventanilla", atencion.Ventanilla);
        cmd.Parameters.AddWithValue("estado", atencion.Estado);
        cmd.Parameters.AddWithValue("fecha_Inicio", atencion.Fecha_Inicio);
        cmd.Parameters.AddWithValue("fecha_Final", atencion.Fecha_Final);
        cmd.Parameters.AddWithValue("observacion", atencion.Observacion);

        cmd.ExecuteNonQuery();

        conexion.Close();

        return atencion;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

Figura 72. Código C# para actualizar la atención

```

public static Turno ActualizarEstadoTurno(Turno turno)
{
    try
    {
        SqlConnection conexion = new SqlConnection(Properties.Settings.Default.ConexionDB);
        conexion.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = @"UPDATE [dbo].[TURNOS]
                            SET
                                [Estado] = @estado
                            WHERE [ID_Turno] = @id";

        cmd.Parameters.AddWithValue("id", turno.ID_Turno);
        cmd.Parameters.AddWithValue("estado", turno.Estado);

        cmd.ExecuteNonQuery();

        conexion.Close();

        return turno;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

Figura 73. Código C# para actualizar el estado del turno

```

const completarAtencion = (e) => {
  let observacion = document.getElementById('observaciones').value;

  if(observacion === '') {
    observacion = 'NINGUNA';
  }

  let fecha = new Date();

  let ml = fecha.getTime();

  const fechaFormateada = `\\Date(${ml})\\`;

  const estado = 5;

  let userID = JSON.parse(localStorage.getItem("user"));

  let turnoID = turno.ID_Turno;

  let atencion = {
    id_Atencion: atencionActual.ID_Atencion,
    id_Usuario: userID,
    id_Turno: turnoID,
    ventanilla: ventanilla,
    estado: estado,
    fecha_Inicio: fechaInicio,
    fecha_Final: fechaFormateada,
    observacion: observacion
  }

  const urlActualizarAtencion = atencionesAPI.actualizarAtencionPorID();
  axios
    .put(urlActualizarAtencion, atencion, head)
    .then(() => {
    })
    .catch((error) => {
      console.error("Error fetching data:", error);
    });

  const urlActualizarTurno = turnosAPI.actualizarEstadoTurnoPorID();

  const nuevoTurno = {
    id_Turno: atencion.id_Turno,
    estado: 5
  }

  axios
    .put(urlActualizarTurno, nuevoTurno, head)
    .then(() => {
    })
    .catch((error) => {
      console.error("Error fetching data:", error);
    });

  let modal = document.getElementById("myModal");

  modal.style.display = "none";
}

```

Figura 74. Código JavaScript para el consumo de las API de atención y turno

La Figura 74 muestra el código que realiza peticiones PUT al servidor para:

- Actualizar la información de la atención a través del código que se muestra en la Figura 72.

- Actualizar el estado del turno atendido mediante el código que se muestra en la Figura 73.
- **Pantalla de administrador**



Figura 75. Pantalla de administrador

En la Figura 75 se observa la pantalla principal del administrador a través de la que navega entre las distintas pantallas para administrar los datos.

Numero Turno	Usuario	Trabajador	Fecha	Hora Turno	Hora Inicio	Hora Final	Sucursal	Calificacion	Observación
1	CO2	JAVIER JIMENEZ CASTRO	2023-11-21	16:55	16:59	17:02	CENTRO	Calificacion	Observación
2	CO3	JAVIER JIMENEZ CASTRO	2023-11-21	16:58	17:03	17:10	CENTRO	Calificacion	Observación

Figura 76. Pantalla de administración de atenciones

La Figura 76 muestra la pantalla en la que se administran los datos sobre las atenciones, que contiene una sección para aplicar filtrado a los datos, dentro de cada fila de datos se encuentran 2 botones para visualizar la calificación y las observaciones respectivamente de cada atención. Además, se encuentra una paginación para facilitar la lectura y distribución de los datos y los botones para generar reportes.

NOTA: Los botones para generar reportes se encuentran en todas las pantallas de administración de datos.

```
public static List<Atencion> DevolverListaDatosAtenciones()
{
    try
    {
        List<Atencion> lista = new List<Atencion>();
        SqlConnection conexion = new SqlConnection(Properties.Settings.Default.ConexionDB);
        conexion.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = @"SELECT at.[ID_Atencion], at.[ID_Turno], at.[ID_Usuario], at.[Ventanilla], at.[Estado], at.[Fecha_Inicio], at.[Fecha_Final], at.[Observacion],
        us.[Nombre], tr.[Primer_Nombre], tr.[Segundo_Nombre], tr.[Primer_Apellido],
        tr.[Segundo_Apellido], tu.[Numero_Turno], tu.[ID_Sucursal], tu.[Fecha] as Fecha_Turno,
        ca.[Pregunta_1], ca.[Pregunta_2], ca.[Pregunta_3], ca.[Pregunta_4], ca.[Valoracion]

        FROM dbo.[USUARIOS] as us, dbo.[TRABAJADORES] as tr, dbo.[TURNOS] as tu, dbo.[ATENCION] as at, dbo.[CALIFICACION] as ca
        WHERE at.[ID_Usuario] = us.[ID_Usuario] AND at.[ID_Turno] = tu.[ID_Turno]
        AND us.[ID_Trabajador] = tr.[ID_Trabajador] AND at.[ID_Atencion] = ca.[ID_Atencion]";

        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                Atencion atencion = new Atencion();
                atencion.ID_Atencion = Convert.ToInt32(dr["ID_Atencion"].ToString());
                atencion.ID_Turno = Convert.ToInt32(dr["ID_Turno"].ToString());
                atencion.ID_Usuario = Convert.ToInt32(dr["ID_Usuario"].ToString());
                atencion.Ventanilla = Convert.ToInt32(dr["Ventanilla"].ToString());
                atencion.Estado = Convert.ToInt32(dr["Estado"].ToString());
                atencion.Fecha_Turno = Convert.ToDateTime(dr["Fecha_Turno"].ToString());
                atencion.Fecha_Inicio = Convert.ToDateTime(dr["Fecha_Inicio"].ToString());
                atencion.Fecha_Final = Convert.ToDateTime(dr["Fecha_Final"].ToString());
                atencion.Observacion = dr["Observacion"].ToString();
                atencion.Nombre_Usuario = dr["Nombre"].ToString();
                string primer_nom = dr["Primer_Nombre"].ToString();
                string segundo_nom = dr["Segundo_Nombre"].ToString();
                string primer_ape = dr["Primer_Apellido"].ToString();
                string segundo_ape = dr["Segundo_Apellido"].ToString();
                atencion.Nombre_Trabajador = primer_nom + " " + segundo_nom + " " + primer_ape + " " + segundo_ape;
                atencion.Numero_Turno = dr["Numero_Turno"].ToString();
                atencion.Sucursal = Convert.ToInt32(dr["ID_Sucursal"].ToString());
                atencion.Pregunta_1 = Convert.ToInt32(dr["Pregunta_1"].ToString());
                atencion.Pregunta_2 = Convert.ToInt32(dr["Pregunta_2"].ToString());
                atencion.Pregunta_3 = Convert.ToInt32(dr["Pregunta_3"].ToString());
                atencion.Pregunta_4 = Convert.ToInt32(dr["Pregunta_4"].ToString());
                atencion.Valoracion = dr["Valoracion"].ToString();

                lista.Add(atencion);
            }
        }

        conexion.Close();

        return lista;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

Figura 77. Código C# para listar las atenciones

```
const fetchData = async (url, setData) => {
    try {
        const response = await axios.get(url);
        const data = await response.json();
        setData(data);
    } catch (error) {
        console.error('Error fetching data:', error);
    }
};

//! CARGA DE ATENCIONES
useEffect(() => {
    fetchData(apiUrlAtenciones, setAtenciones);
}, [atenciones, apiUrlAtenciones]);
```

Figura 78. Código JavaScript para el consumo de la API para listar las atenciones

En la Figura 78 se muestra el código que realiza una petición GET al servidor para ejecutar el código que se observa en la Figura 77, que abre una conexión a la base de datos para obtener la información de las atenciones.

Numero Turno	Estado	Fecha	Motivo de Consulta	Sucursal	
1	CO1	ATENDIENDO	2023-11-21	CONSULTAS	CENTRO
2	PE1	ATENDIENDO	2023-11-21	PETICION DE SERVICIO	CENTRO
3	CO2	ATENDIDO	2023-11-21	CONSULTAS	CENTRO
4	CO3	ATENDIDO	2023-11-21	CONSULTAS	CENTRO
5	PE1	EN ESPERA	2023-11-21	PETICION DE SERVICIO	CENTRO

Figura 79. Página de administración de turnos

En la Figura 79 se observa la pantalla en la que se administran los datos de los turnos generados.

```

public static List<Turno> DevolverListaTurnos()
{
    try
    {
        List<Turno> lista = new List<Turno>();
        SqlConnection conexion = new SqlConnection(Properties.Settings.Default.ConexionDB);
        conexion.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = @"SELECT [ID_Turno], [ID_Tipo_Consulta], [ID_Sucursal], [Fecha], [Numero_Turno], [Estado]
        FROM dbo.[TURNOS]";

        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                Turno turno = new Turno();
                turno.ID_Turno = Convert.ToInt32(dr["ID_Turno"].ToString());
                turno.ID_Tipo_Consulta = Convert.ToInt32(dr["ID_Tipo_Consulta"].ToString());
                turno.ID_Sucursal = Convert.ToInt32(dr["ID_Sucursal"].ToString());
                turno.Fecha = Convert.ToDateTime(dr["Fecha"].ToString());
                turno.Numero_Turno = dr["Numero_Turno"].ToString();
                turno.Estado = Convert.ToInt32(dr["Estado"].ToString());

                lista.Add(turno);
            }
        }

        conexion.Close();

        return lista;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

Figura 80. Código C# para listar todos los turnos


```

const fetchData = async (url, setData) => {
  try {
    const response = await axios.get(url);
    const data = await response.json();
    setData(data);
  } catch (error) {
    console.error("Error fetching data:", error);
  }
};

//! CARGA DE TURNOS, ESTADOS Y SUCURSALES
useEffect(() => {
  fetchData(apiUrlTurnos, setTurnos);
  fetchData(apiUrlEstados, setEstados);
  fetchData(apiUrlSucursales, setSucursales);
}, [turnos, estados, sucursales]);

```

Figura 81. Código JavaScript para el consumo de la API para cargar los turnos

La Figura 81 muestra el código que realiza una petición GET al servidor para ejecutar el código de la Figura 80, que abre una conexión a la base de datos para obtener la lista de turnos.

- **Pantalla de administración de roles**



Figura 82. Pantalla de administración de roles

La Figura 82 muestra la pantalla en la que se administran los datos de roles, en la que se puede ingresar, editar o desactivar un rol.

```

public static List<Rol> DevolverListaRoles()
{
    try
    {
        List<Rol> lista = new List<Rol>();
        SqlConnection conexion = new SqlConnection(Properties.Settings.Default.ConexionDB);
        conexion.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = @"SELECT [ID_Rol], [Nombre], [Estado]
                            FROM dbo.[ROLES]";

        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())
            {
                Rol rol = new Rol();
                rol.ID_Rol = Convert.ToInt32(dr["ID_Rol"].ToString());
                rol.Nombre = dr["Nombre"].ToString();
                rol.Estado = Convert.ToInt32(dr["Estado"].ToString());

                lista.Add(rol);
            }
        }

        conexion.Close();

        return lista;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

```

Figura 83. Código C# para listar los roles

```

const fetchData = async (url, setData) => {
    try {
        const response = await axios.get(url);
        const data = await response.json();
        setData(data);
    } catch (error) {
        console.error("Error fetching data:", error);
    }
};

//! CARGA DE ROLES Y ESTADOS
useEffect(() => {
    fetchData(apiUrlRoles, setRoles);
    fetchData(apiUrlEstados, setEstados);
}, [roles, estados]);

```

Figura 84. Código JavaScript para el consumo de la API para cargar los roles

En la Figura 84, se observa el código que realiza una petición GET al servidor para ejecutar el código que se muestra en la Figura 83, que abre una conexión a la base de datos para obtener los datos de los roles.

Las otras pantallas de administración de datos tienen los mismos elementos, la única sección que varía es la tabla de datos, por lo que las otras pantallas de administración, ventanas y bloques de código utilizados para el desarrollo de la aplicación web se encuentran en el Anexo D.

3.2.4 Fase IV: Pruebas

a. Pruebas de aplicación

Las pruebas de aplicación se realizaron con el objetivo de validar el cumplimiento de los requerimientos a través de casos de prueba establecidos desde el punto de vista del desarrollador.

Tabla 27. Prueba de aplicación – Requerimiento R001

ID		Requerimiento	
R001		Listar sucursales	
Caso	Resultado	Observación	Estado
No hay sucursales disponibles	No se lista ninguna sucursal	Al no haber ninguna sucursal para seleccionar, no se puede acceder a ninguna otra pantalla de la aplicación	Aprobado
Hay sucursales disponibles	Se listan todas las sucursales correctamente		Aprobado

Tabla 28. Prueba de aplicación – Requerimiento R002

ID		Requerimiento	
R002		Selección de sucursal	
Caso	Resultado	Observación	Estado
No se selecciona ninguna sucursal	Se muestra un mensaje para alertar al usuario para informarle que debe seleccionar una sucursal para acceder a las otras pantallas	El mensaje de alerta solo aparece cuando se intenta acceder a otra pantalla	Aprobado
Se selecciona una sucursal	Se redirecciona sin problemas a las otras pantallas		Aprobado

Tabla 29. Prueba de aplicación – Requerimiento R003

ID		Requerimiento	
R003		Generación de turnos	
Caso	Resultado	Observación	Estado
No se listan los tipos de consulta para sacar turnos	No se muestra ninguna opción para sacar turno	La única opción que se muestra es para consultar la planilla de agua en una aplicación externa ya existente	Aprobado
Se listan todos los tipos de consulta correctamente	Se muestra una opción por tipo de consulta para poder sacar un turno		Aprobado
Se selecciona la opción para consultar la planilla de agua	Se redirige a una página externa perteneciente a la empresa		Aprobado
Se selecciona un tipo de consulta para sacar un turno	Se abre una ventana en la que se muestra la información acerca del turno	En esta ventana se encuentran 2 botones para aceptar o cancelar el turno	Aprobado
Se cancela la generación del turno	Se cierra la ventana, el turno no se guarda en la base de datos y el contador no se aumenta		Aprobado
Se confirma la generación del turno	Se muestra una ventana para informar al usuario que se generó el turno	La ventana que confirma la generación del turno debe cerrarse por el usuario, seleccionado la opción “Aceptar”	Aprobado
	El turno se guarda en la base de datos y el contador aumenta		
	Se imprime el ticket con el número de turno		

Tabla 30. Prueba de aplicación – Requerimiento R004

ID		Requerimiento	
R004		Selección de ventanilla	
Caso	Resultado	Observación	Estado
No se listan las ventanillas	No hay ninguna ventanilla para seleccionar	Al no haber ninguna ventanilla para seleccionar, no se puede iniciar sesión como usuario con rol trabajador	Aprobado
Se listan las ventanillas	Se cargan las ventanillas para que puedan ser seleccionadas por el usuario en caso de ser necesario	Un usuario con rol administrador no tiene que seleccionar ninguna ventanilla para poder iniciar sesión	Aprobado
No se selecciona ninguna ventanilla	En caso de ser un usuario con rol trabajador, se muestra un mensaje de alerta avisando al usuario que debe seleccionar una ventanilla para acceder	El mensaje de alerta no se muestra a usuarios con rol administrador	Aprobado
Se selecciona una ventanilla	No se muestra el mensaje de alerta por la selección de ventanilla		Aprobado

Tabla 31. Prueba de aplicación – Requerimiento R005

ID		Requerimiento	
R005		Autenticación del usuario	
Caso	Resultado	Observación	Estado
Se ingresan credenciales incorrectas	Se muestra un mensaje de alerta informando al usuario que su nombre de usuario o clave son incorrectas	En caso de ser un usuario con rol trabajador y no haber seleccionado una ventanilla, el mensaje de credenciales incorrectas tiene prioridad	Aprobado
Se ingresan credenciales correctas	Se realiza una redirección dependiendo del rol del usuario		Aprobado

Tabla 32. Prueba de aplicación – Requerimiento R006

ID		Requerimiento	
R006		Listar turnos en espera de atención	
Caso	Resultado	Observación	Estado
No hay turnos en espera de atención	La tabla que se muestra al usuario con rol trabajador se encuentra vacía	El botón para atender un turno solo se muestra si hay al menos un turno en la tabla	Aprobado
Hay turnos en espera de atención	La tabla carga todos los turnos en espera de atención de dicha sucursal	El botón para atender un turno solo se encuentra activo para el primer dato de la tabla, de manera que se atiendan en orden	Aprobado

Tabla 33. Prueba de aplicación – Requerimiento R007

ID		Requerimiento	
R007		Atender turno	
Caso	Resultado	Observación	Estado
No se selecciona el turno para atender	El primer turno de la tabla seguirá siendo el único al que se le puede brindar atención	Solo se puede atender al primer turno de la tabla, de manera que para atender al segundo turno primero se debe finalizar la atención del primero	Aprobado
Se selecciona el turno para atender	El turno desaparece de la tabla de turnos en espera de atención Se abre una ventana que muestra la información del turno y un apartado para ingresar las observaciones	En esta ventana se encuentran 2 botones para finalizar o cancelar la atención	Aprobado
	Se guarda la atención en la base de datos con algunos valores por defecto	Los valores por defecto guardados en la base de	

	Se guarda una calificación en la base de datos con valores por defecto	datos pueden ser eliminados o actualizados dependiendo de las decisiones del usuario	
Caso	Resultado	Observación	Estado
Se cancela la atención	Se cierra la ventana y el turno vuelve a aparecer en la primera posición de la tabla Se elimina de la base de datos la calificación creada para la atención Se elimina de la base de datos la atención creada		Aprobado
Se finaliza la atención	Se cierra la ventana y el turno ya no aparece en la tabla Se actualizan los valores de la atención en la base de datos		Aprobado

Tabla 34. Prueba de aplicación – Requerimiento R008

ID	Requerimiento		
R008	Visualización de atenciones		
Caso	Resultado	Observación	Estado
No hay atenciones	La tabla que se muestra al usuario con rol administrador se encuentra vacía	El botón para atender un turno solo se muestra si hay al menos un turno en la tabla	Aprobado
Hay atenciones	La tabla muestra los datos especificados de las atenciones De cada atención se muestra una opción para visualizar las observaciones y otra para visualizar la calificación		Aprobado
Se selecciona la opción para visualizar las observaciones de la atención	Se abre una ventana que muestra las observaciones de la atención	Para cerrar la ventana se encuentra un botón	Aprobado

Caso	Resultado	Observación	Estado
Se selecciona la opción para visualizar la calificación de la atención	Se abre una ventana que muestra la información detallada de los valores de calificación de la atención	Para cerrar la ventana se encuentra un botón	Aprobado

Tabla 35. Prueba de aplicación – Requerimiento R009

ID	Requerimiento		
R009	Gestión de datos		
Caso	Resultado	Observación	Estado
No hay datos sobre los turnos (Pantalla de administración de datos sobre los turnos)	La tabla que se muestra al usuario con rol administrador se encuentra vacía	No se puede crear, modificar ni eliminar ningún dato sobre los turnos	Aprobado
Hay datos sobre los turnos (Pantalla de administración de datos sobre los turnos)	La tabla carga y muestra correctamente los datos sobre los turnos	No se puede crear, modificar ni eliminar ningún dato sobre los turnos	Aprobado
No hay datos sobre los usuarios (Pantalla de administración de datos sobre los usuarios)	La tabla que se muestra al usuario con rol administrador se encuentra vacía	Los botones para modificar o desactivar un usuario solo se muestran si hay al menos un usuario en la tabla, el botón para crear un nuevo usuario se muestra siempre independientemente de los datos de la tabla	Aprobado
Hay datos sobre los usuarios (Pantalla de administración de datos sobre los usuarios)	La tabla carga y muestra correctamente los datos sobre los usuarios	Se muestran botones para modificar o desactivar un usuario en específico	Aprobado

Caso	Resultado	Observación	Estado
Se selecciona la opción para crear un nuevo usuario	Se abre una ventana que muestra los valores que se deben ingresar para crear el usuario	En la ventana se encuentran 2 botones, uno para cancelar y otro para guardar	Aprobado
Se cancela la creación	Se cierra la ventana	Este caso se comprobó en todas las pantallas de administración de datos, concluyendo en el mismo resultado	Aprobado
Se guarda el nuevo usuario	Se cierra la ventana Se guarda el nuevo usuario en la base de datos	El único control que se encuentra para el ingreso de datos es: <ul style="list-style-type: none"> Control de trabajador, para que se seleccione el trabajador al que pertenece ese usuario 	Aprobado
Se selecciona la opción para modificar los datos de un usuario	Se abre una ventana en la que se cargan los datos del usuario para que puedan ser modificados	En la ventana se encuentran 2 botones, uno para cancelar y otro para guardar	Aprobado
Se cancela la modificación de los datos	Se cierra la ventana	Este caso se comprobó en todas las pantallas de administración de datos, concluyendo en el mismo resultado	Aprobado
Se guarda el usuario modificado	Se cierra la ventana Se guardan los datos modificados en la base de datos	El único control que se encuentra para el ingreso de datos es: <ul style="list-style-type: none"> Control de trabajador, para que se seleccione el 	Aprobado

Caso	Resultado	Observación	Estado
		trabajador al que pertenece ese usuario	
Se selecciona la opción para desactivar u activar un usuario	Se abre una ventana que muestra un mensaje para confirmar la acción	El texto del botón y del mensaje de la ventana es diferente dependiendo de si el usuario está activo o no, y en la ventana se encuentran 2 botones, uno para cancelar y otro para confirmar	Aprobado
Se cancela la activación o desactivación	Se cierra la ventana	Este caso se comprobó en todas las pantallas de administración de datos, concluyendo en el mismo resultado	Aprobado
Se confirma la activación o desactivación del usuario	Se cierra la ventana Se modifica y se guarda el estado del usuario en la base de datos		Aprobado
No hay datos sobre los trabajadores (Pantalla de administración de datos sobre los trabajadores)	La tabla que se muestra al usuario con rol administrador se encuentra vacía	Los botones para modificar o desactivar un trabajador solo se muestran si hay al menos un trabajador en la tabla, el botón para crear un nuevo trabajador se muestra siempre independientemente de los datos de la tabla	Aprobado
Hay datos sobre los trabajadores (Pantalla de administración de	La tabla carga y muestra correctamente los datos sobre los trabajadores	Se muestran botones para modificar o desactivar un trabajador en específico	Aprobado

datos sobre los trabajadores)			
Caso	Resultado	Observación	Estado
Se selecciona la opción para crear un nuevo trabajador	Se abre una ventana que muestra los valores que se deben ingresar para crear el trabajador	En la ventana se encuentran 2 botones, uno para cancelar y otro para guardar	Aprobado
Se guarda el nuevo trabajador	Se cierra la ventana Se guarda el nuevo trabajador en la base de datos	El único control que se encuentra en el ingreso de datos es: <ul style="list-style-type: none"> Control de cédula, para que se ingrese una cadena numérica de 10 dígitos 	Aprobado
Se selecciona la opción para modificar los datos de un trabajador	Se abre una ventana en la que se cargan los datos del trabajador para que puedan ser modificados	En la ventana se encuentran 2 botones, uno para cancelar y otro para guardar	Aprobado
Se guarda el trabajador modificado	Se cierra la ventana Se guardan los datos modificados en la base de datos	El único control que se encuentra en el ingreso de datos es: Control de cédula, para que se ingrese una cadena numérica de 10 dígitos	Aprobado
Se selecciona la opción para desactivar o activar un trabajador	Se abre una ventana que muestra un mensaje para confirmar la acción	El texto del botón y del mensaje de la ventana es diferente dependiendo de si el trabajador está activo o no, y en la ventana se encuentran 2 botones, uno para cancelar y otro para confirmar	Aprobado

Caso	Resultado	Observación	Estado
Se confirma la activación o desactivación del trabajador	Se cierra la ventana		Aprobado
No hay datos sobre las sucursales (Pantalla de administración de datos sobre las sucursales)	La tabla que se muestra al usuario con rol administrador se encuentra vacía	Los botones para modificar o desactivar una sucursal solo se muestran si hay al menos una sucursal en la tabla, el botón para crear una nueva sucursal se muestra siempre independientemente de los datos de la tabla	Aprobado
Hay datos sobre las sucursales (Pantalla de administración de datos sobre las sucursales)	La tabla carga y muestra correctamente los datos sobre las sucursales	Se muestra botones para modificar o desactivar una sucursal en específico	Aprobado
Se selecciona la opción para crear una nueva sucursal	Se abre una ventana que muestra los valores que se deben ingresar para crear la sucursal	En la ventana se encuentran 2 botones, uno para cancelar y otro para guardar	Aprobado
Se guarda la nueva sucursal	Se cierra la ventana Se guarda la nueva sucursal en la base de datos	No se encuentran controles para el ingreso de datos	Aprobado
Se selecciona la opción para modificar los datos de una sucursal	Se abre una ventana en la que se cargan los datos del usuario para que puedan ser modificados	En la ventana se encuentran 2 botones, uno para cancelar y otro para guardar	Aprobado

Caso	Resultado	Observación	Estado
Se guarda la sucursal modificada	Se cierra la ventana Se guardan los datos modificados en la base de datos	No se encuentran controles para el ingreso de datos	Aprobado
Se selecciona la opción para desactivar o activar una sucursal	Se abre una ventana que muestra un mensaje para confirmar la acción	El texto del botón y del mensaje de la ventana es diferente dependiendo de si la sucursal está activa o no, y en la ventana se encuentran 2 botones, uno para cancelar y otro para confirmar	Aprobado
Se confirma la activación o desactivación de la sucursal	Se cierra la ventana Se modifica y se guarda el estado de la sucursal en la base de datos		Aprobado
No hay datos sobre los tipos de consulta (Pantalla de administración de datos sobre los tipos de consulta)	La tabla que se muestra al usuario con rol administrador se encuentra vacía	Los botones para modificar o desactivar un tipo de consulta solo se muestran si hay al menos un tipo de consulta en la tabla, el botón para crear un nuevo tipo de consulta se muestra siempre independientemente de los datos de la tabla	Aprobado
Hay datos sobre los tipos de consulta (Pantalla de administración de datos sobre los tipos de consulta)	La tabla carga y muestra correctamente los datos sobre los tipos de consulta	Se muestran botones para modificar o desactivar un tipo de consulta en específico	Aprobado

Caso	Resultado	Observación	Estado
Se selecciona la opción para crear un nuevo tipo de consulta	Se abre una ventana que muestra los valores que se deben ingresar para crear el tipo de consulta	En la ventana se encuentran 2 botones, uno para cancelar y otro para guardar	Aprobado
Se guarda el nuevo tipo de consulta	Se cierra la ventana Se guarda el nuevo tipo de consulta en la base de datos	No se encuentran controles para el ingreso de datos	Aprobado
Se selecciona la opción para modificar los datos de un tipo de consulta	Se abre una ventana en la que se cargan los datos del tipo de consulta para que puedan ser modificados	En la ventana se encuentran 2 botones, uno para cancelar y otro para guardar	Aprobado
Se guarda el tipo de consulta modificado	Se cierra la ventana Se guardan los datos modificados en la base de datos	No se encuentran controles para el ingreso de datos	Aprobado
Se selecciona la opción para desactivar o activar un tipo de consulta	Se abre una ventana que muestra un mensaje para confirmar la acción	El texto del botón y del mensaje es diferente dependiendo de si el tipo de consulta está activo o no, y en la ventana se encuentran 2 botones, uno para cancelar y otro para confirmar	Aprobado
Se confirma la activación o desactivación del tipo de consulta	Se cierra la ventana Se modifica y se guarda el estado del tipo de consulta en la base de datos		Aprobado
No hay datos sobre los roles (Pantalla de administración de datos sobre los roles)	La tabla que se muestra al usuario con rol administrador se encuentra vacía	Los botones para modificar o desactivar un rol solo se muestran si hay al menos un rol en la	Aprobado

Caso	Resultado	Observación	Estado
		tabla, el botón para crear un nuevo rol se muestra siempre independientemente de los datos de la tabla	
Hay datos sobre los roles (Pantalla de administración de datos sobre los roles)	La tabla carga y muestra correctamente los datos sobre los roles	Se muestran botones para modificar o desactivar un rol en específico	Aprobado
Se selecciona la opción para crear un nuevo rol	Se abre una ventana que muestra los valores que se deben ingresar para crear el rol	En la ventana se encuentran 2 botones, uno para cancelar y otro para guardar	Aprobado
Se guarda el nuevo rol	Se cierra la ventana Se guarda el nuevo rol en la base de datos	No se encuentran controles para el ingreso de datos	Aprobado
Se selecciona la opción para modificar los datos de un rol	Se abre una ventana en la que se cargan los datos del rol para que puedan ser modificados	En la ventana se encuentran 2 botones, uno para cancelar y otro para guardar	Aprobado
Se guarda el rol modificado	Se cierra la ventana Se guardan los datos modificados en la base de datos	No se encuentran controles para el ingreso de datos	Aprobado
Se selecciona la opción para desactivar o activar un rol	Se abre una ventana que muestra un mensaje para confirmar la acción	El texto del botón y del mensaje de la ventana es diferente dependiendo de si el usuario está activo o no, y en la ventana se encuentran 2 botones, uno para cancelar y otro para confirmar	Aprobado

Caso	Resultado	Observación	Estado
Se confirma la activación o desactivación del rol	Se cierra la ventana Se modifica y se guarda el estado del rol en la base de datos		Aprobado

Tabla 36. Prueba de aplicación – Requerimiento R010

ID	Requerimiento		
R010	Pantalla de información		
Casos	Resultado	Observación	Estado
No hay turnos en espera de atención o siendo atendidos en dicha sucursal	No se muestra ningún dato ni en la sección de turnos en espera, ni en la de turnos que se están atendiendo	Los datos no afectan a la sección en la que se reproduce el vídeo	Aprobado
Hay turnos en espera de atención o siendo atendidos en dicha sucursal	Se muestran los datos en sus respectivas secciones	Los datos se actualizan automáticamente dependiendo de ciertos eventos	Aprobado

Tabla 37. Prueba de aplicación – Requerimiento R011

ID	Requerimiento		
R011	Pantalla de calificación		
Casos	Resultado	Observación	Estado
No hay ventanillas en dicha sucursal	No se carga ninguna ventanilla para seleccionar	Al no haber ninguna ventanilla para seleccionar, no se puede acceder a la encuesta de satisfacción	Aprobado
Hay ventanillas en dicha sucursal	Se cargan las ventanillas correctamente para que puedan ser seleccionadas	La ventanilla se selecciona una sola vez hasta se regrese a la pantalla principal y se acceda nuevamente a la pantalla de calificación	Aprobado

Caso	Resultado	Observación	Estado
No se selecciona la ventanilla	No se cierra la ventana de selección		Aprobado
Se selecciona una ventanilla	Se cierra la ventana de selección		Aprobado
No se selecciona ninguna opción en la encuesta y se intenta pasar a la siguiente pregunta	Se muestra un mensaje de alerta informando al usuario que debe seleccionar una opción para poder pasar a la siguiente pregunta		Aprobado
Se selecciona una opción y se intenta pasar a la siguiente pregunta	Se muestra la siguiente pregunta sin inconvenientes	La encuesta consta de 4 preguntas y la función se repite hasta pasar de la última pregunta	Aprobado
Se pasa de la última de pregunta	Se muestra la ventana para que el usuario califique de manera general la atención	Esta calificación consta de 3 opciones: <ul style="list-style-type: none"> • Buena • Regular • Mala 	Aprobado
No se selecciona ninguna de las opciones de la ventana de valoración general	No se guardan los valores de calificación		Aprobado
Se selecciona una opción en la ventana de valoración	Se actualizan los valores de calificación de la atención en la base de datos		Aprobado

Tabla 38. Prueba de aplicación – Requerimiento R012

ID		Requerimiento	
R012		Filtrado de datos	
Casos	Resultado	Observación	Estado
No se aplica ningún filtro para los datos de la tabla (Pantalla de administración de datos sobre las atenciones)	Se muestra la lista completa de datos en la tabla	Este caso se comprobó en todas las pantallas de administración de datos, concluyendo en el mismo resultado	Aprobado
Se aplican los filtros para los datos de la tabla (Pantalla de administración de datos sobre las atenciones)	Se muestra la lista filtrada de datos en la tabla	Este caso se comprobó en todas las pantallas de administración de datos, concluyendo en el mismo resultado	Aprobado

Tabla 39. Prueba de aplicación – Requerimiento R013

ID		Requerimiento	
R013		Generación de reportes	
Casos	Resultado	Observación	Estado
Se selecciona la opción para generar un reporte en PDF sin aplicar ningún filtrado de datos (Pantalla de administración de datos sobre las atenciones)	El archivo PDF se descarga automáticamente y los datos corresponden a los que se encuentran en la tabla	Este caso se comprobó en todas las pantallas de administración de datos, concluyendo en el mismo resultado	Aprobado
Se selecciona la opción para generar un reporte en PDF aplicando un filtrado de datos (Pantalla de administración de datos sobre las atenciones)	El archivo PDF se descarga automáticamente y los datos corresponden a los que se encuentran en la tabla	Este caso se comprobó en todas las pantallas de administración de datos, concluyendo en el mismo resultado	Aprobado

Tabla 40. Prueba de aplicación – Requerimiento R014

ID		Requerimiento	
R014		Cerrar sesión	
Casos	Resultado	Observación	Estado
Se selecciona la opción para cerrar sesión	Los datos de sesión se eliminan y se regresa a la pantalla principal de la aplicación		Aprobado
Se selecciona la opción para cerrar sesión y se intenta acceder a alguna de las pantallas que requieren tener una sesión iniciada	Se redirige a la página principal de la aplicación		Aprobado

b. Pruebas de aceptación

A través de las pruebas de aceptación realizadas se validó la funcionalidad y el cumplimiento de los requerimientos indispensables de la aplicación establecidos en la Fase I: Recopilación de requerimientos. Las pruebas fueron organizadas en iteraciones con criterios establecidos para ser evaluados.

- **Iteración 1**

Tabla 41. Pruebas de aceptación – Iteración 1

Pruebas de aceptación			
ID	Criterio	Estado	Responsable
1	Verificar que la lista de sucursales se cargue correctamente.	Aceptado	Usuario
2	Comprobar la selección de una sucursal y el control para redireccionar solo si una sucursal está seleccionada.	Aceptado	Usuario
3	Verificar que la lista de tipos de consulta se cargue correctamente.	Aceptado	Usuario
4	Comprobar que el turno se genere para el tipo de consulta seleccionado.	Aceptado	Usuario
5	Comprobar el número de turno generado según la sucursal en la que está funcionando la aplicación.	Aceptado	Usuario

ID	Criterio	Estado	Responsable
6	Verificar el almacenamiento del turno en la base de datos	Aceptado	Programador
7	Verificar el almacenamiento o actualización del contador de turnos en la base de datos.	Aceptado	Programador

- **Iteración 2**

Tabla 42. Pruebas de aceptación – Iteración 2

Pruebas de aceptación			
ID	Criterio	Estado	Responsable
1	Comprobar que la lista de ventanillas se cargue correctamente según la sucursal.	Aceptado	Usuario
2	Si la autenticación de usuario es exitosa, verificar el rol del usuario para controlar la selección de la ventanilla.	Aceptado	Programador
3	Comprobar el control para redireccionar solo si se ha seleccionado una ventanilla en caso de ser un usuario trabajador.	Aceptado	Usuario
4	Validar el usuario y el rol dentro de la pantalla del trabajador para mostrar el contenido.	Aceptado	Programador
5	Comprobar que la lista de turnos mostrados corresponde a la sucursal correspondiente.	Aceptado	Usuario

- **Iteración 3**

Tabla 43. Pruebas de aceptación – Iteración 3

Pruebas de aceptación			
ID	Criterio	Estado	Responsable
1	Comprobar que la ventana para atender el turno se abre correctamente.	Aceptado	Usuario
2	Verificar que los datos del turno seleccionado se cargan correctamente en la ventana.	Aceptado	Usuario
3	Verificar el almacenamiento de la atención en la base de datos.	Aceptado	Programador
4	Confirmar que el turno atendido ya no aparece en la lista en espera de atención.	Aceptado	Usuario

ID	Criterio	Estado	Responsable
5	Validar el usuario y el rol dentro de la pantalla del administrador para mostrar el contenido.	Aceptado	Programador
6	Verificar que la redirección a las diferentes páginas funciona correctamente.	Aceptado	Usuario
7	Comprobar que la lista de datos se carga correctamente correspondiendo a la página seleccionada.	Aceptado	Usuario
8	Verificar que los datos se actualizan correctamente.	Aceptado	Usuario
9	Verificar que los datos se desactivan correctamente.	Aceptado	Usuario
10	Verificar que los datos se crean correctamente.	Aceptado	Usuario
11	Comprobar que los datos se guardan o actualizan en la base de datos.	Aceptado	Programador

- **Iteración 4**

Tabla 44. Pruebas de aceptación – Iteración 4

Pruebas de aceptación			
ID	Criterio	Estado	Responsable
1	Comprobar que la lista de turnos en espera de atención se cargue y corresponda a la sucursal correcta dentro de la pantalla de información.	Aceptado	Usuario
2	Comprobar que la lista de turnos que se están atendiendo se cargue y corresponda a la sucursal correcta dentro de la pantalla de información.	Aceptado	Usuario
3	Verificar que el vídeo se reproduce correctamente dentro de la pantalla de información.	Aceptado	Usuario
4	Confirmar que para ingresar a la pantalla de calificación se debe seleccionar una ventanilla.	Aceptado	Usuario
5	Comprobar que las preguntas corresponden a las que fueron proporcionadas por la empresa.	Aceptado	Usuario
6	Verificar que la calificación se guarda en la base de datos.	Aceptado	Programador
7	Comprobar que hay filtros funcionales para el filtrado de datos dentro de las pantallas para gestionar los datos.	Aceptado	Usuario

- **Iteración 5**

Tabla 45. Pruebas de aceptación – Iteración 5

Pruebas de aceptación			
ID	Criterio	Estado	Responsable
1	Verificar que se encuentren los botones para exportar los datos a un archivo Excel y PDF dentro de las pantallas para gestionar los datos.	Aceptado	Usuario
2	Comprobar que los datos exportados dentro de los archivos corresponden con los datos que se encuentran en las pantallas para gestionar los datos.	Aceptado	Usuario
3	Confirmar que los datos se exportan de acuerdo a los filtros seleccionados.	Aceptado	Usuario
4	Comprobar que al cerrar sesión se debe iniciar de nuevo para poder acceder a las pantallas en las que se requiera estar con sesión iniciada.	Aceptado	Usuario
5	Verificar que los datos de la sesión se eliminan correctamente al cerrar sesión.	Aceptado	Programador

c. Pruebas de usabilidad

Para llevar a cabo las pruebas de usabilidad se utilizaron métricas, para evaluar distintos aspectos desde el punto de vista del usuario:

Tabla 46. Pruebas de usabilidad

Pruebas de usabilidad			
ID	Métrica	Criterio	Estado
1	Texto legible	El usuario puede leer el texto de manera clara y comprendiendo el contexto de este.	Aceptado
2	Consistencia visual	Los elementos de la aplicación se usan de manera coherente (botones, colores, imágenes)	Aceptado
3	Manejo de alertas	La aplicación brinda información dentro de alertas de manera visual en caso de que no funcione o no se maneje correctamente.	Aceptado
4	Navegación	La aplicación permite navegar de manera sencilla a través de las distintas pantallas.	Aceptado

ID	Métrica	Criterio	Estado
5	Experiencia	La aplicación brinda al usuario una experiencia cómoda al utilizarla.	Aceptado

3.2.5 Fase V: Implementación

Para la implementación se elaboró un cronograma de actividades para detallar el tiempo empleado para el desarrollo de la última etapa de la metodología RAD:

Tabla 47. Cronograma de implementación

N°	Actividad	Días						
		1	2	3	4	5	6	7
1	Preparación de datos.	■	■					
2	Verificación de infraestructura.	■						
3	Configuración e importación de la base de datos.	■						
4	Configuración y subida de servicios.	■	■					
5	Despliegue de servicios.	■						
6	Preparación del paquete de aplicación.	■						
7	Configuración y subida de aplicación.	■	■					
8	Despliegue de aplicación.	■	■	■				
9	Pruebas de ejecución.	■	■	■	■			
10	Entrega del producto.	■	■	■	■	■		

Preparación de datos: A través del trabajo en conjunto con el ingeniero a cargo, se obtuvo la información sobre los datos que se van a almacenar y se realizó una base de datos acorde a la información proporcionada.

Verificación de infraestructura: Se realizó una revisión dentro de la sucursal “La Matriz” para verificar que se encuentre la infraestructura adecuada (equipos de cómputo, pantalla para mostrar información, equipo para imprimir tickets) para que la aplicación funcione correctamente.

Configuración e importación de la base de datos: Se configuró el servidor SQL Server en el que se alojó la base de datos que una vez comprobada de manera local se procedió a subirla para comprobar su funcionamiento en un entorno de producción.

Configuración y subida de servicios: Se configuró el servidor IIS en el que se alojaron los servicios desarrollados para comprobar su funcionamiento en producción.

Despliegue de servicios: Se comprobó que los servicios funcionaban correctamente una vez subidos al servidor IIS.

Preparación del paquete de aplicación: Se construyó un paquete de la aplicación para subirlo al servidor IIS a través del comando:

```
PS E:\Projects\sistema_turnos> npm run build
```

Figura 85. Comando de construcción de paquete de aplicación

Configuración y subida de aplicación: Se configuró una página dentro del servidor IIS para alojar la aplicación. Posteriormente se subieron los archivos creados dentro del paquete de la aplicación.

Despliegue de aplicación: Se comprobó que la aplicación funcionaba correctamente una vez subida al servidor IIS.

Pruebas de ejecución: Etapa que hace referencia al apartado de Resultados, en las que se describen las pruebas en el despliegue de la aplicación.

Entrega del producto: Se entregó el producto final, adjuntando la documentación para el usuario en el Anexo F. La evidencia de la entrega de la aplicación se encuentra en la Figura 86, Figura 87, Figura 88 y Figura 89

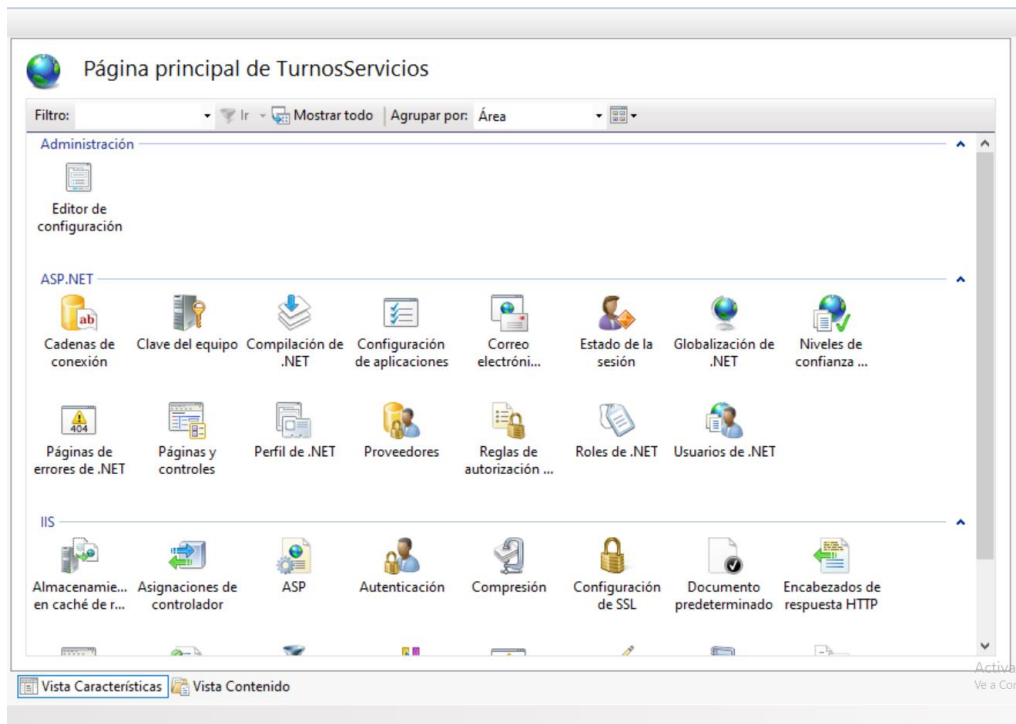


Figura 86. Página IIS Servicios

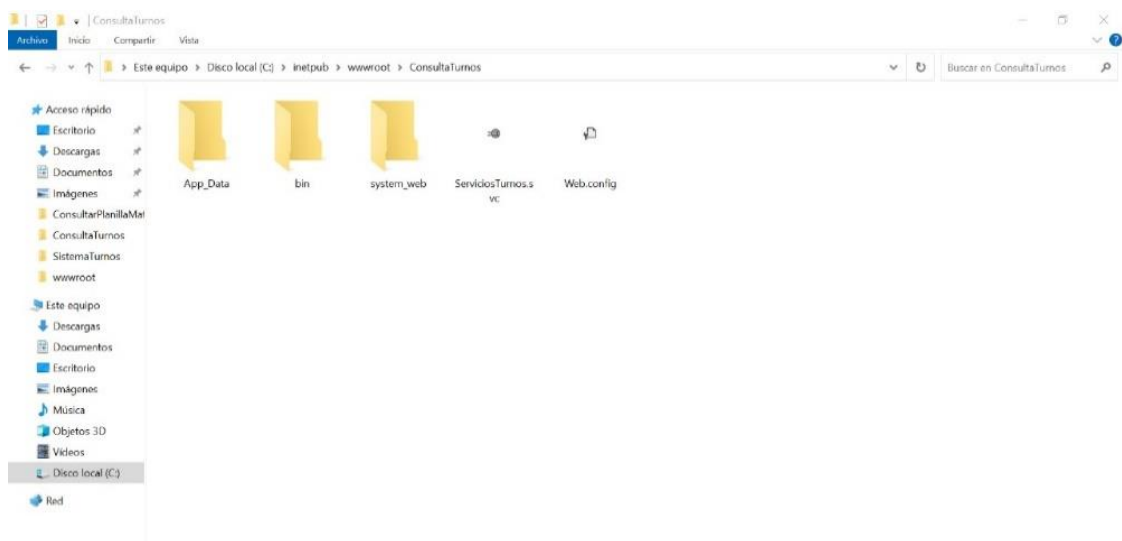


Figura 87. Archivos Servicios

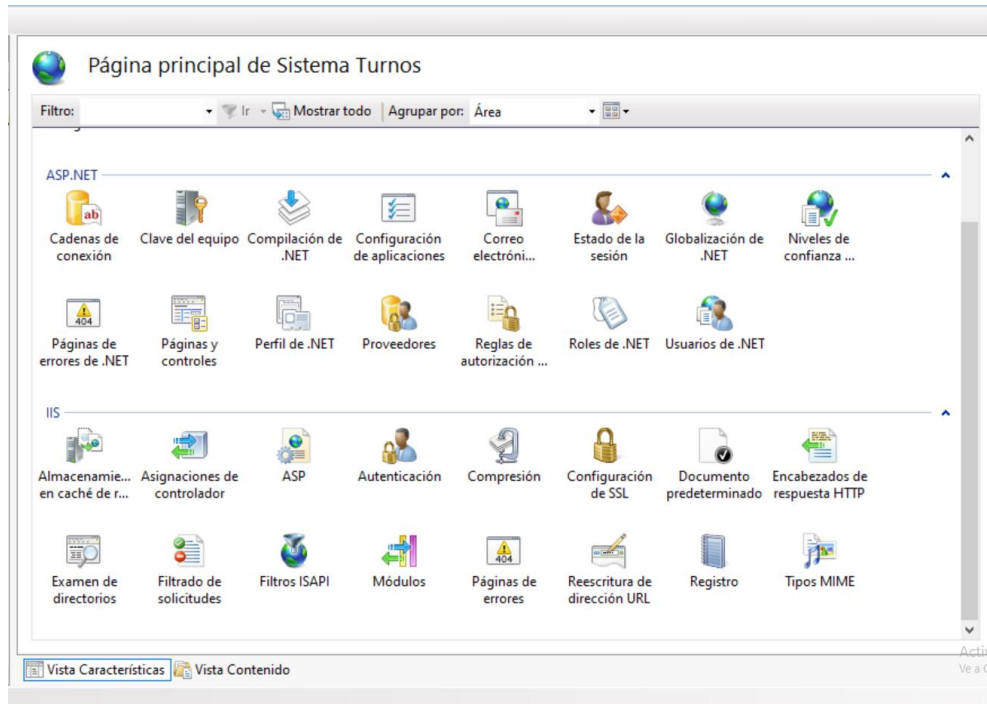


Figura 88. Página IIS Aplicación

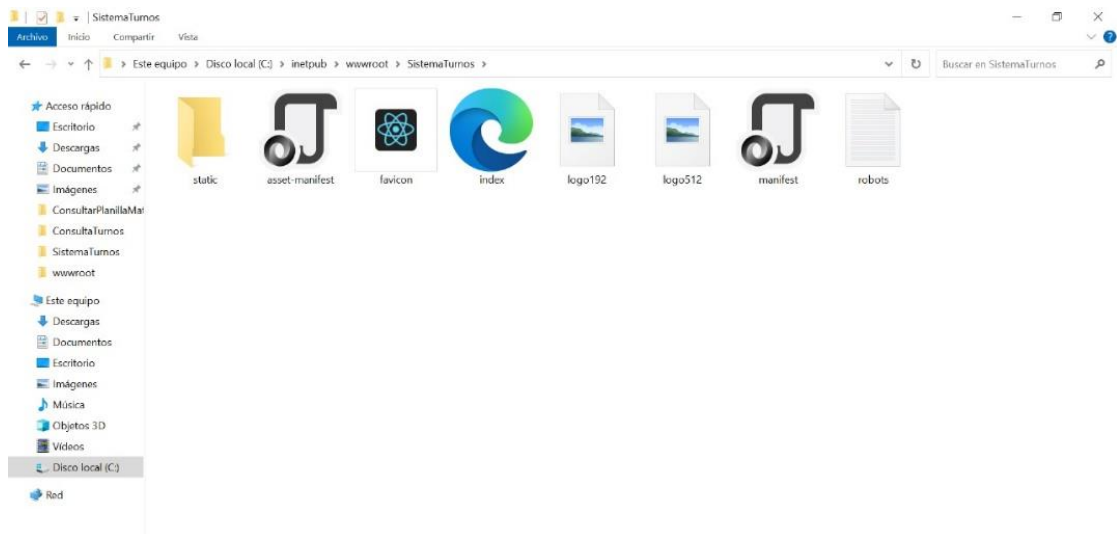


Figura 89. Archivos Aplicación

3.3 Resultados

Para este proyecto se utilizaron los modelos TAM y UTAUT para evaluar de manera integral la aceptación y utilización de la aplicación. Los modelos se escogieron debido a su complementariedad, mientras que TAM se enfoca en las percepciones

individuales, UTAUT expande la perspectiva a las condiciones sociales y experiencia previa.

Para los cuestionarios se utilizó una escala de medida Likert donde:

- 1: Muy en desacuerdo
- 2: En desacuerdo
- 3: Neutral
- 4: De acuerdo
- 5: Muy de acuerdo

Los cuestionarios fueron aplicados a los empleados del balcón de servicios de la empresa.

3.3.1 Modelo TAM

El modelo TAM (Technology Acceptance Model) es una teoría que busca comprender la aceptación de una tecnología por parte de los usuarios [38]. En este proyecto, el modelo permite evaluar distintos aspectos desde el punto de vista del usuario como la utilidad y facilidad de uso, además de la intención y actitud de uso del usuario con la aplicación.

a. Cuestionario del modelo TAM

Tabla 48. Cuestionario de evaluación del modelo TAM

Cuestionario de evaluación	
PU	Percepción de utilidad
PU1	La aplicación es eficaz en comparación con el método anterior para gestionar y atender turno

PU2	La aplicación mejora la eficiencia e interacción del usuario
PU3	La aplicación ofrece beneficios en términos de reducción de tiempos de espera y atención de turnos
PU4	La aplicación mejora la disponibilidad y accesibilidad de los servicios de la empresa

PEU	Percepción de facilidad de uso
------------	---------------------------------------

PEU1	La interfaz de la aplicación proporciona simplicidad y facilidad de navegación
PEU2	La aplicación es intuitiva y fácil de manejar incluso para aquellos que no tienen experiencia técnica
PEU3	La gestión de los datos es sencilla y comprensible para los usuarios
PEU4	El proceso de generación y atención de turnos mediante la aplicación es fácil y comprensible

AU	Actitud hacia el uso
-----------	-----------------------------

AU1	La implementación de la aplicación supone una mejora significativa en general
AU2	Los usuarios perciben positivamente la aplicación y tienen la intención de utilizarla
AU3	La aplicación mejora significativamente la calidad y eficiencia de los servicios de la empresa
AU4	La aplicación supone un paso positivo hacia la modernización y mejora de los servicios

IU	Intención de uso
IU1	Los usuarios tienen la intención de utilizar la aplicación de manera regular en el futuro
IU2	La incorporación de la aplicación influye de manera positiva a la rutina habitual de la gestión y atención de turnos

b. Resultados del cuestionario

- Percepción de utilidad

Tabla 49. Tabla de frecuencias – Modelo TAM – Percepción de utilidad

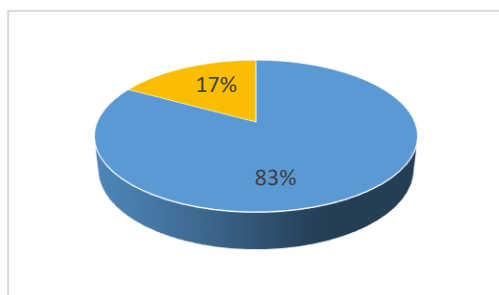
PU	Percepción de utilidad				
	1	2	3	4	5
PU1	0%	0%	0%	83%	17%
PU2	0%	0%	8%	92%	0%
PU3	0%	0%	33%	67%	0%
PU4	0%	0%	17%	58%	25%
MEDIA	0%	0%	14,5%	75%	10,5%

Tabla 50. Resultados modelo TAM – Percepción de utilidad

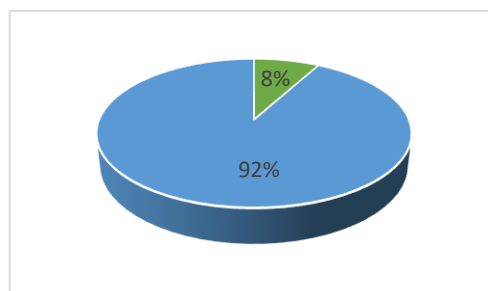
Percepción de utilidad

■ Neutral ■ De acuerdo ■ Muy de acuerdo

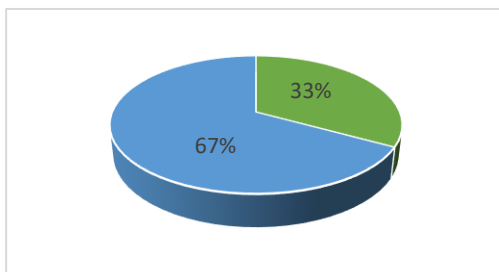
PU1



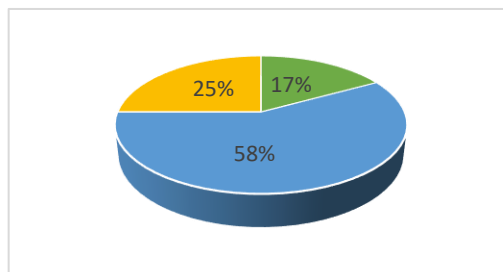
PU2



PU3



PU4



A través de los resultados representados en la Tabla 50, se concluye que los usuarios coinciden en que la aplicación es un método eficaz en comparación con métodos anteriores, además de que mejora la eficiencia e interacción del usuario, ofreciendo beneficios en la reducción de tiempos de espera y atención, mejorando la disponibilidad y accesibilidad de los servicios de la empresa.

- Percepción de facilidad de uso

Tabla 51. Tabla de frecuencias – Modelo TAM – Facilidad de uso

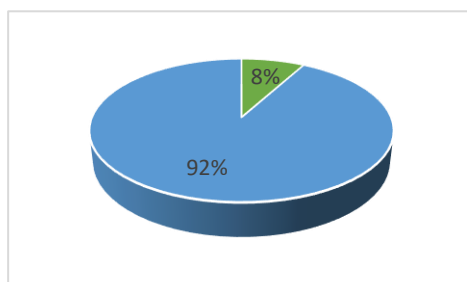
Percepción de facilidad de uso					
PEU	1	2	3	4	5
PEU1	0%	0%	8%	92%	0%
PEU2	0%	0%	33%	67%	0%
PEU3	0%	0%	8%	92%	0%
PEU4	0%	0%	0%	58%	42%
MEDIA	0%	0%	12,25%	77,25%	10,5%

Tabla 52. Resultados modelo TAM – Facilidad de uso

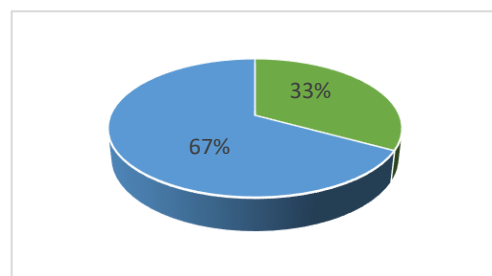
Percepción de facilidad de uso		
---------------------------------------	--	--

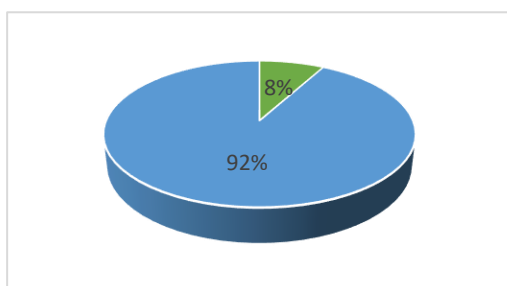
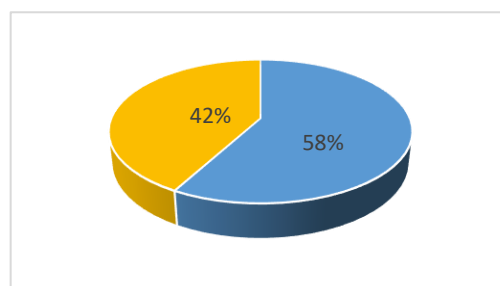
■ Neutral ■ De acuerdo ■ Muy de acuerdo

PEU1



PEU2



PEU3**PEU4**

Mediante los resultados representados en la Tabla 52, se concluye que la aplicación es intuitiva, fácil de manejar y comprensible para los usuarios, proporcionando facilidad en la navegación y gestión de los datos.

- Actitud hacia el uso

Tabla 53. Tabla de frecuencias – Modelo TAM – Actitud hacia el uso

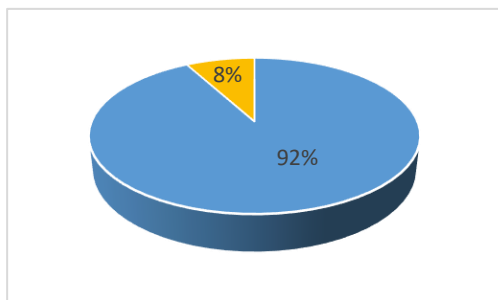
Actitud hacia el uso					
AU	1	2	3	4	5
AU1	0%	0%	0%	92%	8%
AU2	0%	0%	8%	50%	42%
AU3	0%	0%	17%	58%	25%
AU4	0%	0%	0%	92%	8%
MEDIA	0%	0%	6,25%	73%	20,75%

Tabla 54. Resultados modelo TAM – Actitud hacia el uso

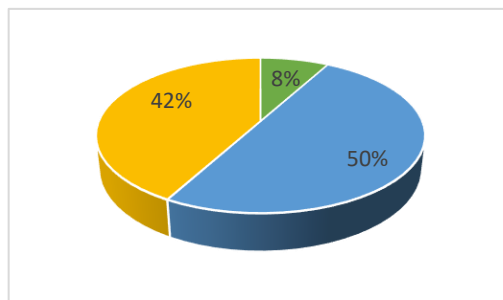
Actitud hacia el uso

■ Neutral ■ De acuerdo ■ Muy de acuerdo

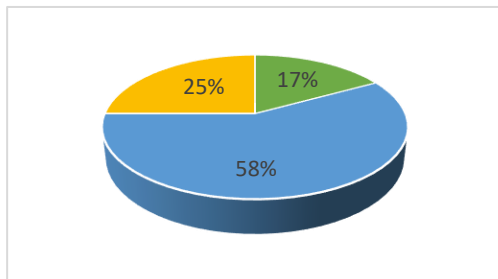
AU1



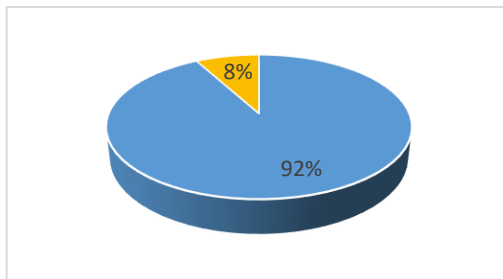
AU2



AU3



AU4



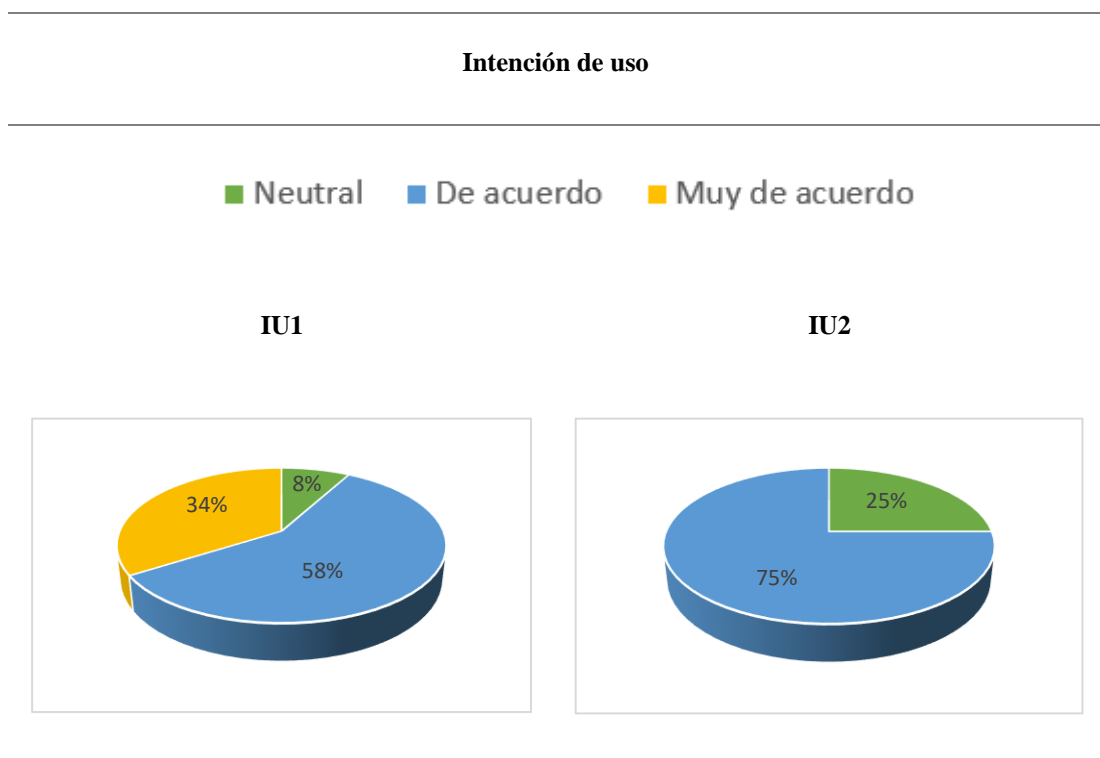
Como se observa en los resultados representados en la Tabla 54, los usuarios muestran una actitud positiva con respecto al uso de la aplicación, opinando que proporciona una mejora significativa en la calidad y eficiencia de los servicios, suponiendo un paso positivo hacia la modernización y mejora de los mismos.

- Intención de uso

Tabla 55. Tabla de frecuencias – Modelo TAM – Intención de uso

Intención de uso					
IU	1	2	3	4	5
IU1	0%	0%	8%	58%	34%
IU2	0%	0%	25%	75%	0%
MEDIA	0%	0%	16,5%	67%	17,5%

Tabla 56. Resultados modelo TAM – Intención de uso



Mediante los resultados representados en la Tabla 56, se concluye que los usuarios en general tienen la intención de utilizar la aplicación de manera regular, opinando que esta influye de manera positiva a la rutina de gestión y atención de los turnos.

3.3.2 Modelo UTAUT

El modelo UTAUT (Unified Theory of Acceptance of Technology) es una extensión del modelo TAM, que combina varios modelos para proporcionar una comprensión completa de la aceptación de la tecnología [38]. En este proyecto, el modelo contribuye a comprender otros factores como la expectativa de rendimiento y esfuerzo que se espera el usuario sobre la aplicación, además de las condiciones sociales y la experiencia previa del usuario.

a. Cuestionario del modelo UTAUT

Tabla 57. Cuestionario de evaluación del modelo UTAUT

Cuestionario de evaluación	
PR	Expectativa de rendimiento
PR1	La aplicación proporcionará una mayor accesibilidad y rapidez al proceso
PR2	La aplicación simplificará el proceso de solicitud y asignación de turnos
PE	Esfuerzo percibido
PE1	El proceso de aprendizaje y uso de la aplicación no es complejo
PE2	El aprendizaje para el uso de la aplicación no conllevará un esfuerzo significativo
PE3	La adaptación a la aplicación será sencilla y sin complicaciones

SC	Condiciones sociales
SC1	La opinión de los demás usuarios no influye en la disposición de un usuario en particular para usar la aplicación
SC2	La empresa ha notificado de manera efectiva los beneficios y objetivos de la aplicación
SC3	Los usuarios están dispuestos a adoptar la aplicación
EP	Experiencia previa
EP1	Los usuarios tienen experiencia previa con respecto al uso de otro sistema similar
EP2	Los conocimientos tecnológicos de los usuarios son adecuados para facilitar la adopción de la aplicación

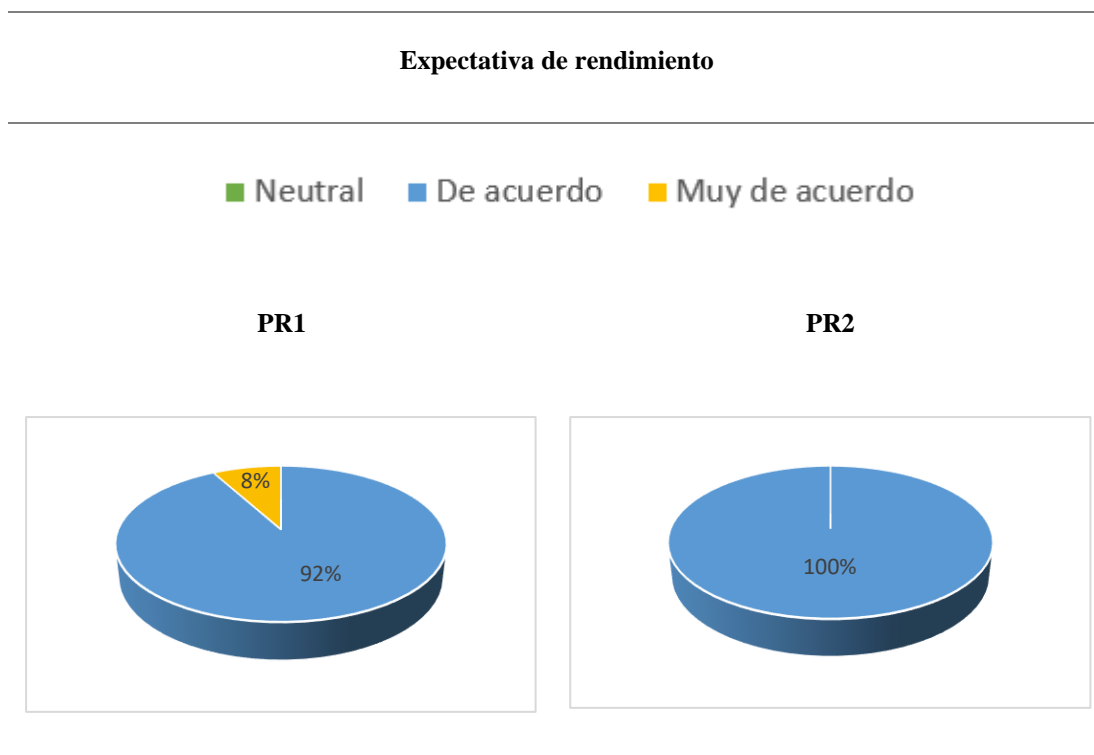
b. Resultados del cuestionario

- Expectativa de rendimiento

Tabla 58. Tabla de frecuencias – Modelo UTAUT – Expectativa de rendimiento

Expectativa de rendimiento					
PR	1	2	3	4	5
PR1	0%	0%	0%	92%	8%
PR2	0%	0%	0%	100%	0%
MEDIA	0%	0%	0%	96%	4%

Tabla 59. Resultados modelo UTAUT – Expectativa de rendimiento



Como se observa en los resultados representados en la Tabla 59, la mayoría de los usuarios opina que la aplicación proporciona accesibilidad y rapidez al proceso, además de que todos coinciden en que simplificará todo el proceso.

- Esfuerzo percibido

Tabla 60. Tabla de frecuencias – Modelo UTAUT – Esfuerzo percibido

Esfuerzo percibido					
PE	1	2	3	4	5
PE1	0%	0%	8%	92%	0%
PE2	0%	0%	17%	83%	0%

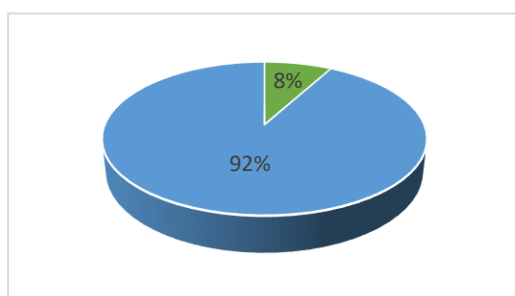
PE3	0%	0%	8%	92%	0%
MEDIA	0%	0%	11%	89%	0%

Tabla 61. Resultados modelo UTAUT – Esfuerzo percibido

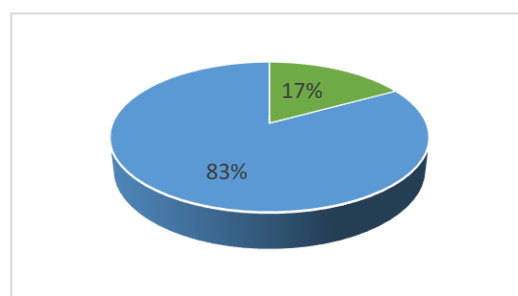
Esfuerzo percibido

■ Neutral ■ De acuerdo ■ Muy de acuerdo

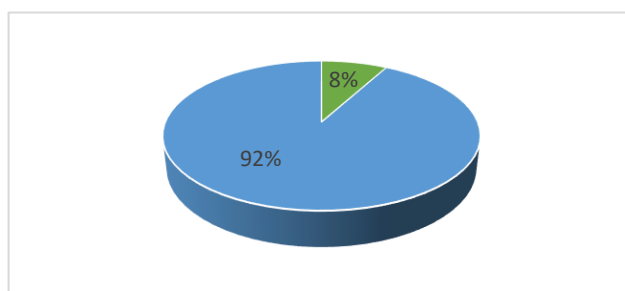
PE1



PE2



PE3



Mediante los resultados representados en la Tabla 61, se concluye que la mayoría de los usuarios coinciden en que el proceso de aprendizaje del uso de la aplicación no es

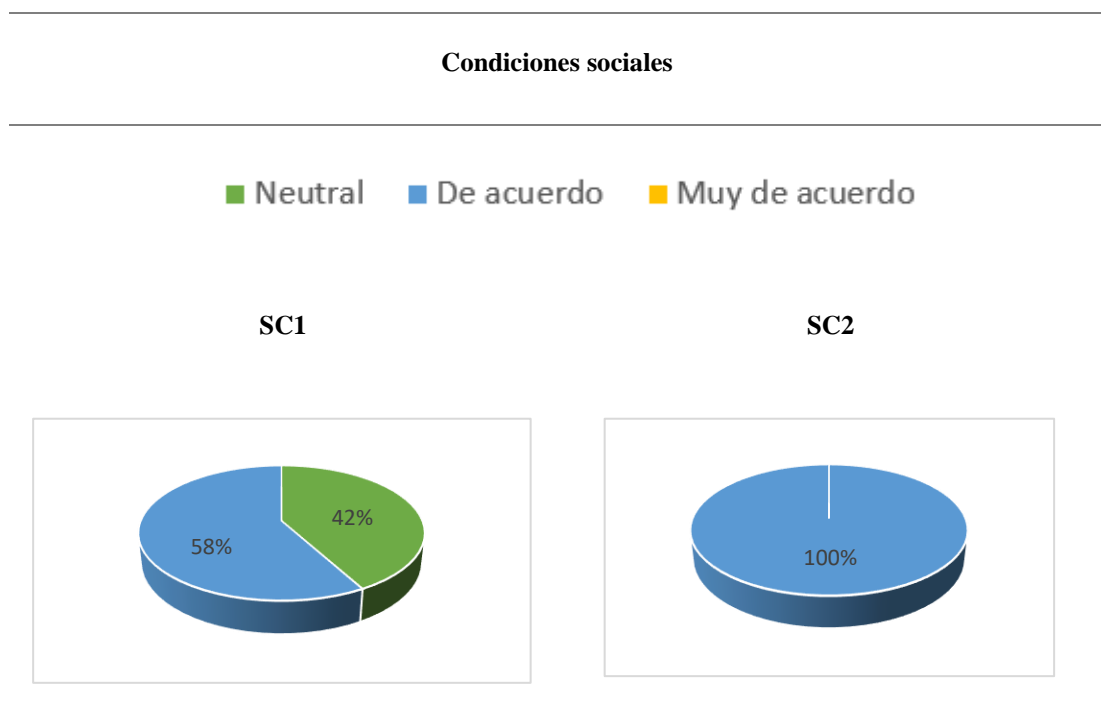
complejo, por lo que no conllevará un esfuerzo significativo y la adaptación a la aplicación será sencilla.

- Condiciones sociales

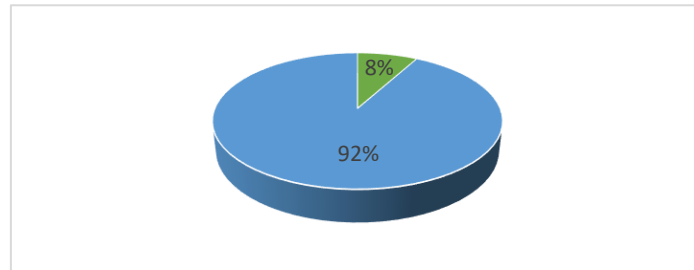
Tabla 62. Tabla de frecuencias – Modelo UTAUT – Condiciones sociales

Condiciones sociales					
SC	1	2	3	4	5
SC1	0%	0%	42%	58%	0%
SC2	0%	0%	0%	100%	0%
SC3	0%	0%	8%	92%	0%
MEDIA	0%	0%	16,67	83,33%	0%

Tabla 63. Resultados modelo UTAUT – Condiciones sociales



SC3



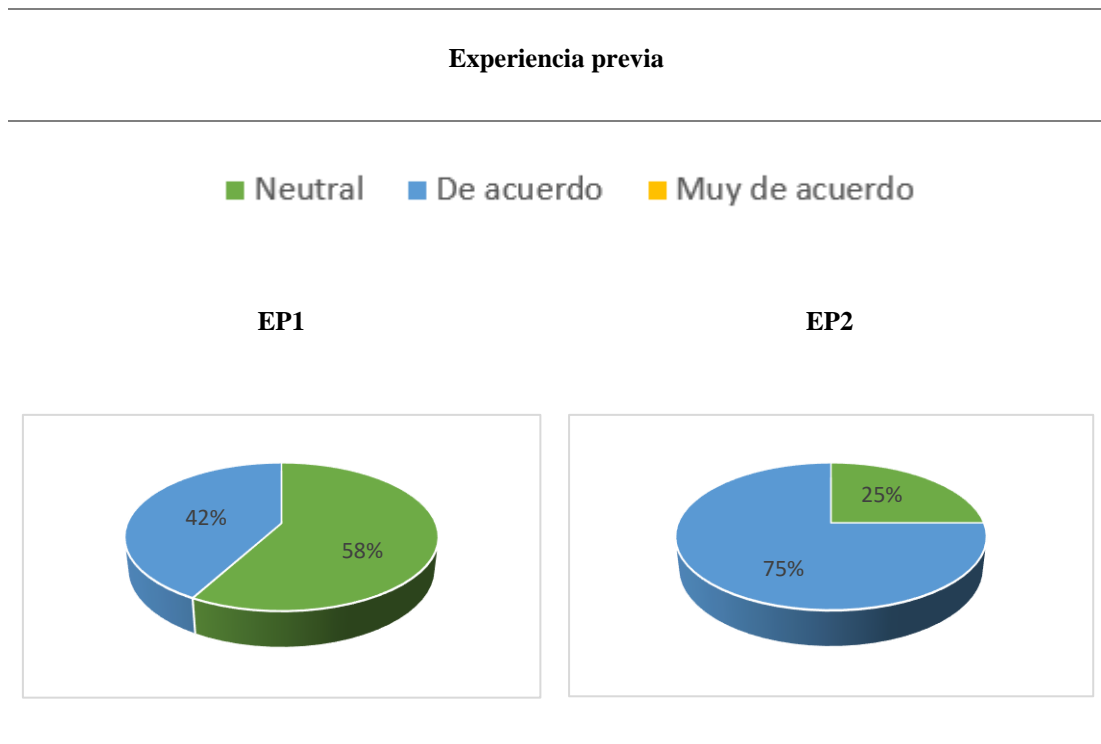
Como se observa en los resultados representados en la Tabla 63, todos los usuarios coinciden en que los beneficios y el objetivo de la aplicación fueron notificados por la empresa, además de que se concluye que un 92% de los usuarios están dispuestos a utilizarla, añadiendo que en general no se guían por la opinión de los demás usuarios.

- Experiencia previa

Tabla 64. Tabla de frecuencias – Modelo UTAUT – Experiencia previa

Experiencia previa					
EP	1	2	3	4	5
EP1	0%	0%	58%	42%	0%
EP2	0%	0%	25%	75%	0%
MEDIA	0%	0%	41,5%	58,5%	0%

Tabla 65. Resultados modelo UTAUT – Experiencia previa



Como se observa a través de los resultados representados en la Tabla 65, solo un 42% de los usuarios consideran que tienen experiencia previa con otro sistema similar, pero la gran mayoría opina que tiene los conocimientos tecnológicos necesarios para utilizar la aplicación correctamente.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

En esta investigación se estableció como objetivo la implantación de una aplicación web para la centralización de la gestión y atención de los turnos dentro de las sucursales de la EP-EMAPA-A, concluyendo lo siguiente:

- A través de la aplicación de encuestas, se consiguió analizar tanto el proceso como la satisfacción y comodidad referente a la gestión y atención de turnos dentro de las sucursales de la EP-EMAPA-A, desde el punto de vista del personal y de los clientes.
- El análisis permitió revelar limitaciones en el proceso, evidenciando áreas de mejora como la carencia de automatización, la ineficacia en el manejo de los datos sobre los turnos y las atenciones, además de un cierto nivel de insatisfacción por parte de los clientes. La identificación de estas deficiencias destacó la necesidad de implementar una solución que permitiera optimizar la eficiencia del proceso.
- Mediante una entrevista a los representantes de la empresa se determinaron las necesidades específicas de la aplicación, así como las herramientas y tecnologías que se utilizaron para el desarrollo de este proyecto. En este contexto, se concluyó que, para mantener la escalabilidad y el mantenimiento por parte de la empresa, se utilizarían para la parte lógica de la aplicación herramientas ya implementadas dentro del entorno tecnológico de la empresa (.NET y SQL Server).
- El desarrollo e implantación de la aplicación web para la gestión y atención de turnos beneficia al personal de la empresa y a los clientes que asisten a sacar turnos para ser atendidos, ayudando en la automatización y permitiendo tener

una visión completa del proceso para que la empresa pueda tomar medidas de mejora.

- Utilizar herramientas como .NET para el desarrollo de la lógica de la aplicación permitió la creación de los servicios que se utilizaron en la aplicación de una manera eficiente, al ser una plataforma que agiliza el proceso de desarrollo y garantiza la solidez y flexibilidad. Dado que es una tecnología previamente adoptada dentro de la empresa, se contó con un servidor dedicado a alojar este tipo de servicios, lo que simplificó el proceso de configuración y despliegue.
- La base de datos se diseñó utilizando SQL Server, donde se creó un esquema incluyendo las tablas que almacenaron los datos manejados dentro de la aplicación, lo que proporcionó una estructura organizada y aseguró la integridad y confiabilidad de los datos almacenados. Además, la plataforma facilitó la implementación de procedimientos almacenados necesarios para el correcto funcionamiento de la aplicación. Al tratarse de una tecnología ya implementada en la empresa, se dispuso de un servidor SQL Server destinado a hospedar bases de datos, lo que facilitó significativamente el proceso.
- La aplicación de la metodología RAD permitió obtener una retroalimentación continua con el cliente a través de las entregas periódicas para evaluar las funciones principales de la aplicación. Este enfoque facilitó la adaptación a las necesidades del cliente para efectuar los cambios que surgieron durante el desarrollo del proyecto y cumplir con los requerimientos establecidos en un menor tiempo.
- El valor principal que aporta la aplicación se centra en la empresa, proporcionando una herramienta que permite al usuario gestionar eficientemente los datos generados, lo que brinda una visión integral del proceso en todas las sucursales y además posibilita un análisis detallado a nivel individual del proceso en cada sucursal. La capacidad de tener una visión

global del proceso en todas las sucursales permite a la empresa tener una perspectiva unificada y centralizada, factor crucial para la toma de decisiones informadas.

- El uso de la metodología RAD en combinación con las herramientas utilizadas para el desarrollo como .NET para el desarrollo de los servicios, SQL Server como gestor de base de datos y React JS para la aplicación web proporcionaron un entorno efectivo que contribuyó a la finalización exitosa de todas las etapas del desarrollo concluyendo en un producto final que satisface las necesidades del cliente.

4.2 Recomendaciones

- Se recomienda que la EP-EMAPA-A, realice un plan de mejora continua para la aplicación, destacando la importancia de mantenerse al tanto de los avances tecnológicos y las necesidades del usuario.
- Para verificar el nivel de satisfacción del personal y de los clientes a lo largo del tiempo se recomienda establecer un canal de comunicación con los usuarios para obtener retroalimentación sobre posibles mejoras y problemas de la aplicación.
- Para mantener el correcto funcionamiento de la aplicación a largo plazo, se considera vital realizar un plan de mantenimiento en el que se revise el rendimiento, adaptabilidad y escalabilidad de la herramienta.
- Se recomienda mantener actualizadas las bibliotecas y frameworks utilizados para el desarrollo de este proyecto con la finalidad de aprovechar nuevas características y mejoras de seguridad para asegurar la estabilidad de la aplicación.
- Se recomienda considerar el presente proyecto de investigación para su adaptación a otras entidades dentro de la EP-EMAPA que aún carezcan de un

método efectivo para la gestión y atención de turnos, y en las que se esté contemplando la automatización del proceso.

- Se propone establecer un plan de capacitación continua al personal encargado de la gestión y atención de los turnos para garantizar el uso efectivo y eficiente de la aplicación.
- Se sugiere la implementación de un sistema de soporte de decisiones (DSS) en futuras investigaciones, con el objetivo de proporcionar a la empresa una fuente de información relevante para la toma de decisiones efectivas.

REFERENCIAS BIBLIOGRÁFICAS

- [1] D. J. Quiroga-Parra, J. Torrent-Sellens, y C. P. Murcia-Zorrilla, «Las tecnologías de la información en América Latina, su incidencia en la productividad: Un análisis comparado con países desarrollados», *DYNA*, vol. 84, n.º 200, pp. 281-290, ene. 2017, doi: 10.15446/dyna.v84n200.60632.
- [2] C. V. Buenaño Pesántez, C. A. Tenesaca Mendoza, X. J. Zúñiga García, y L. Marqués Molías, «Inversión de las empresas ecuatorianas en las TICS durante el siglo XXI y ante la pandemia Covid-19», *Polo del Conocimiento: Revista científico - profesional*, vol. 6, n.º 9, pp. 203-223, 2021.
- [3] «03_Gobierno_Electrnico_Indicencia_de_las_TIC_instituciones_pblicas.pdf». Accedido: 25 de julio de 2023. [En línea]. Disponible en: https://observatorioecuadordigital.mintel.gob.ec/wp-content/uploads/2022/11/03_Gobierno_Electrnico_Indicencia_de_las_TIC_instituciones_pblicas.pdf
- [4] R. F. Cespedes Hurtado, «SISTEMA DE INFORMACIÓN PARA MEJORAR LA GESTIÓN DE TURNOS EN EL ÁREA DE OPERACIONES DE LATINA TV – LIMA, 2020.», *Universidad Privada Telesup - UTELESUP*, 2020, Accedido: 26 de junio de 2023. [En línea]. Disponible en: <https://repositorio.utelesup.edu.pe/handle/UTELESUP/1345>
- [5] J. E. Molina Segovia, «Desarrollo e implantación del sistema de gestión de turnos para consulta externa del Hospital de Especialidades Eugenio Espejo», bachelorThesis, QUITO/ EPN/ 2008, 2008. Accedido: 24 de junio de 2023. [En línea]. Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/924>
- [6] J. A. Bazán Perero, «Desarrollo de aplicación web para gestión de turnos en la escuela de educación básica particular Monseñor Roberto María Del Pozo», bachelorThesis, 2023. Accedido: 23 de junio de 2023. [En línea]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/24734>
- [7] Chávez, Bravo y Adelina, Ana, «Mejoras al sistema de gestión de turnos en una empresa de telecomunicaciones». Repositorio institucional Cybertesis UNMSM, 2018.

- [8] E. C. R. Ramos Rosales, J. A. Verduzco Ramírez, N. García Díaz, y S. Arceo Díaz, «Diseño e implementación de un sistema integral de control de turnos», *3c Empresa: investigación y pensamiento crítico*, vol. 8, n.º 1, pp. 92-111, 2019.
- [9] «EF_News_2022_EN_Screen.pdf». Accedido: 26 de junio de 2023. [En línea]. Disponible en: https://www.eurofunk.com/wp-content/uploads/EF_News_2022_EN_Screen.pdf
- [10] S. Ped, «SISTEMA DE GESTION», Accedido: 25 de junio de 2023. [En línea]. Disponible en: https://www.academia.edu/24415630/SISTEMA_DE_GESTION
- [11] Y. Chávez Montejó y H. Pérez Sousa, «Gestión documental, Gestión de información y Gestión del conocimiento: nociones e interrelaciones.», *Bibliotecas. Anales de Investigación*, n.º 8-9, pp. 222-227, 2013.
- [12] A. Benvenuto Vera, «Implementación de sistemas ERP, su impacto en la gestión de la empresa e integración con otras TIC», *CAPIC REVIEW*, n.º 4, p. 3, 2006.
- [13] C. E. Granados Molina, «CRM (Customer Relationship Management) y servicios de información», *Biblios: Revista electrónica de bibliotecología, archivología y museología*, n.º 10, p. 1, 2001.
- [14] G. Martínez Villalobos, G. D. Camacho Sánchez, y D. A. Biancha Gutiérrez, «Diseño de framework web para el desarrollo dinámico de Aplicaciones», *Scientia et Technica*, vol. 1, n.º 44, pp. 178-183, 2010.
- [15] «Angular - What is Angular?» Accedido: 23 de junio de 2023. [En línea]. Disponible en: <https://angular.io/guide/what-is-angular>
- [16] «Describing the UI – React». Accedido: 24 de junio de 2023. [En línea]. Disponible en: <https://react.dev/learn/describing-the-ui>
- [17] «Installation - Laravel - The PHP Framework For Web Artisans». Accedido: 24 de junio de 2023. [En línea]. Disponible en: <https://laravel.com/docs/10.x>
- [18] L. R. Pacherras Muñoz, «Páginas web Introducción, conceptos, creación de sitios Web, diseño de páginas WEB, herramientas para generar páginas WEB, tipos, principales funciones, aplicaciones», dic. 2018, Accedido: 26 de junio de 2023. [En línea]. Disponible en: <http://repositorio.une.edu.pe/handle/20.500.14039/5006>
- [19] «913.pdf». Accedido: 26 de junio de 2023. [En línea]. Disponible en: <https://www.uoc.edu/pdf/masters/oficiales/img/913.pdf>

- [20] N. Palma Pérez, «Solución informática para la selección del servidor web durante la migración a código abierto», *Revista Cubana de Ciencias Informáticas*, vol. 14, n.º 2, pp. 49-69, jun. 2020.
- [21] «metodologias de desarrollo de software.pdf». Accedido: 4 de diciembre de 2023. [En línea]. Disponible en: <https://cdn2.hubspot.net/hubfs/371274/Desarrollo%20de%20Software/metodologias%20de%20desarrollo%20de%20software.pdf>
- [22] «tema_1-ModeloWaterfall.pdf». Accedido: 4 de diciembre de 2023. [En línea]. Disponible en: https://www.dtagency.tech/cursos/metodologias_gestion_proyectos/tema_1-ModeloWaterfall.pdf
- [23] F. J. G. Peñalvo, A. G. Holgado, y A. V. Ingelmo, «INGENIERÍA DE SOFTWARE I».
- [24] E. Gabriel, «Metodologías de desarrollo de software».
- [25] A. Á. Carulla, «1986 “The New New Product Development Game” Harvard Business Review, 64(1), 137-146.», . *Prólogo*, 2021.
- [26] D. A. A. Moya, «EVALUACIÓN ENTRE LAS METODOLOGÍAS DE DESARROLLO DE SOFTWARE XP - RUP Y SU APLICACIÓN EN LA CONSTRUCCIÓN DE UN SISTEMA PARA LA GESTIÓN DE UNA ESTÉTICA».
- [27] «Alfa de cronbach para validar un cuestionario de uso de tic en docentes universitarios», *mktdescubre*, pp. 37-48, dic. 2017, doi: 10.36779/mktdescubre.v10.141.
- [28] «2227-1899-rcci-15-01-37.pdf». Accedido: 4 de diciembre de 2023. [En línea]. Disponible en: <http://scielo.sld.cu/pdf/rcci/v15n1/2227-1899-rcci-15-01-37.pdf>
- [29] P. Nidagundi y L. Novickis, «Introducing Lean Canvas Model Adaptation in the Scrum Software Testing», *Procedia Computer Science*, vol. 104, pp. 97-103, ene. 2017, doi: 10.1016/j.procs.2017.01.078.
- [30] E. Á. Domenech y A. M. Abad, «Delfdroid y su comparación evaluativa con XP y Scrum mediante el método 4-DAT», vol. 7, n.º 1, 2013.
- [31] «Figma: The Collaborative Interface Design Tool», Figma. Accedido: 4 de diciembre de 2023. [En línea]. Disponible en: <https://www.figma.com/>

- [32] «Home v6.20.1». Accedido: 4 de diciembre de 2023. [En línea]. Disponible en: <https://reactrouter.com/en/main>
- [33] «Getting Started | Axios Docs». Accedido: 4 de diciembre de 2023. [En línea]. Disponible en: <https://axios-http.com/docs/intro>
- [34] «react-datepicker», npm. Accedido: 4 de diciembre de 2023. [En línea]. Disponible en: <https://www.npmjs.com/package/react-datepicker>
- [35] «jspdf», npm. Accedido: 4 de diciembre de 2023. [En línea]. Disponible en: <https://www.npmjs.com/package/jspdf>
- [36] «xlsx», npm. Accedido: 4 de diciembre de 2023. [En línea]. Disponible en: <https://www.npmjs.com/package/xlsx>
- [37] «react-paginate», npm. Accedido: 4 de diciembre de 2023. [En línea]. Disponible en: <https://www.npmjs.com/package/react-paginate>
- [38] P. Palos-Sanchez, A. Reyes-Menendez, y J. R. Saura, «Modelos de Adopción de Tecnologías de la Información y Cloud Computing en las Organizaciones», *Inf. tecnol.*, vol. 30, n.º 3, pp. 3-12, jun. 2019, doi: 10.4067/S0718-07642019000300003.

ANEXOS

Anexo A. Cuestionario aplicado al personal de la empresa

Objetivo: Determinar el beneficio y opinión sobre el desarrollo e implementación de una aplicación web para la gestión y atención de turnos.

1. ¿Cómo califica la gestión actual de la información de los turnos?
2. ¿Con qué frecuencia interactúa con los clientes que solicitan turnos para ser atendidos?
3. ¿Qué métodos o sistemas utiliza actualmente para administrar la información de los turnos?
4. ¿Ha tenido dificultades para manejar la información de los turnos?
5. ¿Con qué frecuencia los clientes han expresado alguna queja o comentario sobre el proceso de solicitud de turnos?
6. ¿Se siente satisfecho con el proceso de administración de turnos de la institución?
7. ¿Cree que una aplicación de gestión y atención de turnos puede mejorar el proceso actual?

Anexo B. Cuestionario aplicado a los clientes de la empresa

Objetivo: Determinar la opinión y satisfacción sobre el proceso manual de gestión y atención de turnos.

1. ¿Cómo califica la calidad de la atención?
2. ¿Con qué frecuencia utiliza los servicios de atención al cliente de la entidad?
3. ¿Qué nivel de satisfacción tiene con respecto al tiempo para generar su turno?
4. ¿Qué nivel de satisfacción tiene con respecto al tiempo de espera para recibir atención?
5. ¿Cuánto tiempo espera normalmente hasta que se atiende su turno?
6. ¿En qué medida cree que el proceso de generación de turnos es accesible y cómodo?
7. ¿Qué cantidad de información sobre la atención de su turno cree que se proporciona?

Anexo C. Guía de entrevista

Tabla C1. Guía de entrevista

N.º	Pregunta	Respuesta	Observación
1	¿Quiénes intervienen actualmente en la gestión y atención de turnos?		
2	¿Qué problemas se han presentado con el proceso actual?		
3	¿En qué aspectos creen que mejoraría la situación la aplicación web de gestión y atención de turnos?		
4	¿Qué características debe tener la aplicación web de gestión y atención de turnos?		
5	¿Cuál es el objetivo de implementar la aplicación web?		
6	¿Qué tipo de consultas se van a atender con la aplicación web?		
7	¿Qué características debe cumplir la aplicación web para cubrir con este tipo de consultas?		
8	¿Cómo piensan que mejoraría la atención al cliente la implementación de la aplicación web?		

Anexo D. Material extra

- **Otras funcionalidades de la aplicación**
 - Filtrado de datos

Para el ejemplo de filtrado de datos se utilizó la pantalla de administración de atenciones debido a que es la pantalla que más filtros contiene.

Numero Turno	Usuario	Trabajador	Fecha	Hora Turno	Hora Inicio	Hora Final	Sucursal	Calificación	Observación
1	CO2	JAVIER ALEXANDER JIMENEZ CASTRO	2023-11-21	16:55	16:59	17:02	CENTRO	Calificación	Observación
2	CO3	JAVIER ALEXANDER JIMENEZ CASTRO	2023-11-21	16:58	17:03	17:10	CENTRO	Calificación	Observación

Figura D1. Pantalla de administración de atenciones

```

//! ACTUALIZACIÓN DE DATOS DE BUSQUEDA
useEffect(() => {
  const filtered = atenciones.filter(item => {
    const matchesSucursal = sucursal === 0 || item.Sucursal === sucursal;
    const matchesValoracion = valoracion === "" || item.Valoracion === valoracion;
    const matchesCadena = cadena === '' ||
      item.Nombre_Usuario.toLowerCase().includes(cadena.toLowerCase()) ||
      item.Nombre_Trabajador.toLowerCase().includes(cadena.toLowerCase()) ||
      item.Numero_Turno.toLowerCase().includes(cadena.toLowerCase());

    const fecha = obtenerFecha(item.Fecha_Turno);
    let matchesFecha = [];
    if (fechaInicial !== null && fechaFinal === null) {
      const fechaInicialFiltro = new Date(fechaInicial);
      const fechaFormateada = fechaInicialFiltro.toISOString().split('T')[0];

      matchesFecha = fechaInicial === null ||
        fecha === fechaFormateada;
    } else if (fechaFinal !== null && fechaInicial !== null) {
      //! FECHA DE ATENCION
      const partesFecha = fecha.split('-');
      const año = parseInt(partesFecha[0]);
      const mes = parseInt(partesFecha[1]);
      const dia = parseInt(partesFecha[2]);
      const fechaItem = new Date(Date.UTC(año, mes - 1, dia + 1));
      fechaItem.setHours(0,0,0,0);

      matchesFecha = (!fechaInicial || fechaItem >= fechaInicial) &&
        (!fechaFinal || fechaItem <= fechaFinal);
    }

    return matchesSucursal && matchesCadena && matchesValoracion && matchesFecha;
  });

  setDatosAtenciones(filtered);
}, [atenciones, sucursal, cadena, valoracion, fechaInicial, fechaFinal]);

```

Figura D2. Código JavaScript para filtrado de datos

El bloque de código que se muestra en la Figura D2, utiliza `useEffect` para actualizar la lista de datos en base a los cambios en los filtros, evaluando las coincidencias de cada filtro con la lista total de datos para crear una nueva lista solo con las coincidencias. En caso de no haber ninguna coincidencia, la lista será el conjunto total de datos sin ningún filtro.

- Obtener hora

```
//TODO: OBTENER HORA
const obtenerHora = (f) => {
  if(f) {
    const match = f.match(/\/Date((\d+)([+-]\d{4})\/\//);

    if (match) {
      const timestamp = parseInt(match[1], 10);
      const timeZoneOffset = -5 * 60 * 60;

      const date = new Date(timestamp + timeZoneOffset);

      const fechaFormateada = `${String(date.getHours() - 5).padStart(2, '0')}:${String(date.getMinutes()).padStart(2, '0')}`;

      return fechaFormateada;
    }
  } else {
    return 1;
  }
}
```

Figura D3. Código JavaScript para obtener la hora

El bloque de código que se muestra en la Figura D3, verifica si la fecha existe para extraer la información de tiempo en un formato específico. Se busca un patrón en el formato de fecha generada en .NET (/Date(timestamp)/) para devolver una cadena formateada que represente la hora.

- Obtener fecha

```
//TODO: OBTENER LA FECHA
const obtenerFecha = (f) => {
  if(f) {
    const match = f.match(/\/Date((\d+)([+-]\d{4})\/\//);

    if (match) {
      const timestamp = parseInt(match[1], 10);
      const timeZoneOffset = -5 * 60 * 60;

      const date = new Date(timestamp + timeZoneOffset);

      const fechaFormateada = `${date.getFullYear()}-${String(date.getMonth() + 1).padStart(2, '0')}-${String(date.getDate()).padStart(2, '0')}`;

      return fechaFormateada;
    }
  } else {
    return 1;
  }
}
```

Figura D4. Código JavaScript para obtener la fecha

El bloque de código que se muestra en la Figura D4, verifica si la fecha existe para extraer la información de tiempo en un formato específico. Se busca un patrón en el formato de fecha generada en .NET (/Date(timestamp)/) para devolver una cadena formateada que represente la fecha.

- Reporte en Excel

Para el ejemplo de reporte en Excel se utilizó la pantalla de administración de turnos.

Tabla de administración de turnos Cerrar Sesión

Sucursales - Tipo de consulta - Estados - Fecha de inicio - Fecha de fin - Buscar - Recargar

Numero Turno	Estado	Fecha	Motivo de Consulta	Sucursal	
1	CA1	ATENDIDO	2023-09-27	CAMBIOS	CENTRO
2	QU1	ATENDIDO	2023-09-27	QUEJAS Y RECLAMOS	CENTRO
3	CA2	ATENDIDO	2023-09-27	CAMBIOS	CENTRO
4	QU2	ATENDIDO	2023-09-27	QUEJAS Y RECLAMOS	CENTRO
5	QU1	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
6	CO1	ATENDIDO	2023-09-30	CONSULTAS	MATRIZ
7	PE1	ATENDIDO	2023-09-30	PETICION DE SERVICIO	MATRIZ
8	CA1	ATENDIDO	2023-09-30	CAMBIOS	MATRIZ
9	QU2	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
10	CO2	ATENDIDO	2023-09-30	CONSULTAS	MATRIZ

Anterior 1 2 3 4 5 ... 12 13 Siguiente

PDF
EXCEL

Figura D5. Pantalla de ejemplo (Reporte en Excel)

	A	B	C	D	E
1	Numero Turno	Estado	Fecha	Motivo de Consulta	Sucursal
2	CA1	ATENDIDO	2023-09-27	CAMBIOS	CENTRO
3	QU1	ATENDIDO	2023-09-27	QUEJAS Y RECLAMOS	CENTRO
4	CA2	ATENDIDO	2023-09-27	CAMBIOS	CENTRO
5	QU2	ATENDIDO	2023-09-27	QUEJAS Y RECLAMOS	CENTRO
6	QU1	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
7	CO1	ATENDIDO	2023-09-30	CONSULTAS	MATRIZ
8	PE1	ATENDIDO	2023-09-30	PETICION DE SERVICIO	MATRIZ
9	CA1	ATENDIDO	2023-09-30	CAMBIOS	MATRIZ
10	QU2	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
11	CO2	ATENDIDO	2023-09-30	CONSULTAS	MATRIZ
12	CA2	ATENDIDO	2023-09-30	CAMBIOS	MATRIZ
13	QU3	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
14	PE2	ATENDIDO	2023-09-30	PETICION DE SERVICIO	MATRIZ
15	QU4	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
16	PE3	ATENDIDO	2023-09-30	PETICION DE SERVICIO	MATRIZ
17	QU5	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
18	CA3	ATENDIENDO	2023-10-01	CAMBIOS	CENTRO

Reporte

Listo Bloq Displ Accesibilidad: todo correcto Configuración de visualización 100%

Figura D6. Reporte en Excel

En el archivo de Excel de la Figura D6 se observa que se encuentran los mismos datos que se muestran en la pantalla de la Figura D5.


```

//TODO: GENERAR EXCEL
const generateExcel = () => {
  const wb = XLSX.utils.book_new();
  const datos = datosTurnos.map((item) => [
    item.Numero_Turno,
    obtenerEstado(item.Estado),
    obtenerFecha(item.Fecha),
    obtenerConsulta(item.ID_Tipo_Consulta),
    obtenerSucursal(item.ID_Sucursal),
  ]);

  const headers = [
    "Numero Turno",
    "Estado",
    "Fecha",
    "Motivo de Consulta",
    "Sucursal",
  ];

  const wsData = [headers, ...datos];
  const ws = XLSX.utils.aoa_to_sheet(wsData);

  XLSX.utils.book_append_sheet(wb, ws, "Reporte");

  const wsCols = headers.map((header, index) => ({
    wch: header.length + 20,
  }));
  ws["!cols"] = wsCols;

  XLSX.writeFile(wb, "Reporte_Turnos.xlsx");
};

```

Figura D7. Código JavaScript para generar reporte en Excel

El bloque de código de la Figura D7 crea una lista de datos ordenados, dependiendo de los filtros que se hayan aplicado, después crea un documento Excel y escribe los datos en una hoja con las cabeceras especificadas. Por último, el documento completo se descarga automáticamente.

- Reporte en PDF

Para el ejemplo de reporte en PDF se utilizó la pantalla de administración de turnos.

Tabla de administración de turnos Cerrar Sesión

Sucursal: Tipo de consulta: Estado: Fecha de inicio: Fecha de fin: Buscar: Recargar

Numero Turno	Estado	Fecha	Motivo de Consulta	Sucursal	
1	CA1	ATENDIDO	2023-09-27	CAMBIO	CENTRO
2	QU1	ATENDIDO	2023-09-27	QUEJAS Y RECLAMOS	CENTRO
3	CA2	ATENDIDO	2023-09-27	CAMBIO	CENTRO
4	QU2	ATENDIDO	2023-09-27	QUEJAS Y RECLAMOS	CENTRO
5	QU1	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
6	CO1	ATENDIDO	2023-09-30	CONSULTAS	MATRIZ
7	PE1	ATENDIDO	2023-09-30	PETICION DE SERVICIO	MATRIZ
8	CA1	ATENDIDO	2023-09-30	CAMBIO	MATRIZ
9	QU2	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
10	CO2	ATENDIDO	2023-09-30	CONSULTAS	MATRIZ

Anterior 1 2 3 4 5 ... 12 13 Siguiente

PDF EXCEL

Figura D8. Pantalla de ejemplo (Reporte en PDF)

Reporte_Turnos.pdf 1 / 6 | 100% +

Tabla de Turnos

Numero Turno	Estado	Fecha	Motivo de Consulta	Sucursal
CA1	ATENDIDO	2023-09-27	CAMBIO	CENTRO
QU1	ATENDIDO	2023-09-27	QUEJAS Y RECLAMOS	CENTRO
CA2	ATENDIDO	2023-09-27	CAMBIO	CENTRO
QU2	ATENDIDO	2023-09-27	QUEJAS Y RECLAMOS	CENTRO
QU1	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
CO1	ATENDIDO	2023-09-30	CONSULTAS	MATRIZ
PE1	ATENDIDO	2023-09-30	PETICION DE SERVICIO	MATRIZ
CA1	ATENDIDO	2023-09-30	CAMBIO	MATRIZ
QU2	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
CO2	ATENDIDO	2023-09-30	CONSULTAS	MATRIZ
CA2	ATENDIDO	2023-09-30	CAMBIO	MATRIZ
QU3	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
PE2	ATENDIDO	2023-09-30	PETICION DE SERVICIO	MATRIZ
QU4	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
PE3	ATENDIDO	2023-09-30	PETICION DE SERVICIO	MATRIZ
QU5	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
CA3	ATENDIENDO	2023-10-01	CAMBIO	CENTRO
PE1	ATENDIENDO	2023-10-01	PETICION DE SERVICIO	CENTRO
CO3	ATENDIDO	2023-10-02	CONSULTAS	MATRIZ
CA3	ATENDIDO	2023-10-02	CAMBIO	MATRIZ
QU8	ATENDIDO	2023-10-02	QUEJAS Y RECLAMOS	MATRIZ
PE4	ATENDIDO	2023-10-02	PETICION DE SERVICIO	MATRIZ

Figura D9. Reporte en PDF

En el archivo PDF de la Figura D9 se observa que se encuentran los mismos datos que se muestran en la pantalla de la Figura D8.

```

//TODO: GENERAR PDF
const generatePDF = () => {
  const datos = datosTurnos.map((item) => [
    item.Numero_Turno,
    obtenerEstado(item.Estado),
    obtenerFecha(item.Fecha),
    obtenerConsulta(item.ID_Tipo_Consulta),
    obtenerSucursal(item.ID_Sucursal),
  ]);

  const doc = new jsPDF({
    orientation: "landscape",
    unit: "mm",
    format: [297, 210],
  });

  doc.text("Tabla de Turnos", 10, 10);

  const headers = [
    "Numero Turno",
    "Estado",
    "Fecha",
    "Motivo de Consulta",
    "Sucursal",
  ];

  doc.autoTable({ head: [headers], body: datos });

  doc.save("Reporte_Turnos.pdf");
};

```

Figura D10. Código JavaScript para generar reporte en PDF

El bloque de código de la Figura D10 crea una lista de datos ordenados, dependiendo de los filtros que se hayan aplicado, después crea un documento PDF y escribe los datos con las cabeceras especificadas. Por último, el documento completo se descarga automáticamente.

- Ventana para visualizar la calificación de una atención

Tabla de administración de atenciones

¿El trato del personal del balcón de servicios e información con los usuarios es cordial y respetuoso?:
1

¿Es adecuado el tiempo de espera para ser atendido en el balcón de servicios e información?:
2

¿El personal del balcón de servicios e información que atiende sus requerimientos o reclamos, muestra estar capacitado para brindarle soluciones?:
2

¿Se le informa adecuadamente cuanto tiempo tomará la atención del trámite solicitado?:
2

Valoración: **REGULAR**

CERRAR

Numero	Observación	Calificación
1	Observación	
2	Observación	
3	Observación	
4	Observación	
5	Observación	
6	Observación	
7	Observación	
8	Observación	
9	Observación	
10	Observación	

Anterior 1 2 3 4 5 6 7 8 9 Siguiente

Figura D11. Ventana de calificación

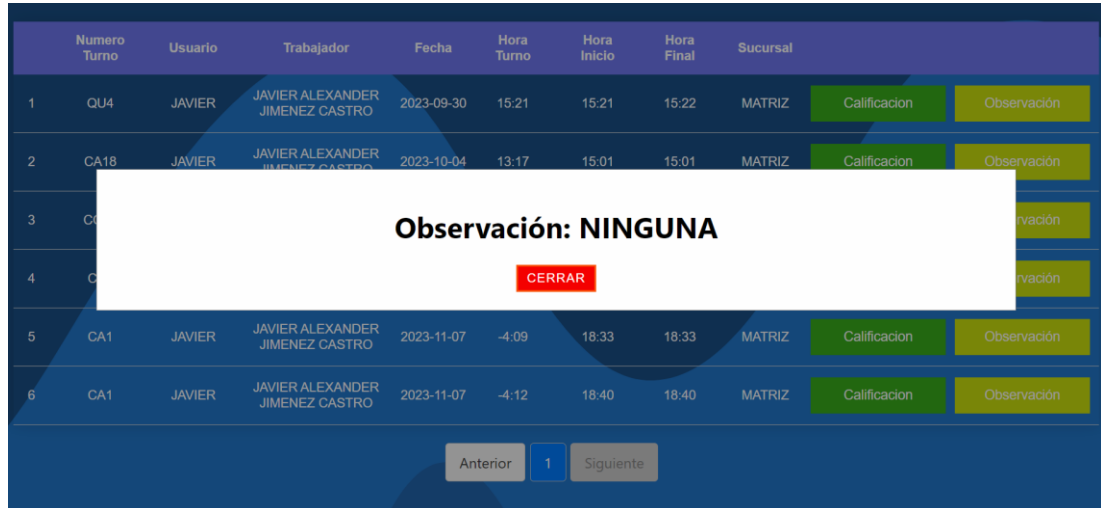
La ventana de la Figura D11 se abre al seleccionar la opción “Calificación” que se encuentra en cada fila y carga los valores de calificación de la atención. En caso de que una atención no se haya calificado, los valores se guardarán con “0” y “NO CALIFICADO” por defecto.

```
//TODO: VER CALIFICACION
const abrirModalCalificacion = (item) => {
  setAtencion(item);
  let modal = document.getElementById("modalCalificacion");
  modal.style.display = "block";
}
```

Figura D12. Código JavaScript para abrir la ventana de calificación

Este bloque de código representado en la Figura D12 coge los datos de la atención de la fila en la que se encuentra el botón para poder presentar los valores de calificación dentro de la ventana que se muestra.

- Ventana para visualizar las observaciones de una atención



The screenshot shows a table with columns: Numero Turno, Usuario, Trabajador, Fecha, Hora Turno, Hora Inicio, Hora Final, Sucursal, Calificación, and Observación. A modal window is open over the table, displaying the text "Observación: NINGUNA" and a red "CERRAR" button. The table data is as follows:

Numero Turno	Usuario	Trabajador	Fecha	Hora Turno	Hora Inicio	Hora Final	Sucursal	Calificación	Observación
1	QU4	JAVIER JIMENEZ CASTRO	2023-09-30	15:21	15:21	15:22	MATRIZ	Calificación	Observación
2	CA18	JAVIER JIMENEZ CASTRO	2023-10-04	13:17	15:01	15:01	MATRIZ	Calificación	Observación
3	CA1	JAVIER JIMENEZ CASTRO	2023-11-07	-4:09	18:33	18:33	MATRIZ	Calificación	Observación
4	CA1	JAVIER JIMENEZ CASTRO	2023-11-07	-4:12	18:40	18:40	MATRIZ	Calificación	Observación
5	CA1	JAVIER JIMENEZ CASTRO	2023-11-07	-4:12	18:40	18:40	MATRIZ	Calificación	Observación
6	CA1	JAVIER JIMENEZ CASTRO	2023-11-07	-4:12	18:40	18:40	MATRIZ	Calificación	Observación

Figura D13. Ventana de observaciones

La ventana de la Figura D13 se abre al seleccionar la opción “Observación” que se encuentra en cada fila y carga la observación u observaciones de la atención. En caso de finalizar la atención sin ingresar ninguna observación, se guardará con “NINGUNA” por defecto.

```

//TODO: VER OBSERVACION
const abrirModalObservacion = (item) => {
  setAtencion(item);
  let modal = document.getElementById("modalObservacion");
  modal.style.display = "block";
}

```

Figura D14. Código JavaScript para abrir la ventana de observaciones

El bloque de código de la Figura D14 coge los datos de la atención de la fila en la que se encuentra el botón para poder presentar la observación u observaciones dentro de la ventana que se muestra.

- Mensajes de alerta

Los mensajes de alerta se manejan a través de la función alert en base a distintos eventos programados.



Figura D15. Mensaje de alerta página principal

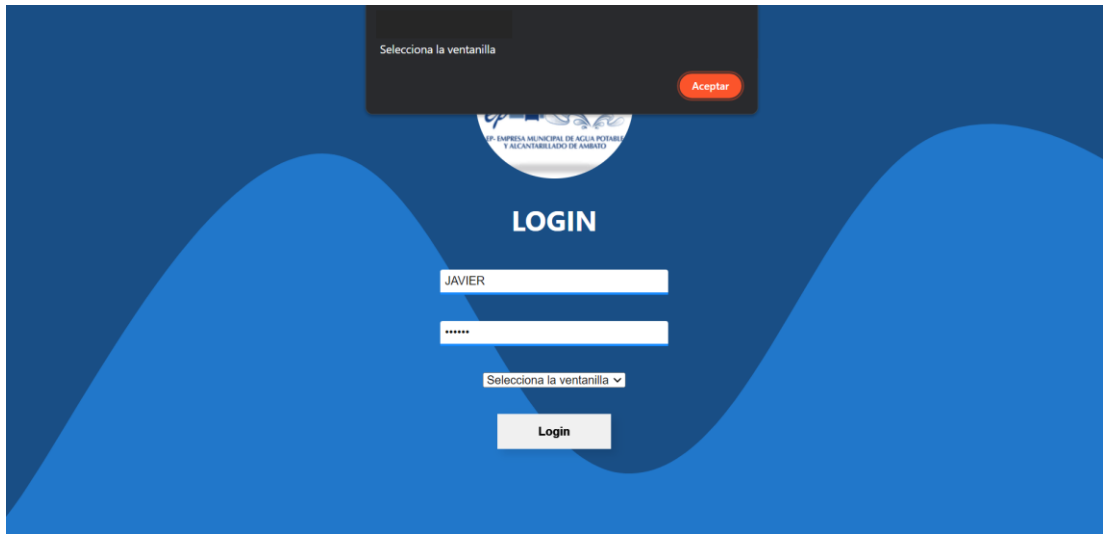


Figura D16. Mensaje de alerta selección de ventanilla

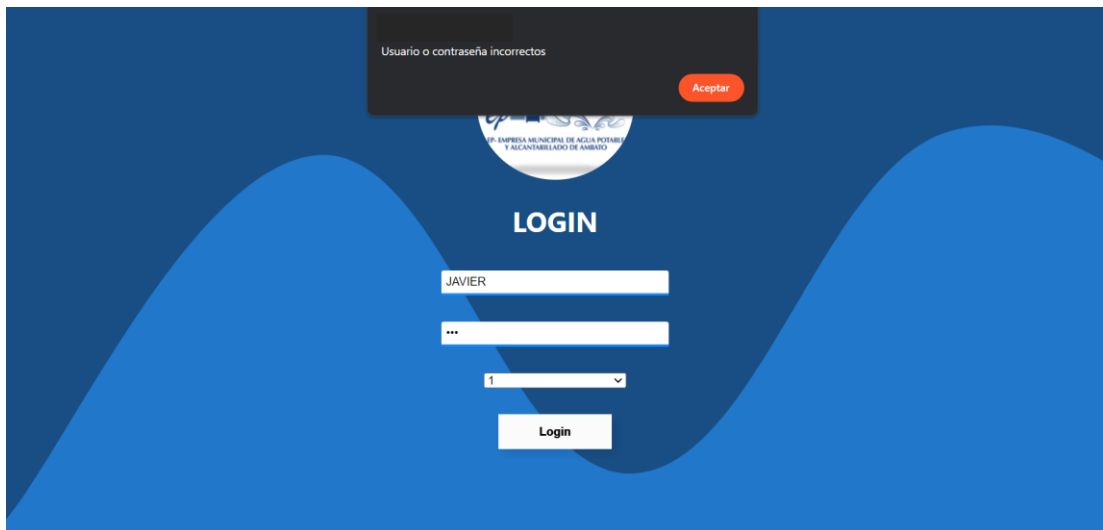


Figura D17. Mensaje de alerta inicio de sesión

- **Entidades**

Clases desarrolladas en C# para designar propiedades y facilitar la manipulación de datos.

- Atención

```

namespace Entidades
{
    public class Atencion
    {
        public Atencion()
        {
        }

        public Atencion(int id_Atencion, int id_Turno, int id_Usuario, int ventanilla, int estado, DateTime fecha_Inicio, DateTime fecha_Final, string observacion)
        {
            ID_Atencion = id_Atencion;
            ID_Usuario = id_Usuario;
            ID_Turno = id_Turno;
            Ventanilla = ventanilla;
            Estado = estado;
            Fecha_Inicio = fecha_Inicio;
            Fecha_Final = fecha_Final;
            Observacion = observacion;
        }

        public Atencion(int id_Turno, int id_Usuario, int ventanilla, int estado, DateTime fecha_Inicio, DateTime fecha_Final, string observacion)
        {
            ID_Usuario = id_Usuario;
            ID_Turno = id_Turno;
            Ventanilla = ventanilla;
            Estado = estado;
            Fecha_Inicio = fecha_Inicio;
            Fecha_Final = fecha_Final;
            Observacion = observacion;
        }

        public int ID_Atencion { get; set; }
        public int ID_Usuario { get; set; }
        public int ID_Turno { get; set; }
        public int Ventanilla { get; set; }
        public int Estado { get; set; }
        public DateTime Fecha_Inicio { get; set; }
        public DateTime Fecha_Final { get; set; }
        public string Observacion { get; set; }
    }
}

```

Figura D18. Entidad Atención

- Calificación

```

namespace Entidades
{
    public class Calificacion
    {
        public Calificacion()
        {
        }

        public Calificacion(int id_Atencion, int pregunta_1, int pregunta_2, int pregunta_3, int pregunta_4, string valoracion)
        {
            ID_Atencion = id_Atencion;
            Pregunta_1 = pregunta_1;
            Pregunta_2 = pregunta_2;
            Pregunta_3 = pregunta_3;
            Pregunta_4 = pregunta_4;
            Valoracion = valoracion;
        }

        public Calificacion(int pregunta_1, int pregunta_2, int pregunta_3, int pregunta_4, string valoracion)
        {
            Pregunta_1 = pregunta_1;
            Pregunta_2 = pregunta_2;
            Pregunta_3 = pregunta_3;
            Pregunta_4 = pregunta_4;
            Valoracion = valoracion;
        }

        public int ID_Atencion { get; set; }
        public int Pregunta_1 { get; set; }
        public int Pregunta_2 { get; set; }
        public int Pregunta_3 { get; set; }
        public int Pregunta_4 { get; set; }
        public string Valoracion { get; set; }
    }
}

```

Figura D19. Entidad Calificación

- Contador

```

namespace Entidades
{
    public class Contador
    {
        public Contador()
        {
        }

        public Contador(int id_Sucursal, int id_Tipo_Consulta, int numero)
        {
            ID_Sucursal = id_Sucursal;
            ID_Tipo_Consulta = id_Tipo_Consulta;
            Numero = numero;
        }

        public int ID_Sucursal { get; set; }
        public int ID_Tipo_Consulta { get; set; }
        public int Numero { get; set; }
    }
}

```

Figura D20. Entidad Contador

- Estado

```

namespace Entidades
{
    public class Estado
    {
        public Estado()
        {
        }

        public Estado(int id_Estado, string nombre)
        {
            ID_Estado = id_Estado;
            Nombre = nombre;
        }

        public Estado(string nombre)
        {
            Nombre = nombre;
        }

        public int ID_Estado { get; set; }
        public string Nombre { get; set; }
    }
}

```

Figura D21. Entidad Estado

- Rol


```

namespace Entidades
{
    public class Rol
    {
        public Rol()
        {
        }

        public Rol(int id_Rol, string nombre, int estado)
        {
            ID_Rol = id_Rol;
            Nombre = nombre;
            Estado = estado;
        }

        public Rol(string nombre, int estado)
        {
            Nombre = nombre;
            Estado = estado;
        }

        public Rol(int id_Rol, int estado)
        {
            ID_Rol = id_Rol;
            Estado = estado;
        }

        public int ID_Rol { get; set; }
        public string Nombre { get; set; }
        public int Estado { get; set; }
    }
}

```

Figura D22. Entidad Rol

- Sucursal

```

namespace Entidades
{
    public class Sucursal
    {
        public Sucursal()
        {
        }

        public Sucursal(string nombre, int numero_Ventanillas, int estado)
        {
            Nombre = nombre;
            Estado = estado;
            Numero_Ventanillas = numero_Ventanillas;
        }

        public Sucursal(int id_Sucursal, string nombre, int numero_Ventanillas, int estado)
        {
            ID_Sucursal = id_Sucursal;
            Nombre = nombre;
            Estado = estado;
            Numero_Ventanillas = numero_Ventanillas;
        }

        public Sucursal(int id_Sucursal, int estado)
        {
            ID_Sucursal = id_Sucursal;
            Estado = estado;
        }

        public int ID_Sucursal { get; set; }
        public string Nombre { get; set; }
        public int Numero_Ventanillas { get; set; }
        public int Estado { get; set; }
    }
}

```

Figura D23. Entidad Sucursal

- Tipo de consulta

```

namespace Entidades
{
    public class TipoConsulta
    {
        public TipoConsulta()
        {
        }

        public TipoConsulta(string nombre, int estado)
        {
            Nombre = nombre;
            Estado = estado;
        }

        public TipoConsulta(int id_Tipo_Consulta, int estado)
        {
            ID_Tipo_Consulta = id_Tipo_Consulta;
            Estado = estado;
        }

        public TipoConsulta(int id_Tipo_Consulta, string nombre, int estado)
        {
            ID_Tipo_Consulta = id_Tipo_Consulta;
            Nombre = nombre;
            Estado = estado;
        }

        public int ID_Tipo_Consulta { get; set; }
        public string Nombre { get; set; }
        public int Estado { get; set; }
    }
}

```

Figura D24. Entidad Tipo de consulta

- Trabajador

```

namespace Entidades
{
    public class Trabajador
    {
        public Trabajador()
        {
        }

        public Trabajador(string cedula, string primer_Nombre, string segundo_Nombre, string primer_Apellido, string segundo_Apellido, DateTime fecha_Nacimiento, int estado)
        {
            Cedula = cedula;
            Primer_Nombre = primer_Nombre;
            Segundo_Nombre = segundo_Nombre;
            Primer_Apellido = primer_Apellido;
            Segundo_Apellido = segundo_Apellido;
            Fecha_Nacimiento = fecha_Nacimiento;
            Estado = estado;
        }

        public Trabajador(int id_Trabajador, string cedula, string primer_Nombre, string segundo_Nombre, string primer_Apellido, string segundo_Apellido, DateTime fecha_Nacimiento, int estado)
        {
            ID_Trabajador = id_Trabajador;
            Cedula = cedula;
            Primer_Nombre = primer_Nombre;
            Segundo_Nombre = segundo_Nombre;
            Primer_Apellido = primer_Apellido;
            Segundo_Apellido = segundo_Apellido;
            Fecha_Nacimiento = fecha_Nacimiento;
            Estado = estado;
        }

        public Trabajador(int id_Trabajador, int estado)
        {
            ID_Trabajador = id_Trabajador;
            Estado = estado;
        }

        public int ID_Trabajador { get; set; }
        public string Cedula { get; set; }
        public string Primer_Nombre { get; set; }
        public string Segundo_Nombre { get; set; }
        public string Primer_Apellido { get; set; }
        public string Segundo_Apellido { get; set; }
        public int Estado { get; set; }
        public DateTime Fecha_Nacimiento { get; set; }
    }
}

```

Figura D25. Entidad Trabajador

- Turno

```

namespace Entidades
{
    public class Turno
    {
        public Turno()
        {
        }

        public Turno(int id_Tipo_Consulta, int id_Sucursal, DateTime fecha, string numero_Turno, int estado)
        {
            ID_Tipo_Consulta = id_Tipo_Consulta;
            ID_Sucursal = id_Sucursal;
            Fecha = fecha;
            Numero_Turno = numero_Turno;
            Estado = estado;
        }

        public Turno(int id_Turno, int id_Tipo_Consulta, int id_Sucursal, DateTime fecha, string numero_Turno, int estado)
        {
            ID_Turno = id_Turno;
            ID_Tipo_Consulta = id_Tipo_Consulta;
            ID_Sucursal = id_Sucursal;
            Fecha = fecha;
            Numero_Turno = numero_Turno;
            Estado = estado;
        }

        public Turno(int id_Turno, int estado)
        {
            ID_Turno = id_Turno;
            Estado = estado;
        }

        public int ID_Turno { get; set; }
        public int ID_Tipo_Consulta { get; set; }
        public int ID_Sucursal { get; set; }
        public DateTime Fecha { get; set; }
        public string Numero_Turno { get; set; }
        public int Estado { get; set; }
    }
}

```

Figura D26. Entidad Turno

- Usuario

```

namespace Entidades
{
    public class Usuario
    {
        public Usuario()
        {
        }

        public Usuario(string nombre, string clave, int id_Trabajador, int id_Rol, int estado)
        {
            Nombre = nombre;
            Clave = clave;
            ID_Trabajador = id_Trabajador;
            ID_Rol = id_Rol;
            Estado = estado;
        }

        public Usuario(int id_Usuario, string nombre, string clave, int id_Trabajador, int id_Rol, int estado)
        {
            ID_Usuario = id_Usuario;
            Nombre = nombre;
            Clave = clave;
            ID_Trabajador = id_Trabajador;
            ID_Rol = id_Rol;
            Estado = estado;
        }

        public Usuario(int id_Usuario, int estado)
        {
            ID_Usuario = id_Usuario;
            Estado = estado;
        }

        public int ID_Usuario { get; set; }
        public string Nombre { get; set; }
        public string Clave { get; set; }
        public int ID_Trabajador { get; set; }
        public int ID_Rol { get; set; }
        public int Estado { get; set; }

        // TOKEN DE VALIDACION
        public string Token { get; set; }
    }
}

```

Figura D27. Entidad Usuario

- **Servicios WCF**

Los servicios utilizan un atributo “OperationContract” para especificar que representan operaciones CRUD y un atributo “WebInvoke” para especificar un formato de solicitud y respuesta. Cada operación se encuentra asociada a un “URI template” para identificar la operación, un método (GET, POST, PUT, DELETE) dependiendo de la acción que se realice y un formato de respuesta JSON.

- Estados

```
//LISTA DE TODOS LOS ESTADOS
[OperationContract]
[WebInvoke(UriTemplate = "ListaEstados", Method = "GET",
    RequestFormat = WebMessageFormat.Json,
    ResponseFormat = WebMessageFormat.Json)]
List<Estado> DevolverListaEstados();

//BÚSQUEDA DE ESTADO POR ID
[OperationContract]
[WebInvoke(UriTemplate = "Estado?id={id}", Method = "GET",
    RequestFormat = WebMessageFormat.Json,
    ResponseFormat = WebMessageFormat.Json)]
Estado DevolverEstado(int id);

//NUEVO ESTADO
[OperationContract]
[WebInvoke(UriTemplate = "NuevoEstado", Method = "POST",
    RequestFormat = WebMessageFormat.Json,
    ResponseFormat = WebMessageFormat.Json,
    BodyStyle = WebMessageBodyStyle.Wrapped)]
Estado NuevoEstado(string nombre);

//ACTUALIZAR ESTADO
[OperationContract]
[WebInvoke(UriTemplate = "ActualizarEstado", Method = "PUT",
    RequestFormat = WebMessageFormat.Json,
    ResponseFormat = WebMessageFormat.Json,
    BodyStyle = WebMessageBodyStyle.Wrapped)]
Estado ActualizarEstado(int id_Estado, string nombre);
```

Figura D28. Servicios Estados

- Roles

```

//LISTA DE TODOS LOS ROLES
[OperationContract]
[WebInvoke(UriTemplate = "ListaRoles", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Rol> DevolverListaRoles();

//BÚSQUEDA DE ROL POR ID
[OperationContract]
[WebInvoke(UriTemplate = "Rol?id={id}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
Rol DevolverRol(int id);

//NUEVO ROL
[OperationContract]
[WebInvoke(UriTemplate = "NuevoRol", Method = "POST",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Rol NuevoRol(string nombre, int estado);

//ACTUALIZAR ROL
[OperationContract]
[WebInvoke(UriTemplate = "ActualizarRol", Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Rol ActualizarRol(int id_Rol, string nombre, int estado);

//ELIMINAR ROL
[OperationContract]
[WebInvoke(UriTemplate = "EliminarRol", Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Rol EliminarRol(int id_Rol, int estado);

```

Figura D29. Servicios Roles

- Tipos de consulta

```

//LISTA DE TODOS LOS TIPOS DE CONSULTA
[OperationContract]
[WebInvoke(UriTemplate = "ListaTiposConsulta", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<TipoConsulta> DevolverListaTiposConsulta();

//BÚSQUEDA DE TIPO DE CONSULTA POR ID
[OperationContract]
[WebInvoke(UriTemplate = "TipoConsulta?id={id}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
TipoConsulta DevolverTipoConsulta(int id);

//NUEVO TIPO DE CONSULTA
[OperationContract]
[WebInvoke(UriTemplate = "NuevoTipoConsulta", Method = "POST",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
TipoConsulta NuevoTipoConsulta(string nombre, int estado);

//ACTUALIZAR TIPO DE CONSULTA
[OperationContract]
[WebInvoke(UriTemplate = "ActualizarTipoConsulta", Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
TipoConsulta ActualizarTipoConsulta(int id_Tipo_Consulta, string nombre, int estado);

//ELIMINAR TIPO DE CONSULTA
[OperationContract]
[WebInvoke(UriTemplate = "EliminarTipoConsulta", Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
TipoConsulta EliminarTipoConsulta(int id_Tipo_Consulta, int estado);

```

Figura D30. Servicios Tipos de consulta

- Sucursales

```

//LISTA DE TODAS LAS SUCURSALES
[OperationContract]
[WebInvoke(UriTemplate = "ListaSucursales", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Sucursal> DevolverListaSucursales();

//BÚSQUEDA DE SUCURSAL POR ID
[OperationContract]
[WebInvoke(UriTemplate = "Sucursal?id={id}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
Sucursal DevolverSucursal(int id);

//NUEVA SUCURSAL
[OperationContract]
[WebInvoke(UriTemplate = "NuevaSucursal", Method = "POST",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Sucursal NuevaSucursal(string nombre, int numero, int estado);

//ACTUALIZAR SUCURSAL
[OperationContract]
[WebInvoke(UriTemplate = "ActualizarSucursal", Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Sucursal ActualizarSucursal(int id_Sucursal, string nombre, int numero, int estado);

//ELIMINAR SUCURSAL
[OperationContract]
[WebInvoke(UriTemplate = "EliminarSucursal", Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Sucursal EliminarSucursal(int id_Sucursal, int estado);

```

Figura D31. Servicios Sucursales

- Trabajadores

```
//LISTA DE TODOS LOS TRABAJADORES
[OperationContract]
[WebInvoke(UriTemplate = "ListaTrabajadores", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Trabajador> DevolverListaTrabajadores();

//BÚSQUEDA DE TRABAJADOR POR ID
[OperationContract]
[WebInvoke(UriTemplate = "Trabajador?id={id}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
Trabajador DevolverTrabajador(int id);

//NUEVO TRABAJADOR
[OperationContract]
[WebInvoke(UriTemplate = "NuevoTrabajador",
Method = "POST",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Trabajador NuevoTrabajador(string cedula, string primer_nombre, string segundo_nombre, string primer_apellido,
string segundo_apellido, DateTime fecha_nacimiento, int estado);

//ACTUALIZAR TRABAJADOR
[OperationContract]
[WebInvoke(UriTemplate = "ActualizarTrabajador",
Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Trabajador ActualizarTrabajador(int id_Trabajador, string cedula, string primer_nombre, string segundo_nombre, string primer_apellido,
string segundo_apellido, DateTime fecha_nacimiento, int estado);

//ELIMINAR TRABAJADOR
[OperationContract]
[WebInvoke(UriTemplate = "EliminarTrabajador", Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Trabajador EliminarTrabajador(int id_Trabajador, int estado);
```

Figura D32. Servicios Trabajadores

- Usuarios

```

//LISTA DE TODOS LOS USUARIOS
[OperationContract]
[WebInvoke(UriTemplate = "ListaUsuarios", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Usuario> DevolverListaUsuarios();

//LISTA DE TODOS LOS USUARIOS CON DATOS DEL TRABAJADOR
[OperationContract]
[WebInvoke(UriTemplate = "ListaDatosUsuarios", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Usuario> DevolverListaDatosUsuarios();

//USUARIO LOGIN
[OperationContract]
[WebInvoke(UriTemplate = "Usuario?nombre={nombre}&clave={clave}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
Usuario DevolverUsuario(string nombre, string clave);

//BÚSQUEDA DE USUARIO POR ID
[OperationContract]
[WebInvoke(UriTemplate = "UsuarioID?id={id}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
Usuario DevolverUsuarioID(int id);

//BÚSQUEDA DE USUARIOS POR ID DE TRABAJADOR
[OperationContract]
[WebInvoke(UriTemplate = "UsuariosTrabajadorID?id={id}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Usuario> DevolverUsuariosTrabajadorID(int id);

//NUEVO USUARIO
[OperationContract]
[WebInvoke(UriTemplate = "NuevoUsuario",
Method = "POST",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Usuario NuevoUsuario(string nombre, string clave, int id_Trabajador, int id_Rol, int estado);

//ACTUALIZAR USUARIO
[OperationContract]
[WebInvoke(UriTemplate = "ActualizarUsuario",
Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Usuario ActualizarUsuario(int id_Usuario, string nombre, string clave, int id_Trabajador, int id_Rol, int estado);

//ELIMINAR USUARIO
[OperationContract]
[WebInvoke(UriTemplate = "EliminarUsuario", Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Usuario EliminarUsuario(int id_Usuario, int estado);

```

Figura D33. Servicios Usuarios

- Turnos


```

//LISTA DE TODOS LOS TURNOS
[OperationContract]
[WebInvoke(UriTemplate = "ListaTurnos", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Turno> DevolverListaTurnos();

//BÚSQUEDA DE TURNO POR ID
[OperationContract]
[WebInvoke(UriTemplate = "TurnoID?id={id}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
Turno DevolverTurnoID(int id);

//BÚSQUEDA DE TURNOS POR ID DE SUCURSAL
[OperationContract]
[WebInvoke(UriTemplate = "TurnosIDSucursal?id={id}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Turno> DevolverTurnosIDSucursal(int id);

//BÚSQUEDA DE TURNOS POR ID DE SUCURSAL Y ESTADO
[OperationContract]
[WebInvoke(UriTemplate = "TurnosSucursalEstado?id={id}&estado={estado}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Turno> DevolverTurnosSucursalEstado(int id, int estado);

//NUEVO TURNO
[OperationContract]
[WebInvoke(UriTemplate = "NuevoTurno",
Method = "POST",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Turno NuevoTurno(int id_Tipo_Consulta, int id_Sucursal, DateTime fecha, string numero_turno, int estado);

//ACTUALIZAR TURNO
[OperationContract]
[WebInvoke(UriTemplate = "ActualizarTurno",
Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Turno ActualizarTurno(int id_Turno, int id_Tipo_Consulta, int id_Sucursal, DateTime fecha, string numero_turno, int estado);

//ACTUALIZAR ESTADO TURNO
[OperationContract]
[WebInvoke(UriTemplate = "ActualizarEstadoTurno", Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Turno ActualizarEstadoTurno(int id_Turno, int estado);

```

Figura D34. Servicios Turnos

- Atenciones

```

//LISTA DE TODAS LAS ATENCIONES
[OperationContract]
[WebInvoke(UriTemplate = "ListaAtenciones", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Atencion> DevolverListaAtenciones();

//LISTA DE TODAS LAS ATENCIONES CON DATOS DE TRABAJADOR, USUARIO, TURNO Y SUCURSAL
[OperationContract]
[WebInvoke(UriTemplate = "ListaDatosAtenciones", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Atencion> DevolverListaDatosAtenciones();

//BÚSQUEDA DE ATENCION POR ID DE TURNO
[OperationContract]
[WebInvoke(UriTemplate = "AtencionIDTurno?id={id}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
Atencion DevolverAtencionIDTurno(int id);

//BÚSQUEDA DE ATENCIONES POR ESTADO
[OperationContract]
[WebInvoke(UriTemplate = "AtencionesEstado?estado={estado}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Atencion> DevolverAtencionesEstado(int estado);

//NUEVA ATENCION
[OperationContract]
[WebInvoke(UriTemplate = "NuevaAtencionPOST",
Method = "POST",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Atencion NuevaAtencion(int id_Turno, int id_Usuario, int ventanilla, int estado, DateTime fecha_Inicio, DateTime fecha_Final, string observacion);

//ACTUALIZAR ATENCION
[OperationContract]
[WebInvoke(UriTemplate = "ActualizarAtencion",
Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Atencion ActualizarAtencion(int id_Atencion, int id_Turno, int id_Usuario, int ventanilla, int estado, DateTime fecha_Inicio, DateTime fecha_Final, string observacion);

//ELIMINAR ATENCION
[OperationContract]
[WebInvoke(UriTemplate = "EliminarAtencion",
Method = "DELETE",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
String EliminarAtencion(int id_Atencion);

```

Figura D35. Servicios Atenciones

▪ Calificaciones

```

//LISTA DE TODAS LAS CALIFICACIONES
[OperationContract]
[WebInvoke(UriTemplate = "ListaCalificaciones", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Calificacion> DevolverListaCalificaciones();

//BÚSQUEDA DE ATENCION POR ID DE ATENCION
[OperationContract]
[WebInvoke(UriTemplate = "CalificacionIDAtencion?id={id}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
Calificacion DevolverCalificacionIDAtencion(int id);

//NUEVA CALIFICACION
[OperationContract]
[WebInvoke(UriTemplate = "NuevaCalificacion",
Method = "POST",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Calificacion NuevaCalificacion(int id_Atencion, int pregunta_1, int pregunta_2, int pregunta_3, int pregunta_4, string valoracion);

//ACTUALIZAR CALIFICACION
[OperationContract]
[WebInvoke(UriTemplate = "ActualizarCalificacion",
Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Calificacion ActualizarCalificacion(int id_Atencion, int pregunta_1, int pregunta_2, int pregunta_3, int pregunta_4, string valoracion);

//ELIMINAR CALIFICACION
[OperationContract]
[WebInvoke(UriTemplate = "EliminarCalificacion",
Method = "DELETE",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
String EliminarCalificacion(int id_Atencion);

```

Figura D36. Servicios Calificaciones

- Contadores

```
//LISTA DE TODOS LOS CONTADORES
[OperationContract]
[WebInvoke(UriTemplate = "ListaContadores", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
List<Contador> DevolverListaContadores();

//BÚSQUEDA DE CONTADOR POR ID DE SUCURSAL Y CONSULTA
[OperationContract]
[WebInvoke(UriTemplate = "Contador?id_Sucursal={id_Sucursal}&id_Consulta={id_Consulta}", Method = "GET",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json)]
Contador DevolverContador(int id_Sucursal, int id_Consulta);

//NUEVO CONTADOR
[OperationContract]
[WebInvoke(UriTemplate = "NuevoContador", Method = "POST",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Contador NuevoContador(int id_Sucursal, int id_Tipo_Consulta, int numero);

//ACTUALIZAR CONTADOR
[OperationContract]
[WebInvoke(UriTemplate = "ActualizarContador", Method = "PUT",
RequestFormat = WebMessageFormat.Json,
ResponseFormat = WebMessageFormat.Json,
BodyStyle = WebMessageBodyStyle.Wrapped)]
Contador ActualizarContador(int id_Sucursal, int id_Tipo_Consulta, int numero);
```

Figura D37. Servicios Contadores

Anexo E. Diccionario de datos de la base de datos

- **Estados**

Tabla E1. Diccionario Estados

NOMBRE DE TABLA	ESTADOS
Atributo	Descripción
ID_Estado	Almacena el identificador único del estado
Nombre	Almacena el nombre del estado

- **Trabajadores**

Tabla E2. Diccionario Trabajadores

NOMBRE DE TABLA	TRABAJADORES
Atributo	Descripción
ID_Trabajador	Almacena el identificador único del trabajador
Cedula	Almacena la cédula del trabajador
Primer_Nombre	Almacena el primer nombre del trabajador
Segundo_Nombre	Almacena el segundo nombre del trabajador
Primer_Apellido	Almacena el primer apellido del trabajador
Segundo_Apellido	Almacena el segundo apellido del trabajador
Fecha_Nacimiento	Almacena la fecha de nacimiento del trabajador
Estado	Almacena el identificador del estado del trabajador

- **Roles**

Tabla E3. Diccionario Roles

NOMBRE DE TABLA	ROLES
Atributo	Descripción
ID_Rol	Almacena el identificador único del rol
Nombre	Almacena el nombre del rol
Estado	Almacena el identificador del estado del rol

- **Usuarios**

Tabla E4. Diccionario Usuarios

NOMBRE DE TABLA	USUARIOS
Atributo	Descripción
ID_Usuario	Almacena el identificador único del usuario
Nombre	Almacena el nombre del usuario
Clave	Almacena la clave del usuario
ID_Trabajador	Almacena el identificador del trabajador al que pertenece el usuario
Rol	Almacena el identificador del rol del usuario
Estado	Almacena el identificador del estado del usuario

- **Tipos de consulta**

Tabla E5. Diccionario Tipos de consulta

NOMBRE DE TABLA	TIPOS_CONSULTA
Atributo	Descripción
ID_Tipo_Consulta	Almacena el identificador único del tipo de consulta
Nombre	Almacena el nombre del tipo de consulta
Estado	Almacena el identificador del estado del tipo de consulta

- **Sucursales**

Tabla E6. Diccionario Sucursales

NOMBRE DE TABLA	SUCURSALES
Atributo	Descripción
ID_Sucursal	Almacena el identificador único de la sucursal
Nombre	Almacena el nombre de la sucursal
Numero_Ventanillas	Almacena el número de ventanillas que hay en la sucursal
Estado	Almacena el identificador del estado de la sucursal

- **Contadores**

Tabla E7. Diccionario Contadores

NOMBRE DE TABLA	CONTADORES
Atributo	Descripción
ID_Sucursal	Almacena el identificador de la sucursal del contador
ID_Tipo_Consulta	Almacena el identificador del tipo de consulta del contador
Numero	Almacena el número del contador

- **Turnos**

Tabla E8. Diccionario Turnos

NOMBRE DE TABLA	TURNOS
Atributo	Descripción
ID_Turno	Almacena el identificador único del turno
ID_Tipo_Consulta	Almacena el identificador el tipo de consulta para la que se generó el turno
ID_Sucursal	Almacena el identificador de la sucursal en la que se generó el turno
Fecha	Almacena la fecha y la hora en la que se generó el turno
Numero_Turno	Almacena el código y número del turno
Estado	Almacena el identificador del estado del turno

- **Atención**

Tabla E9. Diccionario Atención

NOMBRE DE TABLA	ATENCION
Atributo	Descripción
ID_Atención	Almacena el identificador único de la atención
ID_Turno	Almacena el identificador del turno que se atiende

Atributo	Descripción
ID_Usuario	Almacena el identificador del usuario que atiende el turno
Ventanilla	Almacena el número de ventanilla en la que se atiende
Estado	Almacena el identificador del estado de la atención
Fecha_Inicio	Almacena la fecha y hora a la que inicia la atención
Fecha_Final	Almacena la fecha y hora a la que finaliza la atención
Observación	Almacena las observaciones de la atención

- **Calificación**

Tabla E10. Diccionario Calificación

NOMBRE DE TABLA		CALIFICACION
Atributo	Descripción	
ID_Atencion	Almacena el identificador de la atención que se califica	
Pregunta_1	Almacena el valor de la pregunta 1	
Pregunta_2	Almacena el valor de la pregunta 2	
Pregunta_3	Almacena el valor de la pregunta 3	
Pregunta_4	Almacena el valor de la pregunta 4	
Valoracion	Almacena la valoración general de la atención	

Anexo F. Manual de usuario



Figura F1. Manual de usuario – Pag. 1



Figura F2. Manual de usuario – Pag. 2

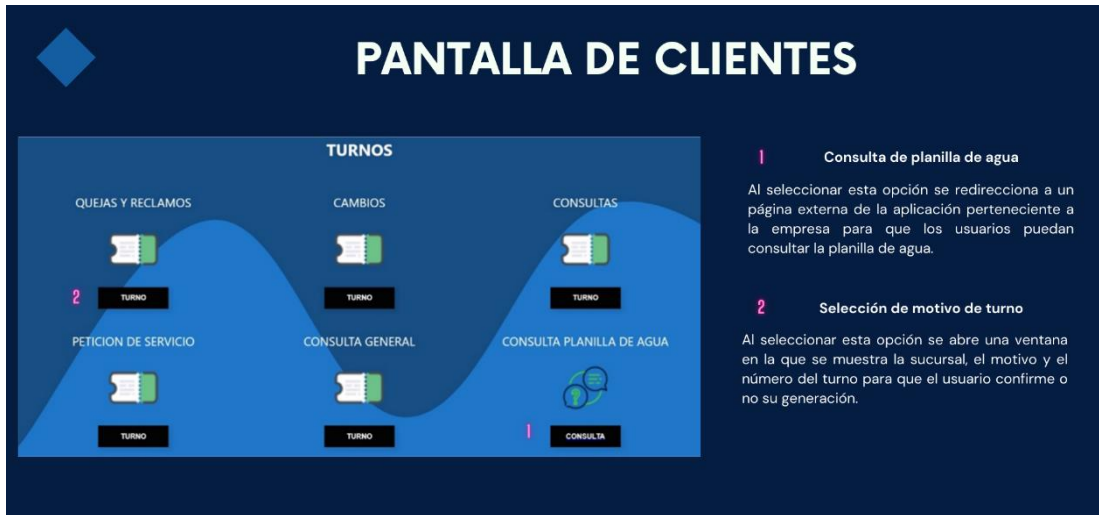


Figura F3. Manual de usuario – Pag. 3



Figura F 4. Manual de usuario – Pag. 4

PANTALLA DE INFORMACIÓN

Siguientes:

- CA2
- CA8
- QU4

Atendiendo:

CA2 EN VENTANILLA 1

EP-EMAPA-A



Pantalla

En esta pantalla no se puede realizar ninguna acción, la pantalla tiene como funcionalidades:

- Mostrar los 3 turnos que están en espera de atención, en el orden en el que van a ser atendidos.
- Mostrar los turnos que están siendo atendidos, mostrando el número del turno atendido además de la ventanilla en la que está siendo atendido.
- Mostrar un vídeo que se reproduce en bucle.

Figura F5. Manual de usuario – Pag. 5

PANTALLA DE CALIFICACIÓN (INICIO)

TURNO 1

¿El trato del personal del balcón de servicios e información con los usuarios es

Seleccione el número de ventanilla

Seleccione la ventanilla
Seleccione la ventanilla

Siguiente

Selección de ventanilla

Para acceder a la encuesta de satisfacción, se debe seleccionar el número de ventanilla de manera obligatoria.

En caso de no seleccionar ninguna ventanilla, la ventana no se cierra y no se carga la encuesta.

Figura F6. Manual de usuario – Pag. 6

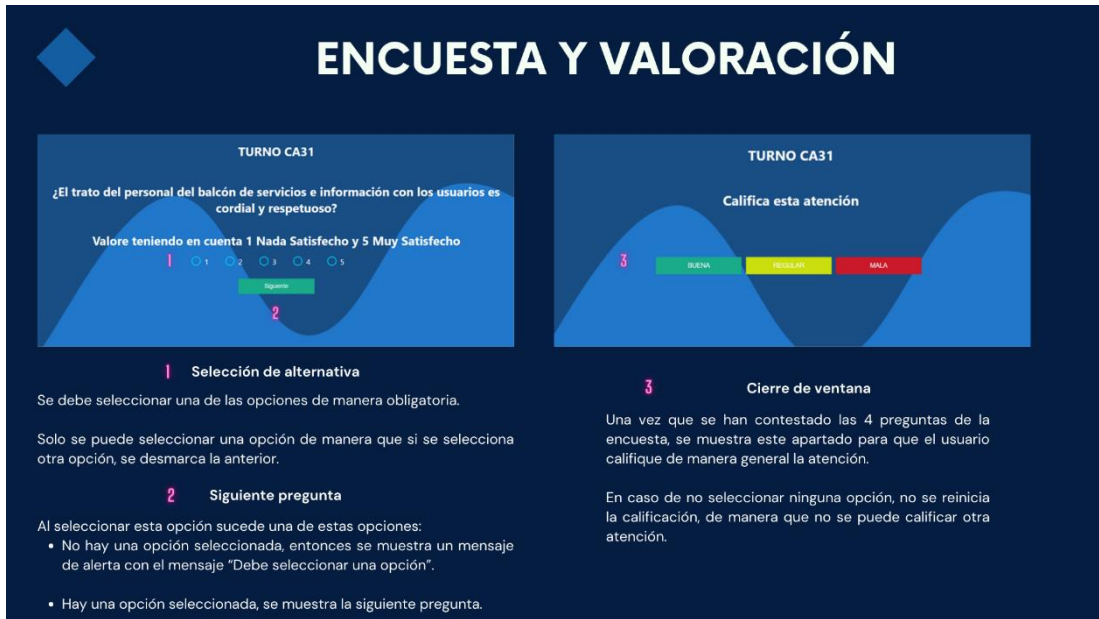


Figura F7. Manual de usuario – Pag. 7



Figura F8. Manual de usuario – Pag. 8

PANTALLA DE ATENCIÓN DE TURNOS

1 Cierre de sesión
Al seleccionar esta opción, se cierra la sesión del usuario y se redirecciona a la pantalla principal de la aplicación.

2 Atención de turno
Al seleccionar esta opción se abre una ventana con la información del turno, además el turno desaparece de la lista.

3 Ingreso de observaciones
En esta sección se ingresan las observaciones de la atención.

4 Cancelación de atención
Al seleccionar esta opción, se cierra la ventana y el turno vuelve a aparecer en la lista.

5 Finalización de atención
Al seleccionar esta opción, se completa la atención y se cierra la ventana.

Figura F9. Manual de usuario – Pag. 9

PANTALLA PRINCIPAL DE ADMINISTRADOR

1 Cierre de sesión
Al seleccionar esta opción, se cierra la sesión del usuario y se redirecciona a la pantalla principal de la aplicación.

2 Administración de atenciones
Al seleccionar esta opción se redirecciona a la pantalla de administración de atenciones.

3 Administración de turnos
Al seleccionar esta opción se redirecciona a la pantalla de administración de turnos.

4 Administración de usuarios
Al seleccionar esta opción se redirecciona a la pantalla de administración de usuarios.

5 Administración de trabajadores
Al seleccionar esta opción se redirecciona a la pantalla de administración de trabajadores.

6 Administración de sucursales
Al seleccionar esta opción se redirecciona a la pantalla de administración de sucursales.

7 Administración de tipos de consulta
Al seleccionar esta opción se redirecciona a la pantalla de administración de tipos de consulta.

8 Administración de tipos de consulta
Al seleccionar esta opción se redirecciona a la pantalla de administración de roles.

2 Administración de atenciones
Al seleccionar esta opción se redirecciona a la pantalla de administración de atenciones.

3 Administración de turnos
Al seleccionar esta opción se redirecciona a la pantalla de administración de turnos.

4 Administración de usuarios
Al seleccionar esta opción se redirecciona a la pantalla de administración de usuarios.

5 Administración de trabajadores
Al seleccionar esta opción se redirecciona a la pantalla de administración de trabajadores.

Figura F10. Manual de usuario – Pag. 10

PANTALLA DE ADMINISTRACIÓN DE ATENCIONES

1 Cierre de sesión
Al seleccionar esta opción, se cierra la sesión del usuario y se redirecciona a la pantalla principal de la aplicación.

2 Reseteo de filtros
Al seleccionar esta opción se quitan todos los filtros de datos.

3 Filtrado de datos
En esta sección se encuentran los filtros que se pueden aplicar.

4 Calificación de atención
Al seleccionar esta opción se abre una ventana que muestra la calificación que corresponde a una atención en particular.

5 Observaciones de la atención
Al seleccionar esta opción se abre una ventana que muestra las observaciones que corresponden a una atención en particular.

6 Paginación de datos
Esta sección corresponde a la paginación de los datos, seleccionando las diferentes opciones se muestran secciones de datos diferentes.

7 Reporte en PDF
Al seleccionar esta opción se descarga un archivo PDF con los datos de la tabla.

8 Reporte en Excel
Al seleccionar esta opción se descarga un archivo Excel con los datos de la tabla.

Numero Turno	Usuario	Trabajador	Fecha	Hora Turno	Hora Inicio	Hora Final	Recursal	Calificación	Observación
1	CO2	JAVIER JAVIER ALEXANDER JIMENEZ CASTRO	2023-11-21	16:55	16:59	17:02	CENTRO	Calificación	Observación
2	CO3	JAVIER JAVIER ALEXANDER JIMENEZ CASTRO	2023-11-21	16:58	17:03	17:10	CENTRO	Calificación	Observación

Figura F11. Manual de usuario – Pag. 11

CALIFICACIÓN Y OBSERVACIONES

1 Cierre de ventana de calificación
Esta ventana solo tienen como finalidad mostrar los valores de calificación de una atención en particular.
Al seleccionar esta opción se cierra la ventana.

2 Cierre de ventana de observaciones
Esta ventana solo tienen como finalidad mostrar las observaciones de una atención en particular.
Al seleccionar esta opción se cierra la ventana.

Figura F12. Manual de usuario – Pag. 12

PANTALLA DE ADMINISTRACIÓN DE TURNOS

1 Cierre de sesión
Al seleccionar esta opción, se cierra la sesión del usuario y se redirecciona a la pantalla principal de la aplicación.

2 Reseteo de filtros
Al seleccionar esta opción se quitan todos los filtros de datos.

3 Filtrado de datos
En esta sección se encuentran los filtros que se pueden aplicar.

4 Paginación de datos
Esta sección corresponde a la paginación de los datos, seleccionando las diferentes opciones se muestran secciones de datos diferentes.

5 Reporte en PDF
Al seleccionar esta opción se descarga un archivo PDF con los datos de la tabla.

6 Reporte en Excel
Al seleccionar esta opción se descarga un archivo Excel con los datos de la tabla.

Figura F13. Manual de usuario – Pag. 13

PANTALLA DE ADMINISTRACIÓN DE USUARIOS

1 Cierre de sesión
Al seleccionar esta opción se cierra la sesión del usuario y se redirecciona a la página principal de la aplicación.

2 Reseteo de filtros
Al seleccionar esta opción se quitan todos los filtros de datos.

3 Filtrado de datos
En esta sección se encuentran los filtros que se pueden aplicar.

4 Edición
Al seleccionar esta opción se abre una ventana en la que se muestran los datos del usuario para editar.

5 Activación / Desactivación
Al seleccionar esta opción se abre una ventana en la que se muestra un mensaje de confirmación de activación o desactivación de un usuario.

6 Creación de usuario
Al seleccionar esta opción se abre una ventana en la que se muestran los datos que se deben ingresar para crear un nuevo usuario.

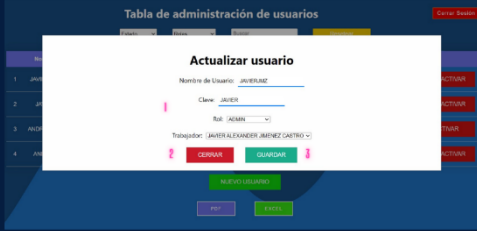
7 Reporte en PDF
Al seleccionar esta opción se descarga un archivo PDF con los datos de la tabla.

8 Reporte en Excel
Al seleccionar esta opción se descarga un archivo Excel con los datos de la tabla.

Figura F14. Manual de usuario – Pag. 14

EDICIÓN Y ACTIVACIÓN / DESACTIVACIÓN (USUARIOS)

Para que se actualicen los datos se debe recargar la página.



1 Edición de datos

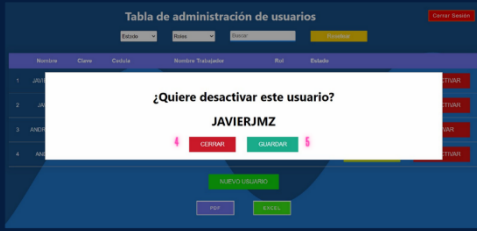
En esta sección se encuentran los datos para que puedan ser editados.

2 Cancelación de edición

Al seleccionar esta opción se cierra la ventana y no se edita ningún dato.

3 Confirmación de edición

Al seleccionar esta opción se cierra la ventana y se guardan los datos editados.



4 Cancelación de activación / desactivación

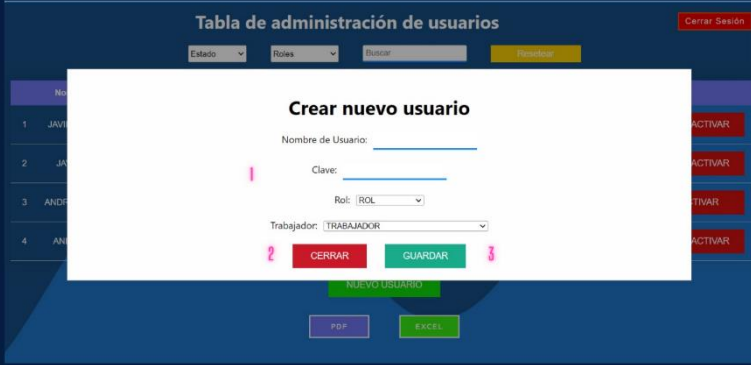
Al seleccionar esta opción se cierra la ventana y no se activa o desactiva el usuario.

5 Confirmación de activación / desactivación

Al seleccionar esta opción se cierra la ventana y se activa o desactiva el usuario.

Figura F15. Manual de usuario – Pag. 15

CREACIÓN (USUARIO)



1 Ingreso de datos

En esta sección se encuentran cada uno de los datos que deben ser ingresados para crear un nuevo usuario.

2 Cancelación de creación

Al seleccionar esta opción se cierra la ventana y no se guarda el nuevo usuario.

3 Confirmación de creación

Al seleccionar esta opción se cierra la ventana y se guarda el nuevo usuario.

Figura F16. Manual de usuario – Pag. 16

PANTALLA DE ADMINISTRACIÓN DE TRABAJADORES

1 Cierre de sesión
Al seleccionar esta opción se cierra la sesión del usuario y se redirecciona a la página principal de la aplicación.

2 Reseteo de filtros
Al seleccionar esta opción se quitan todos los filtros de datos.

3 Filtrado de datos
En esta sección se encuentran los filtros que se pueden aplicar.

4 Edición
Al seleccionar esta opción se abre una ventana en la que se muestran los datos del trabajador para editar.

5 Activación / Desactivación
Al seleccionar esta opción se abre una ventana en la que se muestra un mensaje de confirmación de activación o desactivación de un trabajador.

6 Creación de trabajador
Al seleccionar esta opción se abre una ventana en la que se muestran los datos que se deben ingresar para crear un nuevo trabajador.

7 Reporte en PDF
Al seleccionar esta opción se descarga un archivo PDF con los datos de la tabla.

8 Reporte en Excel
Al seleccionar esta opción se descarga un archivo Excel con los datos de la tabla.

Figura F17. Manual de usuario – Pag. 17

EDICIÓN Y ACTIVACIÓN / DESACTIVACIÓN (TRABAJADORES)

Para que se actualicen los datos se debe recargar la página.

1 Edición de datos
En esta sección se encuentran los datos para que puedan ser editados.

2 Cancelación de edición
Al seleccionar esta opción se cierra la ventana y no se edita ningún dato.

3 Confirmación de edición
Al seleccionar esta opción se cierra la ventana y se guardan los datos editados.

4 Cancelación de activación / desactivación
Al seleccionar esta opción se cierra la ventana y no se activa o desactiva el trabajador.

5 Confirmación de activación / desactivación
Al seleccionar esta opción se cierra la ventana y se activa o desactiva el trabajador.

Figura F18. Manual de usuario – Pag. 18

CREACIÓN (TRABAJADOR)

1 Ingreso de datos
En esta sección se encuentran cada uno de los datos que deben ser ingresados para crear un nuevo trabajador.

2 Cancelación de creación
Al seleccionar esta opción se cierra la ventana y no se guarda el nuevo trabajador.

3 Confirmación de creación
Al seleccionar esta opción se cierra la ventana y se guarda el nuevo trabajador.

Figura F19. Manual de usuario – Pag. 19

PANTALLA DE ADMINISTRACIÓN DE SUCURSALES

1 Cierre de sesión
Al seleccionar esta opción se cierra la sesión del usuario y se redirecciona a la página principal de la aplicación.

2 Reseteo de filtros
Al seleccionar esta opción se quitan todos los filtros de datos.

3 Filtrado de datos
En esta sección se encuentran los filtros que se pueden aplicar.

4 Edición
Al seleccionar esta opción se abre una ventana en la que se muestran los datos de la sucursal para editar.

5 Activación / Desactivación
Al seleccionar esta opción se abre una ventana en la que se muestra un mensaje de confirmación de activación o desactivación de una sucursal.

6 Creación de sucursal
Al seleccionar esta opción se abre una ventana en la que se muestran los datos que se deben ingresar para crear una nueva sucursal.

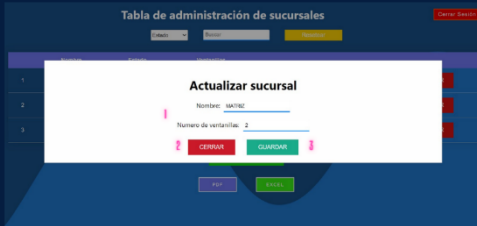
7 Reporte en PDF
Al seleccionar esta opción se descarga un archivo PDF con los datos de la tabla.

8 Reporte en Excel
Al seleccionar esta opción se descarga un archivo Excel con los datos de la tabla.

Figura F20. Manual de usuario – Pag. 20

EDICIÓN Y ACTIVACIÓN / DESACTIVACIÓN (SUCURSALES)

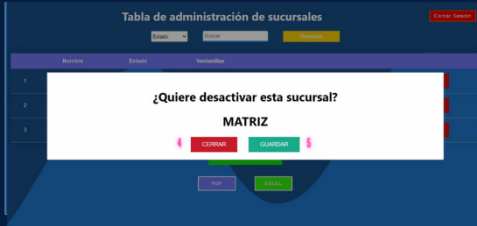
Para que se actualicen los datos se debe recargar la página.



1 Edición de datos
En esta sección se encuentran los datos para que puedan ser editados.

2 Cancelación de edición
Al seleccionar esta opción se cierra la ventana y no se edita ningún dato.

3 Confirmación de edición
Al seleccionar esta opción se cierra la ventana y se guardan los datos editados.




4 Cancelación de activación / desactivación
Al seleccionar esta opción se cierra la ventana y no se activa o desactiva la sucursal.

5 Confirmación de activación / desactivación
Al seleccionar esta opción se cierra la ventana y se activa o desactiva la sucursal.

Figura F21. Manual de usuario – Pag. 21

CREACIÓN (SUCURSAL)



1 Ingreso de datos
En esta sección se encuentran cada uno de los datos que deben ser ingresados para crear una nueva sucursal.

2 Cancelación de creación
Al seleccionar esta opción se cierra la ventana y no se guarda la nueva sucursal.

3 Confirmación de creación
Al seleccionar esta opción se cierra la ventana y se guarda la nueva sucursal.

Figura F22. Manual de usuario – Pag. 22

PANTALLA DE ADMINISTRACIÓN DE TIPOS DE CONSULTA

1 Cierre de sesión
Al seleccionar esta opción se cierra la sesión del usuario y se redirecciona a la página principal de la aplicación.

2 Reseteo de filtros
Al seleccionar esta opción se quitan todos los filtros de datos.

3 Filtrado de datos
En esta sección se encuentran los filtros que se pueden aplicar.

4 Edición
Al seleccionar esta opción se abre una ventana en la que se muestran los datos del tipo de consulta para editar.

5 Activación / Desactivación
Al seleccionar esta opción se abre una ventana en la que se muestra un mensaje de confirmación de activación o desactivación de un tipo de consulta.

6 Creación de tipo de consulta
Al seleccionar esta opción se abre una ventana en la que se muestran los datos que se deben ingresar para crear un nuevo tipo de consulta.

7 Reporte en PDF
Al seleccionar esta opción se descarga un archivo PDF con los datos de la tabla.

8 Reporte en Excel
Al seleccionar esta opción se descarga un archivo Excel con los datos de la tabla.

Figura F23. Manual de usuario – Pag. 23

EDICIÓN Y ACTIVACIÓN / DESACTIVACIÓN (TIPOS DE CONSULTA)

Para que se actualicen los datos se debe recargar la página.

1 Edición de datos
En esta sección se encuentran los datos para que puedan ser editados.

2 Cancelación de edición
Al seleccionar esta opción se cierra la ventana y no se edita ningún dato.

3 Confirmación de edición
Al seleccionar esta opción se cierra la ventana y se guardan los datos editados.

4 Cancelación de activación / desactivación
Al seleccionar esta opción se cierra la ventana y no se activa o desactiva el tipo de consulta.

5 Confirmación de activación / desactivación
Al seleccionar esta opción se cierra la ventana y se activa o desactiva el tipo de consulta.

Figura F24. Manual de usuario – Pag. 24

CREACIÓN (TIPO DE CONSULTA)

1 Ingreso de datos
En esta sección se encuentran cada uno de los datos que deben ser ingresados para crear un nuevo tipo de consulta.

2 Cancelación de creación
Al seleccionar esta opción se cierra la ventana y no se guarda el nuevo tipo de consulta.

3 Confirmación de creación
Al seleccionar esta opción se cierra la ventana y se guarda el nuevo tipo de consulta.

Figura F25. Manual de usuario – Pag. 25

PANTALLA DE ADMINISTRACIÓN DE ROLES

1 Cierre de sesión
Al seleccionar esta opción se cierra la sesión del usuario y se redirecciona a la página principal de la aplicación.

2 Reseteo de filtros
Al seleccionar esta opción se quitan todos los filtros de datos.

3 Filtrado de datos
En esta sección se encuentran los filtros que se pueden aplicar.

4 Edición
Al seleccionar esta opción se abre una ventana en la que se muestran los datos del rol para editar.

5 Activación / Desactivación
Al seleccionar esta opción se abre una ventana en la que se muestra un mensaje de confirmación de activación o desactivación de un rol.

6 Creación de rol
Al seleccionar esta opción se abre una ventana en la que se muestran los datos que se deben ingresar para crear un nuevo rol.

7 Reporte en PDF
Al seleccionar esta opción se descarga un archivo PDF con los datos de la tabla.

8 Reporte en Excel
Al seleccionar esta opción se descarga un archivo Excel con los datos de la tabla.

Figura F26. Manual de usuario – Pag. 26

EDICIÓN Y ACTIVACIÓN / DESACTIVACIÓN (ROLES)

Para que se actualicen los datos se debe recargar la página.

1 Edición de datos
En esta sección se encuentran los datos para que puedan ser editados.

2 Cancelación de edición
Al seleccionar esta opción se cierra la ventana y no se edita ningún dato.

3 Confirmación de edición
Al seleccionar esta opción se cierra la ventana y se guardan los datos editados.

4 Cancelación de activación / desactivación
Al seleccionar esta opción se cierra la ventana y no se activa o desactiva el rol.

5 Confirmación de activación / desactivación
Al seleccionar esta opción se cierra la ventana y se activa o desactiva el rol.

Figura F27. Manual de usuario – Pag. 27

CREACIÓN (ROL)

1 Ingreso de datos
En esta sección se encuentran cada uno de los datos que deben ser ingresados para crear un nuevo rol.

2 Cancelación de creación
Al seleccionar esta opción se cierra la ventana y no se guarda el nuevo rol.

3 Confirmación de creación
Al seleccionar esta opción se cierra la ventana y se guarda el nuevo rol.

1 Ingreso de datos
En esta sección se encuentran cada uno de los datos que deben ser ingresados para crear un nuevo rol.

2 Cancelación de creación
Al seleccionar esta opción se cierra la ventana y no se guarda el nuevo rol.

3 Confirmación de creación
Al seleccionar esta opción se cierra la ventana y se guarda el nuevo rol.

Figura F28. Manual de usuario – Pag. 28



Figura F29. Manual de usuario – Pag. 29



Figura F30. Manual de usuario – Pag. 30

EJEMPLO DE REPORTE EN EXCEL

	A	B	C	D	E
	Numero Turno	Estado	Fecha	Motivo de Consulta	Sucursal
2	CA1	ATENDIDO	2023-09-27	CAMBIOS	CENTRO
3	QU1	ATENDIDO	2023-09-27	QUEJAS Y RECLAMOS	CENTRO
4	CA2	ATENDIDO	2023-09-27	CAMBIOS	CENTRO
5	QU2	ATENDIDO	2023-09-27	QUEJAS Y RECLAMOS	CENTRO
6	QU1	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
7	CO1	ATENDIDO	2023-09-30	CONSULTAS	MATRIZ
8	PE1	ATENDIDO	2023-09-30	PETICION DE SERVICIO	MATRIZ
9	CA1	ATENDIDO	2023-09-30	CAMBIOS	MATRIZ
10	QU2	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
11	CO2	ATENDIDO	2023-09-30	CONSULTAS	MATRIZ
12	CA2	ATENDIDO	2023-09-30	CAMBIOS	MATRIZ
13	QU3	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
14	PE2	ATENDIDO	2023-09-30	PETICION DE SERVICIO	MATRIZ
15	QU4	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
16	PE3	ATENDIDO	2023-09-30	PETICION DE SERVICIO	MATRIZ
17	QU5	ATENDIDO	2023-09-30	QUEJAS Y RECLAMOS	MATRIZ
18	CA3	ATENDIENDO	2023-10-01	CAMBIOS	CENTRO

Figura F31. Manual de usuario – Pag. 31