

**UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE
HUAMANGA**

FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



TESIS:

**Sistema web spa para la gestión de rol de turnos del
personal asistencial, hospital regional de Ayacucho, 2024**

Para optar el título profesional de:
INGENIERO DE SISTEMAS

PRESENTADO POR:
Bach. Cristian POZO ANCHO

ASESOR:
Mg. Ing. Richard ZAPATA CASAVARDE

AYACUCHO - PERÚ
2024

Dedicatoria

*A mi familia, por su cariño y apoyo incondicional
a lo largo de todo mi proceso educativo.*

*A mis amigos y seres cercanos, quienes
contribuyeron de manera desinteresada al logro
de este proyecto.*

Agradecimiento

La culminación de este trabajo no habría sido posible sin el apoyo incondicional y el aliento moral de mi familia: mi abuela, mis padres y mis hermanos, quienes fueron mi mayor fortaleza en los momentos más difíciles.

Expreso mi agradecimiento a la Universidad Nacional de San Cristóbal de Huamanga, que me brindó años de formación profesional y desarrollo personal.

También agradezco a los docentes de la Escuela Profesional de Ingeniería de Sistemas, quienes, con su constante enseñanza, conocimientos y experiencia, me guiaron a lo largo de toda mi etapa universitaria.

RESUMEN

El Hospital Regional de Ayacucho enfrenta desafíos en la gestión eficiente de turnos para su personal operativo, lo que afecta la calidad del servicio y la organización de los recursos humanos. Actualmente, la programación de turnos se realiza de manera manual, lo que conlleva errores, tiempos de espera prolongados y costos adicionales. Con el objetivo de mejorar este proceso, se ha desarrollado un prototipo de aplicación web que permite gestionar los turnos de manera automatizada y eficiente, reduciendo los errores y optimizando los recursos.

El estudio se centró en la implementación de un sistema web para la gestión de roles de turnos en el hospital, con el propósito de reducir los tiempos de programación, disminuir los errores humanos y mejorar la calidad del servicio. Se realizó un estudio de tipo observacional, ya que se intervino solo en la variable denominada gestión de turnos; es prospectivo porque se recopilaron datos en tiempo real a través de encuestas y entrevistas al personal del hospital; es transversal, ya que se midieron las variables en un único momento del tiempo, y es descriptivo, porque se desarrolla un prototipo funcional de la aplicación web.

Para el desarrollo del sistema, se utilizó la metodología ágil Scrum, lo que permitió organizar el trabajo en iteraciones y facilitar la adaptación de los cambios a medida que surgían nuevas necesidades durante el desarrollo. La recolección de datos se realizó a través de entrevistas y observación directa en las áreas de trabajo del hospital.

Los resultados obtenidos incluyen la implementación exitosa de la aplicación web para la gestión de turnos, lo que ha permitido una mejora en la eficiencia del proceso, una reducción significativa de errores y una mayor satisfacción por parte del personal operativo.

Palabras Clave: Scrum, gestión de turnos, hospital, aplicación web, automatización.

ABSTRACT

The Regional Hospital of Ayacucho faces challenges in the efficient management of shifts for its operational staff, which affects the quality of service and the organization of human resources. Currently, shift scheduling is done manually, leading to errors, long wait times, and additional costs. To improve this process, a prototype web application has been developed that allows for the automated and efficient management of shifts, reducing errors and optimizing resources.

The study focused on the implementation of a web-based system for shift role management in the hospital, with the aim of reducing scheduling times, decreasing human errors, and improving service quality. An observational study was conducted, as the intervention was focused only on the variable called shift management; it is prospective because real-time data were collected through surveys and interviews with hospital staff; it is cross-sectional, as the variables were measured at a single point in time; and it is descriptive, as a functional prototype of the web application was developed.

For the system's development, the agile Scrum methodology was used, which allowed the work to be organized in iterations and facilitated the adaptation to changes as new needs emerged during development. Data collection was carried out through interviews and direct observation in the hospital's work areas.

The results include the successful implementation of the web application for shift management, which has led to improvements in process efficiency, a significant reduction in errors, and greater satisfaction from the operational staff.

Palabras Clave: Scrum, shift management, hospital, web application, automation.

Índice

Resumen	iv
Abstract	v
CAPÍTULO I PLANTEAMIENTO DEL PROBLEMA	15
1.1 PLANTEAMIENTO DEL PROBLEMA	15
1.2 FORMULACIÓN DEL PROBLEMA.....	16
1.2.1 Problema general.....	16
1.2.2 Problemas específicos	16
1.3 JUSTIFICACIÓN DE LA INVESTIGACIÓN	16
1.4 LIMITACIONES DE LA INVESTIGACIÓN	17
1.5 OBJETIVOS	18
1.5.1 Objetivo general.....	18
1.5.2 Objetivos específicos.....	18
CAPÍTULO II MARCO TEÓRICO	19
2.1 ANTECEDENTES DE LA INVESTIGACIÓN.....	19
2.2 MARCO CONCEPTUAL	23
2.2.1 Aplicación Web.....	23
2.3 MARCO REFERENCIAL.....	24
2.3.1 Modelo vista controlador (MVC).....	24
2.3.2 Base de datos.....	25
2.3.3 Sistema de Gestión de Base de Datos (SGBD)	26
2.3.4 SCRUM.....	26

2.3.5	TEAM FOUNDATION SERVICES	28
2.3.6	SQL SERVER.....	28
2.3.7	VISUAL STUDIO.....	29
2.3.8	VISUAL STUDIO CODE.....	29
2.3.9	JIRA	29
2.3.10	KANBAN	29
2.3.11	NET CORE.....	30
2.3.12	PRUEBAS UNITARIAS.....	30
2.3.13	COBERTURA DE CODIGO	31
2.3.14	WEB API.....	31
2.3.15	POSTMAN	31
2.3.16	FRAMEWORK VUE JS	32
2.3.17	Marco de Trabajo Scrum	33
	CAPÍTULO III MATERIAL Y MÉTODOS.....	35
3.1	TIPO Y NIVEL DE INVESTIGACIÓN	35
3.1.1	Tipo de investigación	35
3.1.2	Nivel de investigación.....	36
3.2	DISEÑO DE LA INVESTIGACIÓN	36
3.3	VARIABLES	36
3.3.1	Definición conceptual de variables	36
3.3.2	Definición operacional de variables.....	37

3.4	POBLACIÓN Y MUESTRA.....	38
3.4.1	Población.....	38
3.4.2	Muestra.....	38
3.5	OPERACIONALIZACIÓN DE LAS VARIABLES.....	38
3.6	TÉCNICAS E INSTRUMENTOS DE LA INVESTIGACIÓN	39
3.6.1	Técnicas.....	39
3.6.2	Instrumentos	40
3.7	PROCEDIMIENTOS.....	40
3.7.1	Estrategia de prueba de hipótesis	40
3.7.2	Técnicas de procesamiento de datos	40
3.7.3	Diseño estadístico.....	44
	CAPÍTULO IV RESULTADOS Y DISCUSIÓN	45
4.1	ENTREGABLES SCRUM	45
4.1.1	ROLES CENTRALES	45
4.1.2	HISTORIAS DE USUARIO	45
4.1.3	PILA DEL PRODUCTO.....	73
4.1.4	ITERACIONES DEL PROYECTO.....	75
4.1.5	TAREAS DE LAS ITERACIONES	78
4.1.6	GRÁFICO DE TRABAJO PENDIENTE	87
4.2	ENTREGABLES SCRUM EN EL CICLO DE DESARROLLO	91
4.2.1	EJECUCIÓN DE ITERACIONES.....	91

4.2.2	SCRUM DIARIO	136
4.2.3	GRÁFICAS DE TRABAJO PENDIENTE	138
	CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES	141
5.1	CONCLUSIONES	141
5.2	RECOMENDACIONES	143
	Referencias Bibliográficas	144
	Anexos	147

Lista de Tablas

Tabla 1. Operacionalización de variables	33
Tabla 2. Entregables SCRUM	35
Tabla 3. Entregables SCRUM en el ciclo de desarrollo	36
Tabla 4. Herramientas para el procesamiento de datos	37
Tabla 5. Historias de usuario de la aplicación Web	39
Tabla 6. Pila del producto con valoración Plannig Poker.	54
Tabla 7. Iteraciones del proyecto.	56
Tabla 8. Tareas de la iteración I: Autenticación, Seguridad y Pantalla Principal	57
Tabla 9. Tareas de la iteración II: Gestión de Roles y Permisos	58
Tabla 10. Tareas de la iteración III: Gestión de Empleados	59
Tabla 11. Tareas de la iteración IV: Gestión de la Estructura Organizativa y Turnos	60
Tabla 12. Totalidad de horas para las iteraciones	61
Tabla 13. Estimación de tiempo para el desarrollo del equipo de trabajo	61
Tabla 14 Tiempo del desarrollo del proyecto en días	62

Lista de Figuras

Figura 1: Grafico de trabajo pendiente optima del proyecto.	62
Figura 2: Grafico de trabajo pendiente de Iteración I	63
Figura 3: Grafico de trabajo pendiente de Iteración II	63
Figura 4: Grafico de trabajo pendiente de Iteración III	64
Figura 5: Grafico de trabajo pendiente de Iteración VI	64
Figura 6: Estructuración de directorios para la aplicación web y plantilla general	65
Figura 7: Validación de credenciales de usuario	65
Figura 8: Interfaz de error por credenciales incorrectas	66
Figura 9: Mensaje de error desplegado en la pantalla de inicio de sesión	67
Figura 10: Interfaz de recuperación de contraseña	67
Figura 11: Envío de correo electrónico para la recuperación de contraseña	68
Figura 12: Funcionalidad de cierre de sesión	69
Figura 13: Expiración automática de sesión	69
Figura 14: Lógica de bloqueo de cuenta por intentos fallidos	69
Figura 15: Interfaz de pantalla principal del sistema	70
Figura 16: Interfaz de pantalla principal del sistema	70
Figura 17: Diseño de la interfaz del menú de navegación	71
Figura 18: Implementación de la navegación del menú	71
Figura 19: Diseño de la interfaz del calendario de turnos	72
Figura 20: Visualización del calendario de turnos en la pantalla principal	72
Figura 21: Indicador de sesión activa	73
Figura 22: Diseño de la interfaz para la creación de roles	74
Figura 23: Implementación de la creación de roles	74
Figura 24: Diseño de la interfaz para la asignación de roles	74

Figura 25: Implementación de la asignación de roles	75
Figura 26: Diseño de la interfaz para la modificación de roles	75
Figura 27: Implementación de la modificación de roles	75
Figura 28: Diseño de la interfaz para la eliminación de roles	76
Figura 29: Implementación de la eliminación de roles	76
Figura 30: Diseño de la interfaz para la visualización de roles	77
Figura 31: Implementación de la visualización de roles	77
Figura 32: Diseño de la interfaz para la visualización de roles asignados	78
Figura 33: Implementación de la visualización de roles asignados	78
Figura 34: Diseño de la interfaz para la definición de permisos por rol	79
Figura 35: Implementación de la definición de permisos por rol	79
Figura 36: Diseño de la interfaz para la creación de roles basados en grupos	80
Figura 37: Implementación de la creación de roles basados en grupos	80
Figura 38: Diseño de la interfaz para la asignación de roles por departamento	81
Figura 39: Implementación de la asignación de roles por departamento	81
Figura 40: Diseño de la interfaz para la creación de roles de auditoría	82
Figura 41: Implementación de la creación de roles de auditoría	82
Figura 42: Diseño de la interfaz para la creación de roles para gestión de turnos	83
Figura 43: Implementación de la creación de roles para gestión de turnos	83
Figura 44: Diseño de la interfaz de la grilla de roles	84
Figura 45: Implementación de la grilla de roles	84
Figura 46: Diseño de la interfaz para agregar empleados	85
Figura 47: Implementación para agregar empleados	85
Figura 48: Diseño de la interfaz para asignar roles a empleados	86
Figura 49: Implementación para asignar roles a empleados	86

Figura 50: Diseño de la interfaz para agregar información de contacto del empleado	87
Figura 51: Implementación para agregar información de contacto del empleado	87
Figura 52: Diseño de la interfaz para definir disponibilidad inicial del empleado	88
Figura 53: Implementación para definir disponibilidad inicial del empleado	88
Figura 54: Diseño de la interfaz para asignar empleados a un departamento	89
Figura 55: Implementación para asignar empleados a un departamento	89
Figura 56: Diseño de la interfaz para cargar foto del empleado	90
Figura 57: Implementación para cargar foto del empleado	90
Figura 58: Diseño de la lógica para notificar al empleado sobre su registro	91
Figura 59: Implementación para notificar al empleado sobre su registro	91
Figura 60: Diseño de la interfaz para editar empleados	92
Figura 61: Implementación para editar empleados	92
Figura 62: Diseño de la interfaz para la búsqueda de empleados	93
Figura 63: Implementación para la búsqueda de empleados	93
Figura 64: Diseño de la interfaz de la grilla de empleados	93
Figura 65: Implementación de la grilla de empleados	94
Figura 66: Implementación para la validación de datos del empleado	95
Figura 67: Diseño de la interfaz para la gestión de contratos	95
Figura 68: Implementación para la gestión de contratos	96
Figura 69: Diseño de la interfaz para gestionar áreas del negocio	97
Figura 70: Implementación para gestionar áreas del negocio	97
Figura 71: Diseño de la interfaz para gestionar cargos	98
Figura 72: Implementación para gestionar cargos	99
Figura 73: Diseño de la interfaz para la gestión de tipos de turno	99

Figura 74: Implementación para la gestión de tipos de turno	100
Figura 75: Diseño de la interfaz para la gestión de profesiones	100
Figura 76: Implementación para la gestión de profesiones	101
Figura 77: Diseño de la interfaz para programar turnos	101
Figura 78: Implementación para programar turnos	102
Figura 79: Diseño de la interfaz para gestionar turnos	103
Figura 80: Implementación para gestionar turnos	103
Figura 81: Diseño de la lógica para solicitar aprobación de turnos	104
Figura 82: Implementación para solicitar aprobación de turnos	104
Figura 83: Diseño de la interfaz para visualizar la programación de turnos	105
Figura 84: Implementación para visualizar la programación de turnos	105
Figura 85: Diseño de la lógica para exportar la programación de turnos	105
Figura 86: Implementación para exportar la programación de turnos	106
Figura 87: Diseño de la lógica para aprobar la programación de turnos	106
Figura 88: Implementación para aprobar la programación de turnos	106
Figura 89: Gráfico de trabajo pendiente de iteración I y del proyecto al finalizar la iteración	108
Figura 90: Gráfico de trabajo pendiente de iteración II y del proyecto al finalizar la iteración	109
Figura 91: Gráfico de trabajo pendiente de iteración III y del proyecto al finalizar la iteración	109
Figura 92: Gráfico de trabajo pendiente de iteración IV y del proyecto al finalizar la iteración	110

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

El Hospital Regional de Ayacucho enfrenta desafíos significativos en la gestión y asignación del rol de turnos del personal asistencial, que incluye médicos, enfermeras y otros profesionales de la salud. Actualmente, este proceso se lleva a cabo de manera manual, comenzando aproximadamente el día 15 del mes anterior, lo que implica coordinar la disponibilidad del personal y la demanda de servicios de manera física y burocrática. Este método es altamente ineficiente, ya que consume tiempo valioso del personal administrativo, genera errores y no permite una reprogramación o ajuste oportuno en caso de cambios imprevistos, como emergencias médicas o licencias. Además, la duplicidad de roles y la falta de un sistema centralizado que permita una visión global de los roles de turnos afectan directamente la calidad de la atención y la eficiencia operativa del hospital.

El flujo de información actual, que pasa desde los servicios médicos hasta la dirección del hospital a través de múltiples departamentos, es lento y propenso a errores, lo que retrasa la aprobación y publicación de los roles de turnos. Esto es especialmente problemático en un entorno donde la demanda de servicios es alta y la disponibilidad del personal es crítica. La falta de automatización no solo afecta la asignación de recursos humanos en momentos cruciales, sino que también genera una carga administrativa innecesaria, dificultando la programación de atenciones y el cálculo de pagos al personal asistencial.

Ante estos desafíos, es crucial implementar un sistema web SPA que permita gestionar de manera eficiente la asignación y control del rol de turnos del personal

asistencial. Este sistema no solo agilizará el proceso, sino que también mejorará la precisión, reducirá la carga administrativa y pretenderá que los roles de turnos sean aprobados y publicados al menos cinco días antes del inicio de cada mes, asegurando así una mejor planificación y atención en el hospital.

1.2 FORMULACIÓN DEL PROBLEMA

1.2.1 Problema general

¿Cómo automatizar la **gestión de rol de turnos** del personal asistencial en el Hospital Regional de Ayacucho, 2024?

1.2.2 Problemas específicos

- ¿De qué manera automatizar el **control de rol de turnos** del personal asistencial del Hospital Regional de Ayacucho?
- ¿De qué manera automatizar la **atención de rol de turnos** del personal asistencial del Hospital Regional de Ayacucho?
- ¿Cómo mejorar la **calidad del servicio** al personal asistencial del Hospital Regional de Ayacucho?

1.3 JUSTIFICACIÓN DE LA INVESTIGACIÓN

La implementación de un sistema web SPA para la automatización de la asignación y gestión del rol de turnos en el Hospital Regional de Ayacucho es fundamental para superar las ineficiencias actuales y mejorar la calidad del servicio. El proceso manual actualmente en uso no solo es laborioso y propenso a errores, sino que también retrasa la planificación y afecta negativamente la programación de las atenciones médicas y el cálculo del pago del personal asistencial. Este sistema permitirá automatizar y agilizar la elaboración de los roles de turnos, reduciendo significativamente el tiempo y los recursos necesarios para su

aprobación y publicación.

Además, al centralizar y digitalizar la información, se podrá mejorar la transparencia y precisión en la asignación de los roles de turnos, evitando duplicidades y asegurando que el personal adecuado esté disponible en los momentos críticos. La automatización también permitirá escalar la información con datos del reloj biométrico, facilitando auditorías y garantizando que el personal esté cumpliendo con sus roles de turnos asignados. La reducción de la cantidad de hojas impresas y el manejo digital de la información no solo contribuirán a un ambiente más sostenible, sino que también romperán con la burocracia existente, permitiendo un flujo de información más eficiente y una mayor flexibilidad en la gestión de roles de turnos.

Se propone automatizar la gestión de rol de turno del personal asistencial del Hospital Regional de Ayacucho con las siguientes variables asociadas a la gestión de rol de turno: control de rol de turnos, atención de rol de turnos, calidad de servicio.

1.4 LIMITACIONES DE LA INVESTIGACIÓN

Las limitaciones de la implementación de este sistema radican en la capacidad de la infraestructura tecnológica actual del Hospital Regional de Ayacucho. El estudio se enfoca en variables clave relacionadas con la asignación y gestión del rol de turnos, como la disponibilidad del personal, la reprogramación de roles de turnos y la cobertura de roles de turnos críticos, ya que son las únicas que se pueden monitorear con la tecnología disponible sin incurrir en costos adicionales. Esta restricción podría dejar fuera otros factores relevantes, como la integración con sistemas externos o la evaluación de la satisfacción del personal, que no pueden ser abordados con las herramientas disponibles en la versión básica del sistema. Estas limitaciones podrían influir en la capacidad del sistema para proporcionar una solución completamente integral, aunque la automatización de los aspectos clave seguirá

proporcionando mejoras significativas en la eficiencia operativa y en la calidad del servicio.

1.5 OBJETIVOS

1.5.1 Objetivo general

Desarrollar un sistema web SPA para la **gestión de rol de turnos** del personal asistencial en el Hospital Regional de Ayacucho

1.5.2 Objetivos específicos

- a. Explorar, planificar e iterar para automatizar el **control de rol de turnos** del personal asistencial del Hospital Regional de Ayacucho, 2024.
- b. Explorar, planificar e iterar para automatizar la **atención de rol de turnos** del personal asistencial del Hospital Regional de Ayacucho, 2024.
- c. Explorar, planificar e iterar para brindar **calidad de servicio** al personal asistencial del Hospital Regional de Ayacucho, 2024.

CAPÍTULO II

MARCO TEÓRICO

2.1 ANTECEDENTES DE LA INVESTIGACIÓN

2.1.1 Antecedentes Internacionales

Delgado Caicedo, L. M. (2019). “Sistema Clínico Integral Para Atención Médica Y Hospitalaria Que Maneje Integración E Interoperabilidad En Base A Un Bus De Servicios”. Tesis presentada en la Pontificia Universidad Católica del Ecuador, Ecuador, utilizando una investigación observacional con un enfoque cuantitativo, llegando a la siguiente conclusión. El sistema web posee funcionalidades como el registro de usuarios, agendamiento de citas médicas, generación de reporte de citas a cargo de cada médico, también cuenta con un módulo de horarios que permite al médico registrar las horas de su turno asignado; toda esta información está almacenada en una base de datos que puede ser consultada dependiendo de su rol en la institución. La gestión del personal dentro de una organización sanitaria, involucrando funcionalidades como el registro de usuarios, se convirtieron en integraciones que buscan optimizar la eficacia y el alcance de un proceso en un contexto hospitalario.

Llamba Pelaez, J. R. y Imbaquingo Tocagón, L. R. (2018). “*Sistema web para la automatización de turnos e historias clínicas de la Fundación Clínica Mosquera*”. Tesis presentada en la Universidad Tecnológica Israel, Ecuador, utilizando la investigación de campo con un enfoque cuantitativo, llegando a la siguiente conclusión. La implantación del sistema web en la Fundación Clínica Mosquera, proporciona una herramienta tecnológica que automatiza los procesos de generación de turnos e historias clínicas, además, permite generar reportes en apoyo a la toma de decisiones; es de suma importancia capacitar a los usuarios en el manejo del sistema para mejorar la calidad de atención en la clínica. La implementación de un sistema web que automatiza tanto la generación de turnos como el manejo de historias clínicas proporciona una base sólida para entender cómo la tecnología

puede optimizar los procesos hospitalarios. Además, se toma importancia sobre la capacitación de los usuarios, un aspecto crucial para la optimización del sistema.

Montenegro, A. A. (2020). *“Desarrollo de un Sistema Prototipo de Gestión de Turnos Mediante Teléfonos Móviles Con Sistema Android Para El Departamento Médico De La Escuela Politécnica Nacional”*. Tesis presentada en la Escuela Politécnica Nacional, Ecuador, aplicando una investigación aplicada, llegó a la siguiente conclusión. La implementación del prototipo de gestión de turnos mediante teléfonos móvil, logra automatizar el proceso de asignación de turnos médicos, gracias al módulo Consulta de turnos que almacena turnos anteriores, lo que permite tener mejor gestión de las asignaciones para el futuro. La importancia de la portabilidad y disponibilidad de un sistema, son factores cruciales y relevantes para una actualización en tiempo real, comodidad y una buena gestión de turnos en una organización donde es vital la organización del tiempo del personal.

Ramos Rosales, E. C. R. et al. (2019). *“Diseño e implementación de un sistema integral de control de turnos. Artículo de investigación publicada por la editorial 3C Empresa: Investigación y pensamiento crítico”*, llegaron a la siguiente conclusión. La implantación del sistema integral de control de turnos en la empresa es una herramienta que agrega valor en sus trámites y servicios ya que permite mejor fluidez en la atención a la vez que permite a los usuarios generar reportes para ayudar en la toma de decisiones. La importancia de los sistemas que gestionan los turnos, además de una importante eficacia operativa, también tiene intervención estratégica a nivel institucional o departamental, la cual brinda una considerable información para la mejora de este proceso y otros.

Vega Carrillo, I. M. (2018). *“Automatización del sistema de agendación de turnos mediante aplicación web para el Hospital Básico Antonio Ante”*. Tesis presentada en la Pontificia Universidad Católica del Ecuador, Ecuador, utilizando una investigación

observacional con enfoque cualitativo, llegando a la siguiente conclusión. Al implantarse la aplicación web permitirá al hospital automatizar el proceso de gestión de citas para los pacientes y la agendación de turnos del personal administrativo y médicos. Así mismo el almacenamiento de la información en bases de datos digitales facilitará a la institución la toma de decisiones mediante la generación de reportes especializados, en específico en el tema de asignación de roles a los médicos; a pesar de que se está implementando este sistema informático, aún falta automatizar muchos procesos en la institución. La implementación de este sistema da el inicio o un precedente hacia la automatización de procesos, además de la digitalización de la institución, temas importantes hacia el camino del futuro de la empresa; mostrando la importancia de adaptarse y renovar los procesos de la empresa.

2.1.2 Antecedentes Nacionales

Herrera Bravo, J. W. y Veliz Landázuri, A. P. (2020). *“Desarrollo de una aplicación web para la mejora de procesos del área de atención integral del niño en el centro de salud José Leonardo Ortiz, distrito de José Leonardo Ortiz, provincia de Chiclayo, departamento de Lambayeque”*. Tesis presentada en la Universidad Nacional Pedro Ruíz Gallo, Perú, utilizando una investigación de tipo tecnológico, de nivel correlacional – causal, con un enfoque cuantitativo, llegando a la siguiente conclusión. El aplicativo web ZURIXER automatizó procesos de registro, modificación y consulta de historial clínico ahorrando tiempos de atención, el aplicativo también es una herramienta que ayuda en la toma de decisiones gracias a la información almacenada en la base de datos que puede ser consultada por los médicos y administrativos. La importancia del almacenamiento de la información para su tratamiento, todo con el fin de buscar una mejora en la toma de decisiones y optimización de los procesos operativos.

Huamani, J. M. (2019). *“Desarrollo de un sistema de control del personal asistencial para mejorar el proceso de atención de los pacientes en el Hospital Víctor Ramos*

Guardia de Huaraz”. Tesis presentada en la Universidad Nacional Santiago Túnez de Mayolo, Perú, utilizando una investigación aplicada, descriptiva, no experimental, llegando a la siguiente conclusión. La implementación del sistema de control del personal asistencial resultó satisfactoria para los usuarios del área de persona, permitiéndoles programar los horarios laborales mensuales. Los precedentes a implementaciones exitosas de sistemas de control de personal, sirven como base y justificación para la implementación de estos tipos de sistemas, además de servir de inicio para la incentivación a la automatización de procesos en instituciones.

Ortíz Cárdenas, F. y Condori Velásquez, J. M. (2019). “*Sistema de Gestión y Control de Disponibilidad para el personal militar que labora en la Dirección Gubernamental Peruana*”. Tesis presentada en la Universidad Peruana de Ciencias Aplicadas, Perú, utilizando una investigación de campo con un enfoque mixto, llegando a la siguiente conclusión. La implantación del sistema permitió reducir el tiempo en la elaboración del patrón de rol de guardias, utilizando algoritmos expertos basados en reglas y restricciones, también permite generar una lista de disponibilidad del personal militar para asignar turnos de guardia, reduciendo los errores en la programación de turnos. El manejo del tiempo es importante en toda institución que trabaje con personal con turnos, además los sistemas optimizan mucho la distribución del tiempo en turnos con algoritmos o distribuciones inteligentes, además de facilitar la conclusión de los distintos subprocesos que conllevan estas tareas por eso al implementar estos tipos de sistemas se maximizan mucho los usos de los tiempos.

Rabanal Rodríguez, R. H. (2021). “*Sistema Web para la Gestión de Programación de Turnos del Personal Asistencial en el Hospital Alta Complejidad Virgen de la Puerta utilizando Iconix*”. Tesis presentada en la Universidad César Vallejo, Perú, utilizando una investigación aplicada, experimental puro, llegando a la siguiente conclusión. Al

implementar el sistema web haciendo uso de la metodología ICONIX, se comprueba que mejora la gestión en la programación asistencial, disminuye el tiempo, costo y los errores en la programación de turnos en el Hospital Alta Complejidad Virgen de la Puerta. Los sistemas automatizados de procesos ofrecen beneficios operativos, pero también ofrecen beneficios económicos ya que se pueden reducir los gastos en administración y gastos en reparación de los errores humanos.

Silva Mejía, D. J. (2022). *“Programación de roles médicos y la atención en consultas externas de un hospital nacional nivel III-1 Región Lambayeque”*. Tesis presentada en la Universidad César Vallejo, Perú, utilizando una investigación aplicada, no experimental con un enfoque cuantitativo, llegando a la siguiente conclusión. Existe una relación positiva entre la programación de roles médicos y la atención de consultas, quiere decir, que si hay una buena programación habrá una buena atención en las consultas externas evitando la espera e incomodidad de los usuarios. La importancia de un correcto desarrollo de la aplicación es uno de los principales objetivos a considerar en los distintos trabajos de este tipo, ya que al no realizar un buen sistema no se aborda bien las tareas que arrastran este tipo de procesos conllevando errores y pérdidas de recursos y tiempo.

2.2 MARCO CONCEPTUAL

2.2.1 Aplicación Web

En palabras de (Valarezo et al., 2018) la aplicación web es un programa informático o sitio web que se ejecuta directamente en el internet haciendo uso de navegadores sin la necesidad de ser instalado en el dispositivo electrónico.

En palabras de (Lujan, 2002) la aplicación web no necesita ser implementado por el programador ya que a diferencia de las aplicaciones estándar los componentes orientados al cliente, el servidor y el protocolo mediante el que se comunican ya están

estandarizados.

A. Nivel de funcionalidad

En palabras de (Aizprua et al., 2019) la funcionalidad se evalúa en base a sub- atributos como la idoneidad, corrección, inter-operatividad, conformidad y seguridad. Para una aplicación web se relaciona al nivel de satisfacción de los atributos mencionados para satisfacer las necesidades del usuario.

B. Nivel de fiabilidad

De acuerdo a (Aizprua et al., 2019) la confiabilidad en una aplicación web se define como aquella capacidad que le permite mantener su rendimiento en diferentes circunstancias por un periodo de tiempo, capacidad de tolerancia a fallos y la facilidad de recuperación.

C. Nivel de usabilidad

Según (Aizprua et al., 2019) la usabilidad es la característica de la aplicación que permite ser más fácil de usar para de los usuarios, la facilidad de comprensión, aprendizaje y operatividad son algunos atributos con los que podemos medir la usabilidad.

D. Nivel de eficiencia

La eficiencia es la capacidad que permite a la aplicación optimizar el empleo de los recursos del sistema. Los sub-atributos como el tiempo de uso y los recursos utilizados bajo ciertas circunstancias son los que facilitan medirlos para demostrar su utilidad (Aizprua et al., 2019).

2.3 MARCO REFERENCIAL

2.3.1 Modelo vista controlador (MVC)

La arquitectura de software determina reglas y el marco bajo el cual se

desarrollará el software, existen muchos patrones entre ellos el patrón modelo-vista-controlador (MVC) (Santiago et al., 2021).

El patrón de arquitectura Modelo-Vista-Controlador permite distribuir responsabilidades entre las distintas capas de aplicación facilitando su implementación, es decir, separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes (Orozco, 2019).

El modelo MVC ofrece los motores de vista aspx y razor que brindan una gran versatilidad al momento de personalizar la capa del lado del cliente (vistas) (Santiago et al., 2021).

A. Modelo

Interactúa directamente con los datos, contiene submódulos o clases que permiten acceder, mostrar o refrescar la información (Alonso, 2019). En palabras sencillas, es el backend de la aplicación.

B. Vista

Se encargan de los módulos responsables de materializar las interfaces que se mostraran al usuario (Alonso, 2019). Es decir, el frontend de la aplicación. Generalmente son conocidos como Interfaz Gráfica de Usuario (GUI).

C. Controlador

El controlador es un enlace imprescindible sirviendo de canal bidireccional que conecta al modelo con la vista, se encarga de dar respuesta a la comunicación (Alonso, 2019). Redirige una petición ya sea al modelo o a la vista dependiendo de su naturaleza.

2.3.2 Base de datos

Es una colección de datos que se muestran de forma organizada para ser accedidos por un programa y obtenga los datos que necesite, se encuentran

almacenados en ficheros digitales relacionados entre sí gracias a una determinada estructura lógica (Daqui, 2018).

A. Base de Datos Relacionales

Es un repositorio compartido de datos. Permiten establecer interconexiones entre los datos, y a través de dichas conexiones relacionar los datos de ambas tablas, de ahí proviene el nombre de Modelo Relacional este tipo de base de datos es el más utilizada en la actualidad para implementar aplicaciones. (Alexis,2019)

B. Base de Datos No Relacionales

Están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Son ampliamente reconocidas porque son fáciles de desarrollar, tanto en funcionalidad como en rendimiento a escala. Usan una variedad de modelos de datos, que incluyen documentos, gráficos, clave-valor, en-memoria y búsqueda. (Alexis,2019)

2.3.3 Sistema de Gestión de Base de Datos (SGBD)

Un SGBD es un conjunto de programas cuya función es permitir al usuario gestionar (manipular, almacenar y consultar) uno o varios ficheros organizados dentro de las bases de datos (Daqui, 2018).

2.3.4 SCRUM

“Scrum es un marco ligero que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptables para problemas complejos”.(Schwaber et al., 2020).

El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso. Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e

interacciones entre ellos.

A. PLANNING

Reunión en la que el Product Owner establece las tareas prioritarias, las historias de usuario que se desarrollaran durante el sprint.

Participa todo el equipo scrum y marca el inicio de un sprint.

El equipo de desarrollo y el scrum master realiza consulta sobre todas las dudas que se tienen con el fin de estimar el tiempo que tomará desarrollar cada tarea.

B. DAILY MEETING

Reunión diaria de aproximadamente 15 minutos, en el que el equipo de desarrollo rinde cuentas sobre los avances desarrolladas el día anterior y definir el reto tomado para el resto del día.

En esta reunión se responden tres preguntas.

- ¿Qué he hecho desde la última daily para ayudar al equipo a cumplir su objetivo?

- ¿Qué voy a hacer a partir de este momento para ayudar al equipo a cumplir su objetivo?

- ¿Qué impedimentos tengo o voy a tener que nos impidan conseguir nuestro objetivo?

C. REVIEW

“Reunión que se celebra con el propósito de evaluar los resultados que obtuvo el equipo Scrum luego de un sprint. Esta permite, a su vez, analizar el progreso que está teniendo el desarrollo con miras a cumplir con el objetivo establecido” (Fernanda Aguirre, 2021).

Es el evento en el cual se revisan las tareas o sprint que se habían tomado acuerdo durante el sprint planning. Participa todo el equipo scrum, no debe tardar

más de 4 horas.

D. RETROSPECTIVE

“El equipo scrum reflexiona sobre su forma de trabajar. Se identifican fortalezas y puntos débiles, para afianzar las primeras y planificar acciones de mejora sobre los segundos” (Palacio Marta, 2022).

Una retrospectiva de sprint es una forma de reunir a los equipos de scrum y planificar las mejores formas de aumentar la calidad y la efectividad con respecto a las personas, las interacciones, los procesos, las herramientas y la definición de hecho de cada persona.

2.3.5 TEAM FOUNDATION SERVICES

“Microsoft TFS se integra con Git, haciendo posible clonar un repositorio, añadir código y crear ramas sin necesidad de ninguna herramienta de terceros. La colaboración entre equipos y el código compartido pueden maximizarse buscando código en todos los proyectos y utilizando filtros, código de vista previa, historial de vistas y otras funciones de mejora de la gestión del código fuente” (Visure, s. f.).

TFS permite que los desarrolladores trabajen mejor juntos, principalmente a través de una herramienta denominada el panel de tareas. Se puede obtener acceso a este panel de tareas desde cualquier explorador moderno que le permita crear paneles personalizados.

2.3.6 SQL SERVER

“Es uno de los principales sistemas de gestión de bases de datos relacional del mercado que presta servicio a un amplio abanico de aplicaciones de software destinadas a la inteligencia empresarial y análisis sobre entornos corporativos” (Intelequia, s. f.).

Es un sistema cuya función es la administración de la información de una

determinada plataforma. De este modo dicha información está disponible para el servidor y para los clientes.

2.3.7 VISUAL STUDIO

“Plataforma de lanzamiento creativa que puede utilizar para editar, depurar y compilar código y, finalmente, publicar una aplicación. Además del editor y depurador estándar que ofrecen la mayoría de IDE, Visual Studio incluye compiladores, herramientas de completado de código, diseñadores gráficos y muchas más funciones para mejorar el proceso de desarrollo de software” (microsoft, 2023).

Cómo lo menciona la propia plataforma de Microsoft, Visual Studio brinda al desarrollador un editor avanzado con sugerencias y autocompletado, esto favorece un desarrollo legible y con menor cantidad de errores.

2.3.8 VISUAL STUDIO CODE

Editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C ++, C #, Java, Python, PHP, Go, .NET).

2.3.9 JIRA

Jira es una aplicación web que con el tiempo se ha convertido en el estándar del mercado en las áreas de gestión de proyectos, gestión de tareas y gestión de errores. Especialmente para el desarrollo de software.

Jira nos permite realizar la planificación de nuestro proyecto con gran facilidad y ligereza teniendo cómo gran fortaleza el tablero Kanban que muestra las tareas, las personas encargadas y la gestión del trabajo.

2.3.10 KANBAN

“Con el Método Kanban, se visualiza el trabajo y cómo se mueve a través de un

flujo de trabajo. Esto te ayuda a gestionar de manera eficiente tu negocio, incluyendo la comprensión y gestión de los riesgos en la entrega de servicios a los clientes”(Kanban University, 2021).

La herramienta Kanban nos permite visualizar de forma sencilla los estados del trabajo mediante un tablero colorido y amistoso. En Kanban se usan las tarjetas, estas nos permiten mantener conocer con certeza el flujo de trabajo de todo el equipo y nos permite una rápida sincronización.

2.3.11 NET CORE

“Es la plataforma de desarrollo de Microsoft más moderna, de código fuente abierto, multiplataforma y de alto rendimiento para la creación de todo tipo de aplicaciones” (Díaz Alcotela, 2020).

“Está diseñado para permitir que los componentes en tiempo de ejecución, las API, los compiladores y los lenguajes evolucionen rápidamente, a la vez que proporciona una plataforma estable y compatible para mantener las aplicaciones en ejecución” (Microsoft Learn, s. f.).

2.3.12 PRUEBAS UNITARIAS

Las pruebas unitarias consisten en aislar una parte del código y comprobar que funciona a la perfección. Son pequeños tests que validan el comportamiento de un objeto y la lógica.

Hace referencia a los procesos de validación del funcionamiento del software de un determinado programa o aplicación. Este mecanismo se caracteriza por proporcionar una garantía de calidad del sistema, para lo que utiliza recursos como las pruebas unitarias de software, que se encargan de comprobar que un fragmento del código fuente esté funcionando de forma adecuada.

2.3.13 COBERTURA DE CODIGO

“La supervisión de las métricas de cobertura del código ayuda a garantizar que se mantiene un nivel suficiente de pruebas automatizadas. Si la cobertura del código comienza a disminuir, podría ser una señal de que no está tratando las pruebas automatizadas como un elemento esencial de la escritura de nuevo código” (JetBrains, 2022).

En resumen, podríamos indicar que la cobertura de código es un proceso que busca verificar el correcto funcionamiento de un determinado código o aplicación. Mediante este mecanismo podemos llamar a una función o parte de código para comparar el resultado que esperamos con el resultado que regresa dicha función.

Si ambos resultados son iguales se dice que el código cumple su función. Esta misma medida se debe realizar en todo el código del aplicativo o sistema desarrollado.

2.3.14 WEB API

“Las API son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos. Por ejemplo, el sistema de software del instituto de meteorología contiene datos meteorológicos diarios. La aplicación meteorológica de su teléfono “habla” con este sistema a través de las API y le muestra las actualizaciones meteorológicas diarias en su teléfono” (aws, s. f.).

Las API le permiten al usuario final utilizar una aplicación, software o incluso una simple hoja de cálculo, consultando, cambiando y almacenando datos de diferentes sistemas, sin que el usuario tenga que ingresar a ellos, directamente.

2.3.15 POSTMAN

“Postman es una plataforma API para crear y usar API. Postman simplifica cada paso del ciclo de vida de la API y optimiza la colaboración para que pueda crear mejores

API más rápido” (Plataforma Postman, s. f.).

Postman es una plataforma que permite y hace más sencilla la creación y el uso de APIs. Esta herramienta es muy útil para programar porque da la posibilidad hacer pruebas y comprobar el correcto funcionamiento de los proyectos que realizan los desarrolladores web.

2.3.16 FRAMEWORK VUE JS

Vue.js es un framework de JavaScript de código abierto para la construcción de interfaces de usuario y aplicaciones de una sola página. Fue creado por Evan You, y es mantenido por él y por el resto de los miembros activos del equipo central que provienen de diversas empresas como Netlify y Netguru. Vue.js cuenta con una arquitectura de adaptación gradual que se centra en la representación declarativa y la composición de componentes. La biblioteca central se centra sólo en la capa de vista. Las características avanzadas necesarias para aplicaciones complejas como el enrutamiento, la gestión de estados y las herramientas de construcción se ofrecen a través de librerías y paquetes de apoyo mantenidos oficialmente. Vue.js permite extender el HTML con atributos HTML llamados directivas. Las directivas ofrecen funcionalidad a las aplicaciones HTML, y vienen como directivas incorporadas o definidas por el usuario.

A. COMPONENTES

Los componentes Vue extienden los elementos básicos de HTML para encapsular el código reutilizable. En un nivel alto, los componentes son elementos personalizados a los que el compilador de Vue adjunta comportamiento. En Vue, un componente es esencialmente una instancia de Vue con opciones predefinidas.

B. TEMPLATES

Vue utiliza una sintaxis de plantilla basada en HTML que permite vincular el

DOM renderizado a los datos de la instancia subyacente de Vue. Todas las plantillas de Vue son HTML válido que puede ser analizado por navegadores que cumplan con las especificaciones y por analizadores HTML. Vue compila las plantillas en funciones de representación de DOM virtual. Un Modelo de Objeto de Documento virtual (o "DOM") permite a Vue renderizar componentes en su memoria antes de actualizar el navegador. Combinado con el sistema de reactividad, Vue es capaz de calcular el número mínimo de componentes a ser renderizados y aplicar la cantidad mínima de manipulaciones DOM cuando el estado de la aplicación cambia.

2.3.17 Marco de Trabajo Scrum

Scrum como marco de trabajo permite que cada equipo de desarrollo defina libremente su propia forma de trabajar, de modo que puedan ofrecer el mayor valor posible al final de cada iteración. (Marcal, et al., 2007).

Scrum se ha transformado en el marco ágil más utilizado en la implementación de software en el mundo. Uno de los beneficios de Scrum es ofrecer incrementos de entregables en poco tiempo para mejorar el aprendizaje de los clientes y del mercado, mejorando así los resultados comerciales a través de la innovación de productos. Scrum promueve la auto organización, el trabajo en equipo, la mejora continua y la entrega de valiosos productos incrementales en cada Sprint. Scrum controla el proceso en función de la experiencia y ayuda a gestionar el desarrollo de productos o servicios. Al usar el empirismo, aceptamos el hecho de que las cosas pueden cambiar en cualquier momento. La mejor manera de enfrentar estos cambios es ayudar reducir los riesgos comerciales mediante el uso de la transparencia, la inspección y la adaptación. (Francia, 2020).

De acuerdo a Kent et al. (2001), Scrum se basa en los principios del manifiesto ágil: priorizar el valor de las personas en lugar del valor del proceso; entregar software funcional lo antes posible; atender y responder a los cambios; fortalecer la comunicación y la

colaboración; la comunicación verbal directa entre las personas involucradas en el proyecto; simplicidad; eliminar la gestión del proyecto Artefactos innecesarios.

Para Schwaber y Sutherland (2011), Scrum es un marco de procesos que se ha utilizado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o tecnología para la construcción de productos; por el contrario, es un marco en el que se pueden utilizar varios procesos y tecnologías. Scrum aclara la efectividad relativa de sus prácticas de gestión de productos y desarrollo para que pueda mejorarlas.

CAPÍTULO III

MATERIAL Y MÉTODOS

3.1 TIPO Y NIVEL DE INVESTIGACIÓN

3.1.1 Tipo de investigación

Según Díaz et al. (2015), una investigación observacional se caracteriza por la recolección de datos sin intervención directa del investigador, lo que permite obtener una visión precisa de los eventos tal como ocurren en su entorno natural. En este caso, el estudio es de tipo observacional porque se analiza cómo el personal del Hospital Regional de Ayacucho gestiona actualmente los turnos sin modificar su comportamiento durante el proceso de recolección de datos.

Además, Prieto y Contreras (2018) mencionan que una investigación prospectiva implica la recopilación de datos anticipados para observar el desarrollo de un fenómeno a lo largo del tiempo. Este estudio es prospectivo, ya que se obtendrán datos en tiempo real relacionados con la asignación de turnos del personal de salud, los cuales servirán como base para desarrollar la aplicación web propuesta.

Por otro lado, Hernández y Mendoza (2017) explican que un estudio transversal se realiza en un solo punto en el tiempo, permitiendo una evaluación específica de las variables en un momento dado. En esta investigación, se recolectarán datos en un único momento, lo que permitirá analizar la situación actual del control de turnos en el hospital y proponer mejoras a través de la nueva aplicación.

Finalmente, según Álvarez y Bisquerra (2016), un estudio descriptivo busca detallar las características de un fenómeno sin intervenir directamente en él. Este estudio es descriptivo, ya que se enfoca en analizar cómo se gestionan actualmente los turnos del personal del Hospital Regional de Ayacucho y cómo una aplicación web podría optimizar este proceso.

3.1.2 Nivel de investigación

Según Bravo y Ato (2015), el nivel de investigación descriptivo se utiliza para detallar y caracterizar un fenómeno o situación específica, sin necesidad de intervenir en ella. En este estudio, se ha adoptado un nivel descriptivo porque se busca analizar y documentar de manera detallada el proceso actual de asignación de turnos en el Hospital Regional de Ayacucho, con el fin de proponer una solución a través del desarrollo de una aplicación web que optimice dicho proceso.

3.2 DISEÑO DE LA INVESTIGACIÓN

El diseño de la investigación es no experimental, siguiendo las recomendaciones de Babbie (2016), donde se observa y analiza el fenómeno tal como ocurre en su contexto natural, sin manipular las variables. Esto es adecuado para describir el estado actual de la gestión de turnos en el Hospital Regional de Ayacucho. Este diseño también es prospectivo porque se recopilarán datos de manera anticipada a través de encuestas y entrevistas con el personal del hospital, y es transversal, ya que todas las observaciones y recolección de datos se realizarán en un solo momento, permitiendo una evaluación clara y específica de la situación actual.

3.3 VARIABLES

3.3.1 Definición conceptual de variables

Variable de interés

Gestión de rol de turnos: La gestión de rol de turnos es el proceso mediante el cual se organiza y administra la asignación de horarios y funciones del personal en una institución. Este proceso asegura que el personal operativo cumpla con sus responsabilidades en los tiempos estipulados, optimizando la distribución de recursos humanos y garantizando una cobertura adecuada de los servicios. (MINSa RM242, 2024)

Variables descriptivas (Dimensiones)

Control de rol de turnos: Esta dimensión se refiere a la capacidad de la organización para generar, recibir y distribuir los roles de turno de manera eficiente. Es fundamental para garantizar que todos los turnos estén correctamente asignados y que el personal esté informado de sus responsabilidades. (MINSAs RM242, 2024).

Atención de rol de turnos: Esta dimensión mide la eficiencia temporal en la creación, aprobación y solución de cualquier problema observado en los roles de turno. Es crucial para asegurar que los cambios o ajustes en los turnos se realicen de manera oportuna, minimizando el impacto en las operaciones del hospital. (MINSAs RM242, 2024).

Calidad del servicio: Esta dimensión evalúa la satisfacción de los usuarios con respecto al sistema de gestión de turnos, lo cual es indicativo de la efectividad y aceptabilidad del sistema implementado. (MINSAs RM242, 2024).

3.3.2 Definición operacional de variables

Variable de interés

Gestión de rol de turnos

Variables descriptivas (Dimensiones)

Control de rol de turnos: Involucra la creación, recepción, y distribución de los roles de turnos entre el personal, garantizando que cada miembro esté informado y los turnos estén correctamente asignados.

Atención de rol de turnos: Mide la eficiencia en la gestión de roles, incluyendo el tiempo de creación, aprobación y solución de problemas relacionados con los turnos.

Calidad del servicio: Evalúa la satisfacción del personal con el sistema de gestión de turnos, considerando la facilidad de uso y la equidad en la asignación de turnos.

3.4 POBLACIÓN Y MUESTRA

3.4.1 Población

De acuerdo con Mora y González (2019), la población se define como "el conjunto total de individuos, objetos o elementos que presentan una característica común que es objeto de estudio". En este caso, la población está compuesta por un total de 1,500 miembros del personal asistencial del Hospital Regional de Ayacucho, quienes participan en la gestión y ejecución de roles de turnos en diversas áreas de atención.

3.4.2 Muestra

Según Hernández (2014), una muestra adecuada debe representar de manera precisa la población de estudio para garantizar la validez de los resultados. En este estudio, se seleccionó una muestra representativa de 306 miembros del personal asistencial, calculada con un nivel de confianza del 95% y un margen de error del 5%, utilizando un muestreo probabilístico estratificado. Esta muestra proporcionará una visión precisa y generalizable de los procesos actuales de gestión de turnos en el hospital, así como de la implementación y efectividad del sistema web SPA.

3.5 OPERACIONALIZACIÓN DE LAS VARIABLES

La operacionalización de la variable gestión de rol de turnos se realiza para medir con precisión la variable de interés, a través de las dimensiones identificadas y sus respectivos indicadores. Esto permitirá la formulación de preguntas o ítems en los instrumentos de recolección de datos, necesarios para evaluar y optimizar los procesos de gestión de turnos en el hospital.

Tabla 1

Operacionalización de variables

Variable	Dimensión	Indicador	Pregunta
Ge	Control de	Número de roles	¿Cuántos roles de turno se crean en

stión de	rol de	generados	una semana?
rol de	turnos	Número de roles recibidos	¿Cuántos roles de turno son recibidos y confirmados por el personal operativo?
turnos		Número de roles distribuidos	¿Cuál es la cantidad de roles de turno distribuidos correctamente entre el personal?
		Tiempo de creación de roles	¿Cuánto tiempo toma la creación de un rol de turno desde la solicitud hasta la generación?
	Atención de rol de turnos	Tiempo de aprobación de roles	¿Cuánto tiempo pasa entre la creación y la aprobación del rol de turno?
		Tiempo de solución de roles observados	¿Cuánto tiempo toma resolver cualquier problema identificado en los roles de turno?
	Calidad del servicio	Satisfacción del usuario	¿Qué tan satisfecho está el personal operativo con la gestión de sus turnos?

Fuente: Elaboración propia.

3.6 TÉCNICAS E INSTRUMENTOS DE LA INVESTIGACIÓN

3.6.1 Técnicas

Según Espinoza (2014), la observación y la entrevista son técnicas fundamentales

para la recolección de datos en investigaciones descriptivas. La observación permite obtener información directa sobre el comportamiento del personal en relación con la gestión de turnos, mientras que la entrevista facilita recopilar percepciones, opiniones y sugerencias de los usuarios directos del sistema de roles de turnos en el Hospital Regional de Ayacucho.

En este estudio, se utilizarán ambas técnicas. La observación se enfocará en la operación diaria del sistema de turnos, mientras que las entrevistas se dirigirán tanto a los responsables de la gestión de turnos como al personal operativo.

3.6.2 Instrumentos

Para la observación, se empleará una guía de observación diseñada específicamente para registrar la eficiencia y los problemas observados en la gestión de turnos. Para la entrevista, se utilizará una guía de entrevista semi-estructurada que permitirá explorar en profundidad los desafíos y oportunidades percibidos por los usuarios del sistema actual, así como sus expectativas respecto al nuevo aplicativo web.

3.7 PROCEDIMIENTOS

3.7.1 Estrategia de prueba de hipótesis

En el presente estudio no se considera prueba de hipótesis por ser una investigación de nivel descriptivo que es el desarrollo de un prototipo de software.

3.7.2 Técnicas de procesamiento de datos

A. Técnicas para procesamiento de datos con el marco de trabajo Scrum

Para procesar los datos recopilados a través de entrevistas y observaciones en el contexto de la gestión de turnos en el Hospital Regional de Ayacucho, se utilizará un enfoque basado en el método ágil Scrum. A continuación, se describen las fases principales del proceso:

Tabla 2

Entregables SCRUM

TAREA	ENTREGABLE	TÉCNICA	RESPONSABLE
Asignar roles centrales y no centrales	Lista de roles Scrum	Identificar los responsables de cada rol.	Propietario del producto
Recopilar los requerimientos de dominio	Requerimientos de dominio	Recopilar los requerimientos con entrevista	Propietario del producto y socios.
Elaboración de las historias de usuarios	Historias de usuario	Agrupar en historias de usuarios los requerimientos.	Equipo Scrum
Elaboración de la pila del producto.	Pila del producto	Revisión de las historias de usuario y elaboración de la pila del producto	Equipo Scrum
Definir las Iteraciones y asignarles tareas.	Iteraciones del proyecto	Clasificar las tareas de la pila del producto.	Equipo Scrum
Estimar tiempo de esfuerzo	Gráfico de trabajo pendiente	Calcular la capacidad diaria de desarrollo del	Facilitador

equipo y el
total de horas
de las
iteraciones

Fuente: Ochoa (2018).

Tabla 3

Entregables SCRUM en el ciclo de desarrollo

TAREA	ENTREGABLE	TÉCNICA	RESPONSABLE
Ejecución de cada iteración	Ejecución de la iteración	Desarrollo de cada tarea de los sprint.	Equipo de desarrollo
Reunión diaria de planificación y análisis del avance de la iteración.	Reunión diaria	Reunión del equipo para planificar el trabajo por hacer y analizar el trabajo realizado y problemas encontrados.	Facilitador y Equipo Scrum
Realizar el gráfico de trabajo pendiente de la iteración.	Gráfico de trabajo pendiente	Revisar la iteración y consolidar los esfuerzos realizados para elaborar el gráfico de trabajo pendiente.	Facilitador
Realizar la reunión de revisión y aprobación la	Revisión de la iteración	Revisión de cumplimiento de cada tarea de la iteración para su aprobación o rechazo.	Propietario del producto y equipo Scrum

iteración.

Reunión de análisis de trabajo realizado al finalizar la iteración.	Retrospectiva general	Reunión de debate de dificultades y aciertos en la ejecución la iteración.	Facilitador y Equipo de desarrollo
---	-----------------------	--	------------------------------------

Reunión de análisis de trabajo realizado y pendiente con el objetivo de mejorar la iteración.	Refinamiento general	Reunión de profundización en el entendimiento de los requerimientos.	Propietario del producto y equipo Scrum
---	----------------------	--	---

Fuente: Ochoa (2018).

B. Herramientas para el procesamiento de datos

A continuación, se muestra una lista de las herramientas tecnológicas utilizadas para el procesamiento de datos a lo largo del ciclo de desarrollo del aplicativo:

Tabla 4

Herramientas para el procesamiento de datos

SOFTWARE	FABRICANTE	SERVICIO
JIRA	Atlassian	Herramienta de gestión de proyectos ágil que permite planificar, realizar seguimiento y gestionar el trabajo en equipos de

		desarrollo.
Confluence	Atlassian	Espacio de trabajo colaborativo donde se documentan las historias de usuario, requisitos y decisiones clave del proyecto.
Git	Software Libre	Sistema de control de versiones utilizado para gestionar el código fuente y facilitar la integración continua.
Docker	Docker Inc.	Herramienta de contenedorización que permite empaquetar el código y sus dependencias en contenedores para una implementación consistente.
Jenkins	Jenkins	Servidor de automatización que facilita la integración continua y la entrega continua.

Nota. Tecnologías usadas para procesar la información y los datos.

Fuente: Elaboración propia.

C. Técnicas de análisis de datos

Los datos obtenidos a través de la observación y las entrevistas se evaluarán mediante ciclos iterativos de desarrollo, pruebas y retroalimentación constante con los usuarios. Este enfoque facilitará la identificación y el ajuste continuo del sistema de gestión de turnos, asegurando que se adapte mejor a las necesidades del hospital, empleando métodos ágiles que enfatizan la colaboración continua con los stakeholders y la entrega progresiva de valor.

3.7.3 Diseño estadístico

Dado que la investigación es de carácter descriptivo, no se ha considerado un diseño para el contraste de hipótesis. Los resultados obtenidos a través del desarrollo del prototipo de software se presentarán utilizando tablas y figuras.

CAPÍTULO IV RESULTADOS Y DISCUSIÓN

4.1 ENTREGABLES SCRUM

4.1.1 ROLES CENTRALES

A. PROPIETARIO DEL PRODUCTO

Responsable: Bach.

La voz de los clientes y de la organización. Actúa como enlace entre las partes interesadas, asegurando que el producto cumpla con sus expectativas.

B. FACILITADOR

Responsable: Bach.

El encargado de asegurar un flujo de trabajo eficiente. Supervisa el progreso del equipo, garantiza que tengan los recursos necesarios y facilita la comunicación entre el equipo y el propietario del producto.

C. EQUIPO SCRUM

Responsable: Bach.

Este equipo se encarga de construir el producto, siguiendo rigurosamente los principios de Scrum. Desde la concepción hasta la entrega, garantizan que cada iteración sea funcional y de alta calidad.

4.1.2 HISTORIAS DE USUARIO

Trabajando en conjunto, el propietario del producto y el facilitador modelaron los requerimientos del proyecto a través de historias de usuario.

Tabla 5

Historias de usuario de la aplicación Web

HUW01	: Inicio de Sesión Básico
Como	: Usuario, Administrador, Jefatura, Aprobador Jefatura, Control Asistencia

Quiero : Iniciar sesión en el sistema utilizando mi nombre de usuario y contraseña.

Para : Acceder a mis funciones y datos.

Criterios de aceptación:

- ✓ El sistema debe validar que el nombre de usuario y la contraseña sean correctos.
- ✓ Las contraseñas deben ser almacenadas de manera segura utilizando un algoritmo de hash.
- ✓ Una vez que el usuario ha iniciado sesión correctamente, debe ser redirigido a su página de inicio personalizada.

HUW02 : Error de Credenciales

Como : Usuario, Administrador, Jefatura, Aprobador Jefatura, Control Asistencia

Quiero : Recibir un mensaje de error claro si ingreso un nombre de usuario o contraseña incorrectos.

Para : Saber que debo corregir.

Criterios de aceptación:

- ✓ Si el nombre de usuario es incorrecto, el sistema debe mostrar un mensaje de error indicando que el nombre de usuario no existe.
 - ✓ Si la contraseña es incorrecta, el sistema debe mostrar un mensaje de error indicando que la contraseña es incorrecta.
 - ✓ El mensaje de error debe ser visible y claro, ubicado cerca del campo de entrada correspondiente.
-

HUW03	: Recuperación de Contraseña
Como	: Usuario, Administrador, Jefatura, Aprobador Jefatura, Control Asistencia
Quiero	: Poder recuperar mi contraseña a través de mi correo electrónico en caso de que la olvide.
Para	: No perder acceso al sistema.

Criterios de aceptación:

- ✓ El sistema debe proporcionar un enlace “¿Olvidaste tu contraseña?” en la página de inicio de sesión.
- ✓ Al hacer clic en el enlace, el usuario debe ser redirigido a una página donde pueda ingresar su correo electrónico registrado.
- ✓ El sistema debe enviar un correo electrónico con un enlace para restablecer la contraseña.
- ✓ El enlace de restablecimiento debe ser válido por un tiempo limitado (por ejemplo, 24 horas).
- ✓ El usuario debe poder establecer una nueva contraseña a través del enlace proporcionado.

HUW04	: Cierre de Sesión
Como	: Usuario, Administrador, Jefatura, Aprobador Jefatura, Control Asistencia
Quiero	: Cerrar sesión de manera fácil y rápida desde cualquier parte de la aplicación.

Para : Asegurar que nadie más acceda a mi cuenta en el dispositivo.

Criterios de aceptación:

- ✓ Debe haber un botón o enlace visible para cerrar sesión en todas las páginas de la aplicación.
- ✓ Al hacer clic en el botón de cierre de sesión, el usuario debe ser redirigido a la página de inicio de sesión.
- ✓ La sesión del usuario debe ser terminada y no debe poder acceder a ninguna página protegida sin iniciar sesión nuevamente.

HUW05	Sesión Expirada
Como	: Usuario, Administrador, Jefatura, Aprobador Jefatura, Control Asistencia
Quiero	: Que mi sesión expire automáticamente después de un período de inactividad.
Para	: Proteger mi cuenta en caso de que deje el sistema abierto.

Criterios de aceptación:

- ✓ La sesión del usuario debe expirar automáticamente después de un período de inactividad predefinido (por ejemplo, 20 minutos).
 - ✓ El usuario debe recibir una advertencia de que su sesión está a punto de expirar, con la opción de extender la sesión.
 - ✓ Si la sesión expira, el usuario debe ser redirigido a la página de inicio de sesión y cualquier intento de acceder a páginas protegidas debe requerir un nuevo inicio de sesión.
-

HUW06	: Bloqueo de Cuenta por Intentos Fallidos
Como	: Usuario, Administrador, Jefatura, Aprobador Jefatura, Control Asistencia
Quiero	: Que mi cuenta se bloquee temporalmente después de varios intentos fallidos de inicio de sesión.
Para	: Proteger mi cuenta de accesos no autorizados.

Criterios de aceptación:

- ✓ Después de tres intentos fallidos consecutivos de inicio de sesión, la cuenta debe bloquearse automáticamente.
- ✓ Se debe mostrar un mensaje informando al usuario que su cuenta ha sido bloqueada temporalmente y proporcionándole instrucciones sobre cómo desbloquearla.
- ✓ El sistema debe permitir al usuario solicitar el desbloqueo de su cuenta mediante un proceso de verificación de identidad, como responder preguntas de seguridad o recibir un enlace de desbloqueo por correo electrónico.

HUW07	: Crear Roles de Usuario
Como	: Administrador
Quiero	: Crear nuevos roles en el sistema.
Para	: Asignar diferentes niveles de acceso y permisos.

Criterios de aceptación:

- ✓ El administrador debe poder crear un nuevo rol definiendo un nombre único y una descripción clara de sus funciones.
- ✓ El sistema debe permitir al administrador asignar permisos específicos a cada rol creado, como acceso a ciertas funciones o datos dentro de la aplicación.
- ✓ Los roles creados deben ser visibles y gestionables desde un panel de administración dedicado a la gestión de roles y permisos.

HUW08 : Asignar Roles a Usuarios

Como : Administrador

Quiero : Asignar roles específicos a los usuarios.

Para : Definir qué acciones pueden realizar en la aplicación.

Criterios de aceptación:

- ✓ El administrador debe poder asignar uno o varios roles a cada usuario durante la creación de su cuenta o posteriormente desde el panel de administración.
- ✓ Los usuarios deben heredar automáticamente los permisos y restricciones asociados con los roles asignados.
- ✓ El sistema debe notificar a los usuarios sobre cualquier cambio en sus roles que pueda afectar su acceso o funcionalidades dentro de la aplicación.

HUW09 : Modificar Roles Existentes

Como : Administrador

Quiero : Modificar los permisos asociados a un rol existente.

Para : Adaptarlos a los cambios en las políticas del hospital.

Criterios de aceptación:

- ✓ El administrador debe poder editar los permisos de un rol existente desde el panel de administración, ajustando sus capacidades según las necesidades actuales.
- ✓ Los cambios realizados en los permisos de un rol deben reflejarse inmediatamente para todos los usuarios que posean dicho rol.
- ✓ El sistema debe permitir la previsualización de cómo los cambios en los permisos afectarán las funcionalidades disponibles para los usuarios antes de aplicarlos.

HUW10	: Eliminar Roles
Como	: Administrador
Quiero	: Eliminar roles que ya no son necesarios.
Para	: Mantener la estructura de permisos limpia y actualizada.

Criterios de aceptación:

- ✓ El administrador debe poder eliminar un rol desde el panel de administración, siempre y cuando no esté asignado a ningún usuario activo.
- ✓ Si el rol está asignado a usuarios activos, el sistema debe requerir que estos usuarios sean reasignados a un nuevo rol antes de permitir la eliminación.
- ✓ Se debe mostrar una confirmación antes de proceder con la eliminación de un rol para evitar errores.
- ✓ El sistema debe actualizar automáticamente la lista de roles disponibles y sus permisos tras la eliminación de un rol.

HUW11	: Visualizar Roles de Usuario
Como	: Administrador
Quiero	: Ver una lista de todos los roles creados en el sistema.
Para	: Revisar sus permisos y usos.

Criterios de aceptación:

- ✓ El administrador debe poder visualizar los roles y funciones permitidos dentro del sistema antes de su asignación
- ✓ Cada rol en la lista debe mostrar su nombre, descripción, cantidad de usuarios asignados y permisos asociados.
- ✓ El administrador debe poder filtrar y ordenar la lista según sus necesidades.
- ✓ El sistema debe permitir al administrador seleccionar un rol para ver detalles más específicos sobre sus permisos y usuarios asignados
- ✓ La lista de roles debe actualizarse automáticamente cuando se cree, elimine o modifique un rol en el sistema

HUW12 : Visualizar Roles Asignados

Como : Administrador

Quiero : Ver los roles asignados a cada usuario

Para : Asegurarme de que tienen los permisos correctos.

Criterios de aceptación:

- ✓ El administrador debe poder acceder a una vista que muestre una lista de usuarios junto con los roles asignados a cada uno.
 - ✓ La lista debe incluir el nombre del usuario, su rol actual y cualquier rol adicional que pueda tener asignado.
 - ✓ El administrador debe poder filtrar y ordenar la lista según sus necesidades.
 - ✓ El sistema debe permitir al administrador hacer clic en un usuario para ver los detalles completos de sus roles y permisos.
-

- ✓ La lista de usuarios y sus roles asignados debe actualizarse automáticamente cuando se modifique un rol o se reasignen roles a los usuarios.

HUW13 : Definir Permisos por Rol

Como : Administrador

Quiero : Definir qué acciones o accesos están permitidos.

Para : Cada rol para controlar el uso del sistema.

Criterios de aceptación:

- ✓ El administrador debe poder acceder a una vista donde pueda seleccionar un rol específico para definir o modificar sus permisos.
- ✓ El sistema debe mostrar una lista de acciones o accesos disponibles que pueden ser asignados o restringidos para cada rol.
- ✓ El administrador debe poder seleccionar, añadir o remover permisos de forma granular, según las acciones permitidas en el sistema.
- ✓ El sistema debe mostrar un resumen claro de los permisos actuales del rol, con opciones para confirmar los cambios antes de guardarlos.
- ✓ Una vez modificados los permisos, el sistema debe actualizar automáticamente los accesos de los usuarios que tienen ese rol asignado, y reflejar los cambios de manera inmediata en su interacción con el sistema.

HUW14 : Asignación Automática de Roles

Como : Administrador

Quiero : Que los roles se asignen automáticamente a los nuevos usuarios según sus cargos.

Para : Ahorrar tiempo en la gestión manual.

Criterios de aceptación:

- ✓ El sistema debe permitir configurar reglas automáticas de asignación de roles basadas en el cargo o perfil del nuevo usuario al momento de su registro.
- ✓ El administrador debe poder definir qué roles corresponden a cada cargo, con opciones para editar o modificar estas reglas en cualquier momento.
- ✓ El administrador debe recibir una notificación o resumen confirmando la asignación automática del rol al nuevo usuario.
- ✓ El sistema debe permitir la reasignación manual de roles si es necesario, después de la asignación automática.
- ✓ Las reglas de asignación automática deben ser fácilmente accesibles y modificables desde el panel de administración.

HUW15 : Roles Basados en Grupos

Como : Administrador

Quiero : Asignar roles a grupos de usuarios.

Para : Facilitar la gestión de permisos en áreas específicas de la clínica.

Criterios de aceptación:

- ✓ El administrador debe poder crear y definir grupos de usuarios en el sistema basados en áreas.
 - ✓ El sistema debe permitir asignar uno o varios roles a todo un grupo de usuarios, aplicando los permisos correspondientes a cada miembro del grupo de manera automática.
 - ✓ Los cambios realizados en los roles asignados a un grupo deben aplicarse
-

automáticamente a todos los usuarios pertenecientes a ese grupo.

- ✓ El administrador debe poder visualizar y gestionar los roles asignados tanto a nivel de grupo como de usuario individual.
- ✓ El sistema debe permitir modificar los grupos y roles asignados en cualquier momento.
- ✓ La interfaz debe ser clara y mostrar un resumen de los grupos existentes, los roles asignados a cada grupo, y los usuarios dentro de cada grupo.

HUW16 : Asignación de Roles por Departamento

Como : Administrador

Quiero : Poder asignar roles basados en el departamento al que pertenece el usuario.

Para : Reflejar la estructura organizativa.

Criterios de aceptación:

- ✓ El sistema debe permitir al administrador seleccionar un departamento al asignar un rol a un usuario.
- ✓ Los roles disponibles deben estar filtrados según el departamento seleccionado.
- ✓ El usuario debe tener acceso solo a las funciones y datos correspondientes a su rol y departamento.

HUW17 : Roles de Auditoría

Como : Administrador

Quiero : Crear un rol específico de auditoría.

Para : Permitir a ciertos usuarios revisar actividades sin poder modificarlas.

Criterios de aceptación:

- ✓ El sistema debe permitir la creación de un rol de auditoría con permisos de solo lectura.
- ✓ Los usuarios con el rol de auditoría deben poder acceder a los registros de actividades sin opción de modificación.
- ✓ El rol de auditoría debe estar claramente identificado en el sistema.

HUW18 : Roles para Gestión de Turnos

Como : Administrador

Quiero : Crear roles específicos

Para : La gestión de turnos que permitan a ciertos usuarios programar y modificar horarios sin acceso a otras funciones.

Criterios de aceptación:

- ✓ El sistema debe permitir la creación de roles con permisos específicos para la gestión de turnos.
- ✓ Los usuarios con estos roles deben poder programar y modificar horarios.
- ✓ Los usuarios con estos roles no deben tener acceso a otras funciones fuera de la gestión de turnos.

HUW19 : Asignación Dinámica de Roles

Como : Administrador

Quiero : Que el sistema pueda ajustar los roles de los usuarios automáticamente en función de su desempeño y actividad en la

aplicación.

Para : Optimizar la asignación de roles

Criterios de aceptación:

- ✓ El sistema debe monitorear el desempeño y la actividad de los usuarios.
- ✓ Basado en reglas predefinidas, el sistema debe ajustar los roles de los usuarios automáticamente.
- ✓ El administrador debe recibir notificaciones sobre los cambios de roles automáticos.

HUW20 : Grilla de roles

Como : Administrador

Quiero : Ver el listado de roles.

Para : Una rápida administración

Criterios de aceptación:

- ✓ El sistema debe mostrar una grilla con todos los roles disponibles.
- ✓ La grilla debe incluir información relevante como nombre del rol, permisos asociados y usuarios asignados.
- ✓ El administrador debe poder filtrar y buscar roles en la grilla.

HUW21 : Agregar Empleado

Como : Administrador

Quiero : Poder agregar un nuevo empleado al sistema introduciendo su información básica (nombre, DNI, cargo, departamento).

Para : Que pueda ser gestionado dentro de la aplicación.

Criterios de aceptación:

- ✓ El sistema debe permitir al administrador ingresar la información básica del empleado.
- ✓ La información básica debe incluir nombre, DNI, cargo y departamento.
- ✓ El nuevo empleado debe ser añadido a la base de datos del sistema.

HUW22 : Asignar Roles al Empleado

Como : Administrador

Quiero : Asignar un rol específico al nuevo empleado al momento de agregarlo.

Para : Definir sus permisos dentro del sistema.

Criterios de aceptación:

- ✓ El sistema debe permitir al administrador asignar un rol al nuevo empleado durante el proceso de adición.
- ✓ El rol asignado debe determinar los permisos y accesos del empleado en el sistema.
- ✓ El empleado debe poder acceder a las funciones y datos según su rol asignado.

HUW23 : Agregar Información de Contacto del Empleado

Como : Administrador

Quiero : Registrar la información de contacto del empleado (teléfono, correo electrónico) al agregarlo.

Para : Facilitar la comunicación.

Criterios de aceptación:

- ✓ El sistema debe permitir al administrador ingresar la información de contacto del empleado.
- ✓ La información de contacto debe incluir teléfono y correo electrónico.
- ✓ La información de contacto debe ser almacenada de manera segura y accesible para la comunicación.

HUW24 : Definir Disponibilidad Inicial del Empleado

Como : Administrador

Quiero : Configurar la disponibilidad inicial del empleado (días y horas) cuando lo agrego al sistema

Para : Integrarlo rápidamente en la planificación de turnos.

Criterios de aceptación:

- ✓ El sistema debe permitir al administrador definir la disponibilidad inicial del empleado.
- ✓ La disponibilidad debe incluir días y horas específicas.
- ✓ La disponibilidad inicial debe ser utilizada para la planificación de turnos del empleado.

HUW25 : Asignar Empleado a un Departamento

Como : Administrador

Quiero : Asignar al empleado a un departamento específico al agregarlo.

Para : Para organizar mejor la gestión de turnos según las áreas de trabajo.

Criterios de aceptación:

- ✓ El sistema debe permitir al administrador asignar un departamento al nuevo empleado.
- ✓ El empleado debe ser asociado al departamento seleccionado.
- ✓ La asignación del departamento debe reflejarse en la planificación de turnos y otras funciones relacionadas.

HUW26 : Cargar foto del empleado

Como : Administrador

Quiero : Poder la foto y almacenar la foto del empleado

Para : Una mejor identificación.

Criterios de aceptación:

- ✓ El sistema debe permitir al administrador cargar una foto del empleado.
- ✓ La foto debe ser almacenada de manera segura y asociada al perfil del empleado.
- ✓ La foto del empleado debe ser visible en su perfil y en las listas de empleados.

HUW27 : Notificar al Empleado sobre su Registro

Como : Administrador

Quiero : Que el sistema envíe una notificación al empleado con sus credenciales de acceso después de ser agregado.

Para : Que pueda iniciar sesión y comenzar a usar la aplicación.

Criterios de aceptación:

- ✓ Una vez registrado un nuevo empleado en el sistema, este debe recibir automáticamente un correo electrónico con su nombre de usuario y una contraseña temporal.
- ✓ El correo de notificación debe incluir instrucciones claras sobre cómo cambiar la contraseña temporal la primera vez que inicie sesión.
- ✓ El sistema debe permitir la personalización del contenido del correo de notificación, incluyendo el logotipo del hospital y un mensaje de bienvenida.
- ✓ Se debe registrar la fecha y hora en la que se envió la notificación al empleado, junto con un estado de entrega exitoso o fallido.

HUW28 : Editar Empleados

Como : Administrador

Quiero : Editar la información de un empleado.

Para : Mantener los datos actualizados en la aplicación.

Criterios de aceptación:

- ✓ El administrador debe poder acceder a un formulario de edición de empleados desde el panel de control, donde puede actualizar datos como nombre, cargo, y departamento.
- ✓ El sistema debe validar cualquier cambio antes de permitir la actualización, asegurando que no haya campos vacíos ni datos incorrectos.
- ✓ Los cambios realizados deben ser visibles inmediatamente en todas las secciones del sistema donde se muestra la información del empleado.

HUW29 : Búsqueda de Empleados

Como	: Administrador
Quiero	: Contar con un buscador de empleados para una rápida administración.
Para	: Facilitar la gestión del personal en la aplicación.

Criterios de aceptación:

- ✓ El administrador debe poder buscar empleados por nombre, número de DNI, o departamento desde un campo de búsqueda en el panel de administración.
- ✓ El sistema debe permitir al administrador hacer clic en un empleado de los resultados para acceder a su perfil y realizar acciones adicionales como editar o eliminar.
- ✓ La funcionalidad de búsqueda debe estar optimizada para manejar grandes volúmenes de datos sin retrasos perceptibles.

HUW30 : Grilla de Empleados

Como	: Administrador
Quiero	: Ver el listado de empleados para un rápido acceso a su información.
Para	: Gestionar eficientemente al personal.

Criterios de aceptación:

- ✓ El sistema debe mostrar una tabla con todos los empleados, incluyendo columnas para nombre, cargo, departamento, y acciones.
- ✓ Cada fila en la tabla debe incluir botones de acción para editar, eliminar, o ver detalles adicionales del empleado.

HUW31 : Validación de Datos del Empleado

Como	: Administrador
Quiero	: Que el sistema valide automáticamente los datos ingresados al agregar un empleado.
Para	: Evitar duplicados o errores en el registro.

Criterios de aceptación:

- ✓ El sistema debe validar que el número de DNI ingresado no esté duplicado en la base de datos antes de permitir el registro de un nuevo empleado.
- ✓ Debe haber validaciones para asegurar que los campos obligatorios como nombre y cargo estén completos y que los formatos sean correctos (por ejemplo, el formato de correo electrónico).
- ✓ El sistema debe permitir correcciones rápidas y validar nuevamente antes de completar el registro.

HUW32 : Diseñar la Pantalla Principal

Como	: Usuario, Administrador, Jefatura, Aprobador Jefatura, Control Asistencia
Quiero	: Que la pantalla principal muestre las opciones generales de la aplicación.
Para	: Facilitar la navegación en la aplicación.

Criterios de aceptación:

- ✓ La pantalla principal debe incluir un menú de navegación lateral o superior, accesible en todo momento, con enlaces a las secciones principales como empleados, calendario, y configuración.
-

- ✓ El diseño debe ser intuitivo y adaptarse tanto a dispositivos de escritorio como móviles, asegurando que todas las opciones sean fácilmente accesibles.
- ✓ La pantalla principal debe mostrar un resumen o dashboard con información relevante como el número de empleados activos, próximos turnos, y notificaciones recientes.

HUW33	: Acceder a Funciones desde el Menú
Como	: Usuario, Administrador, Jefatura, Aprobador Jefatura, Control Asistencia
Quiero	: Acceder a las funciones generales desde el menú de la aplicación.
Para	: Navegar de manera rápida y sencilla por las diferentes áreas de la aplicación.

Criterios de aceptación:

- ✓ El menú debe ser accesible desde cualquier parte de la aplicación y debe permitir el acceso a todas las funcionalidades disponibles según el rol del usuario.
- ✓ Las opciones del menú deben estar organizadas de manera lógica y permitir la expansión o colapso de categorías para mejorar la navegación.
- ✓ Se debe resaltar visualmente la opción del menú que está activa para que el usuario sepa en qué sección se encuentra.
- ✓ El menú debe incluir accesos directos a funciones críticas como "Dashboard", "RRHH", y "Configuración".

HUW34	: Visualizar Calendario de Turnos
Como	: Administrador, Jefatura, Control Asistencia, Aprobador Jefatura
Quiero	: Que el sistema me muestre un calendario con la programación de turnos para la semana o mes.

Para : Facilitar la planificación.

Criterios de aceptación:

- ✓ El calendario debe permitir la visualización de turnos programados para una semana, mes, o un rango de fechas personalizado.
- ✓ Cada turno debe mostrarse con detalles como el nombre del empleado asignado, el horario, y el departamento.
- ✓ El calendario debe actualizarse en tiempo real cuando se realizan cambios en la programación de turnos.

HUW35 : Indicador de Sesión Activa

Como : Usuario, Administrador, Jefatura, Aprobador Jefatura, Control Asistencia

Quiero : Ver un indicador que me muestre si tengo una sesión activa y el tiempo restante de la sesión.

Para : Asegurarme de que mi sesión no expire sin que me dé cuenta.

Criterios de aceptación:

- ✓ El sistema debe mostrar un indicador visible en todo momento que informe al usuario si su sesión está activa.
- ✓ Si la sesión expira, el usuario debe ser redirigido a la pantalla de inicio de sesión con un mensaje informativo.

HUW36 : Acceso a la Configuración de Cuenta desde el Menú

Como : Usuario, Administrador, Jefatura, Aprobador Jefatura, Control Asistencia

Quiero : Acceder a la configuración de mi cuenta desde el menú de la

pantalla principal.

Para : Modificar mi perfil y cambiar mi contraseña.

Criterios de aceptación:

- ✓ El menú debe incluir un enlace claro a la "Configuración" que sea accesible desde cualquier parte de la aplicación.
- ✓ La sección de configuración debe permitir al usuario actualizar su información personal, como nombre, correo electrónico, y contraseña.
- ✓ El sistema debe notificar al usuario de manera segura si su contraseña ha sido cambiada exitosamente y sugerir iniciar sesión nuevamente por seguridad.

HUW37 : Gestión de Contratos

Como : Administrador

Quiero : Gestionar los tipos de contrato

Para : Mantener un control claro sobre la contratación del personal.

Criterios de aceptación:

- ✓ El sistema debe permitir al administrador agregar, editar, y eliminar tipos de contrato desde un módulo dedicado a la gestión de contratos.
- ✓ Cada contrato debe incluir detalles como el tipo, duración, y cualquier cláusula especial, con la opción de adjuntar documentos relevantes.
- ✓ El sistema debe garantizar que ningún empleado esté asociado a un contrato eliminado o expirado sin la reasignación a un nuevo contrato.

HUW38 : Gestionar Áreas del Negocio

Como : Administrador

Quiero : Gestionar y poder reorganizar las áreas de la empresa.

Para : Optimizar la gestión administrativa y operativa.

Criterios de aceptación:

- ✓ El administrador debe poder crear, editar, eliminar y reorganizar las áreas de la empresa desde una interfaz amigable.
- ✓ El sistema debe permitir la asignación de áreas a empleados o departamentos para una mejor organización.
- ✓ Los cambios realizados en la gestión de áreas deben actualizarse en tiempo real y reflejarse en todos los módulos relacionados.
- ✓ Se debe mostrar un resumen de las áreas creadas, con la posibilidad de visualizarlas en una estructura clara.

HUW39 : Gestionar Cargos

Como : Administrador

Quiero : Gestionar y poder reorganizar los cargos.

Para : Mantener una estructura clara de roles y responsabilidades en la empresa.

Criterios de aceptación:

- ✓ El administrador debe poder crear, editar, eliminar y asignar cargos desde el panel de administración.
 - ✓ El sistema debe permitir asociar cargos a empleados específicos para mantener una estructura organizativa clara.
 - ✓ Los cambios en los cargos deben reflejarse en los permisos y accesos asignados a los empleados según su cargo.
 - ✓ Debe mostrarse un resumen claro de los cargos disponibles y la cantidad de
-

empleados asignados a cada cargo.

HUW40	: Gestionar Tipos de Turno
Como	: Administrador
Quiero	: Gestionar los tipos de turno
Para	: Organizar los diferentes horarios y turnos de los empleados de manera eficiente.

Criterios de aceptación:

- ✓ El administrador debe poder crear, modificar y eliminar diferentes tipos de turnos en el sistema.
- ✓ Los turnos deben poder asociarse fácilmente a los empleados y áreas, y el sistema debe reflejar los turnos asignados en los reportes.
- ✓ Los cambios en los tipos de turno deben actualizarse automáticamente en los horarios y planes de turnos ya establecidos.
- ✓ Se debe mostrar un resumen de los tipos de turno disponibles, junto con su asignación actual a los empleados.

HUW41	: Validación de Maestros Duplicados
Como	: Administrador
Quiero	: Que el sistema valide automáticamente que no se dupliquen maestros de datos al momento de crear o modificar registros.
Para	: Mantener la integridad de la base de datos y evitar la duplicación de información.

Criterios de aceptación:

- ✓ El sistema debe realizar una verificación automática cuando se crea o edita un registro maestro para evitar duplicados.
- ✓ Si se detecta un duplicado, el sistema debe mostrar un mensaje de advertencia y no permitir la creación del registro hasta que se resuelva el conflicto.
- ✓ Debe ofrecer una opción para gestionar manualmente posibles conflictos en caso de errores.

HUW42 : Gestión de Profesiones

Como : Administrador

Quiero : Gestionar las profesiones de los empleados para poder asignarlas en la base de datos del sistema.

Para : Tener una mejor organización y categorización de los empleados según sus profesiones.

Criterios de aceptación:

- ✓ El administrador debe poder crear, editar y eliminar profesiones desde el panel de administración.
- ✓ El sistema debe permitir asignar una o varias profesiones a cada empleado según su perfil.
- ✓ Los cambios en la gestión de profesiones deben reflejarse automáticamente en las bases de datos de empleados y en los reportes correspondientes.
- ✓ Se debe mostrar un resumen de las profesiones creadas y la cantidad de empleados que las poseen.

HUW43 : Programar Turnos

Como : Administrador

Quiero : Tener una opción "Programar"

Para : Poder asignar horarios a los empleados de manera organizada.

Criterios de aceptación:

- ✓ El sistema debe permitir al administrador asignar y planificar turnos para los empleados según los departamentos.
- ✓ La interfaz debe ser intuitiva y permitir la visualización de los turnos programados por departamento en una vista de calendario.
- ✓ Los empleados deben ser notificados automáticamente sobre los turnos programados y cualquier cambio que se realice en su asignación.
- ✓ Se debe permitir la exportación de la programación de turnos.

HUW44 : Gestionar Turnos

Como : Administrador

Quiero : Poder gestionar los turnos

Para : Mantener una planificación adecuada del personal.

Criterios de aceptación:

- ✓ El administrador debe poder modificar, eliminar o reasignar turnos desde una vista centralizada del sistema.
 - ✓ El sistema debe reflejar automáticamente los cambios realizados en los turnos en los calendarios de los empleados afectados.
 - ✓ Los empleados deben ser notificados de cualquier cambio en sus turnos a través de alertas automáticas.
-

- ✓ Debe haber una opción para generar reportes de turnos asignados, modificados o pendientes de aprobación.

HUW45 : Solicitar Aprobación de Turnos

Como : Administrador

Quiero : Que el sistema permita que las programaciones de turnos se aprueben

Para : Garantizar la revisión y validación antes de su ejecución.

Criterios de aceptación:

- ✓ El sistema debe ofrecer una opción para que los supervisores o encargados revisen y aprueben los turnos programados.
- ✓ Debe generarse una notificación cuando se requiera la aprobación de turnos, con la opción de aprobar o rechazar desde una interfaz dedicada.
- ✓ El sistema no debe permitir la ejecución de turnos programados hasta que hayan sido aprobados.
- ✓ Debe registrarse un historial de las solicitudes de aprobación y sus resultados para auditoría.

HUW46 : Visualizar Programación de Turnos

Como : Administrador

Quiero : Ver todas las programaciones de turno

Para : Tener un acceso rápido a la información

Criterios de aceptación:

- ✓ El administrador debe poder acceder a una vista que muestre todas las programaciones de turnos en una interfaz clara y organizada.
-

- ✓ La vista debe ofrecer opciones para ver los turnos en diferentes formatos, como una lista o calendario, para facilitar su análisis.
- ✓ Los cambios realizados en la programación de turnos deben reflejarse de manera inmediata en esta vista.
- ✓ El administrador debe poder hacer clic en una programación específica para ver detalles adicionales o realizar modificaciones.

HUW47	: Exportar Programación de Turnos
Como	: Administrador
Quiero	: Exportar la programación de turnos en un documento PDF
Para	: Mantener un registro accesible y respaldado de los turnos programados.

Criterios de aceptación:

- ✓ El sistema debe permitir exportar la programación de turnos seleccionada en formato PDF.
- ✓ La exportación debe incluir todos los detalles relevantes del turno.
- ✓ El archivo generado debe estar formateado de manera clara y ser apto para impresión o archivo digital.
- ✓ El sistema debe ofrecer opciones de filtrado para que el administrador elija qué turnos desea exportar.

HUW48	: Aprobar Programación de Turnos
Como	: Administrador
Quiero	: Recibir todas las solicitudes de programación de turnos
Para	: Asegurarme de que la planificación de turnos es correcta.

Criterios de aceptación:

- ✓ El supervisor debe recibir notificaciones automáticas cuando haya solicitudes de programación de turnos pendientes.
- ✓ El sistema debe permitir que el supervisor revise las solicitudes con los detalles del turno y el empleado involucrado antes de tomar una decisión.
- ✓ Debe haber una opción clara para aprobar o rechazar las solicitudes de turnos desde una interfaz sencilla y accesible.
- ✓ Las aprobaciones o rechazos deben registrarse y ser visibles en un historial para futuras referencias y auditorías.

Fuente: Elaboración propia.

4.1.3 PILA DEL PRODUCTO

Después de identificar las historias de usuario, se organizaron según la secuencia de ejecución o su relevancia para el proyecto, y se estimaron utilizando la técnica de Planning Poker, que proporciona una valoración cualitativa basada en su importancia.

Tabla 6

Pila del producto con valoración Plannig Poker.

ITEM	ID	HISTORIA	AL
1	HUW01	Inicio de Sesión Básico	58
2	HUW02	Error de Credenciales	23
3	HUW03	Recuperación de Contraseña	38
4	HUW04	Cierre de Sesión	23
5	HUW05	Sesión Expirada	14
6	HUW06	Bloqueo de Cuenta por Intentos Fallidos	38
7	HUW07	Crear Roles de Usuario	100

8	HUW08	Asignar Roles a Usuarios	61
9	HUW09	Modificar Roles Existentes	100
10	HUW10	Eliminar Roles	38
11	HUW11	Visualizar Roles de Usuario	23
12	HUW12	Visualizar Roles Asignados	23
13	HUW13	Definir Permisos por Rol	38
14	HUW14	Asignación Automática de Roles	61
15	HUW15	Roles Basados en Grupos	61
16	HUW16	Asignación de Roles por Departamento	100
17	HUW17	Roles de Auditoría	100
18	HUW18	Roles para Gestión de Turnos	100
19	HUW19	Asignación Dinámica de Roles	61
20	HUW20	Grilla de roles	61
21	HUW21	Agregar Empleado	100
22	HUW22	Asignar Roles al Empleado	38
23	HUW23	Agregar Información de Contacto del Empleado	100
24	HUW24	Definir Disponibilidad Inicial del Empleado	61
25	HUW25	Asignar Empleado a un Departamento	38
26	HUW26	Cargar foto del empleado	14
27	HUW27	Notificar al Empleado sobre su Registro	23
28	HUW28	Editar empleados	23
29	HUW29	Búsqueda de empleados	38
30	HUW30	Grilla de empleados	61
31	HUW31	Validación de Datos del Empleado	38

32	HUW32	Diseñar la Pantalla Principal	100
33	HUW33	Acceder a Funciones desde el Menú	61
34	HUW34	Visualizar Calendario de Turnos	14
35	HUW35	Indicador de Sesión Activa	23
36	HUW36	Acceso a la Configuración de Cuenta desde el Menú	61
37	HUW37	Gestión de contratos	61
38	HUW38	Gestionar Áreas del negocio	61
39	HUW39	Gestionar cargos	61
40	HUW40	Gestión de tipos de turno	38
41	HUW41	Validación de Maestros Duplicados	100
42	HUW42	Gestión de profesiones	61
43	HUW43	Programar turnos	38
44	HUW44	Gestionar turnos	23
45	HUW45	Solicitar aprobación de turnos	61
46	HUW46	Visualizar programación de turnos	38
47	HUW47	Exportar programación de turnos	100
48	HUW48	Aprobar programación de turnos	100

Fuente: Elaboración propia.

4.1.4 ITERACIONES DEL PROYECTO

Las iteraciones (sprints) se definió acorde a la valoración obtenida del Planning Poker y la secuencia del ciclo de ejecución planteado, se agruparon las historias de usuario con la finalidad de que cada iteración de como resultado un sub producto funcional. Las iteraciones están ordenadas por su importancia.

Tabla 7*Iteraciones del proyecto.*

ITERACIÓN	ITEM	ID	HISTORIA	AL
I Autenticación, Seguridad y Pantalla Principal	1	HUW01	Inicio de Sesión Básico	8
	2	HUW02	Error de Credenciales	3
	3	HUW03	Recuperación de Contraseña	8
	4	HUW04	Cierre de Sesión	3
	5	HUW05	Sesión Expirada	4
	6	HUW06	Bloqueo de Cuenta por Intentos Fallidos	8
	7	HUW32	Diseñar la Pantalla Principal	100
	8	HUW33	Acceder a Funciones desde el Menú	61
	9	HUW34	Visualizar Calendario de Turnos	14
	10	HUW35	Indicador de Sesión Activa	23
	11	HUW36	Acceso a la Configuración de Cuenta desde el Menú	61
II Gestión de Roles y Permisos	12	HUW07	Crear Roles de Usuario	100
	13	HUW08	Asignar Roles a Usuarios	61
	14	HUW09	Modificar Roles Existentes	100
	15	HUW10	Eliminar Roles	38
	16	HUW11	Visualizar Roles de Usuario	23
	17	HUW12	Visualizar Roles Asignados	23
	18	HUW13	Definir Permisos por Rol	38
	19	HUW14	Asignación Automática de Roles	61
	20	HUW15	Roles Basados en Grupos	61

	21	HUW16	Asignación de Roles por Departamento	100
	22	HUW17	Roles de Auditoría	100
	23	HUW18	Roles para Gestión de Turnos	100
	24	HUW19	Asignación Dinámica de Roles	61
III Gestión de Empleados	25	HUW20	Grilla de roles	61
	26	HUW21	Agregar Empleado	100
	27	HUW22	Asignar Roles al Empleado	38
	28	HUW23	Agregar Información de Contacto del Empleado	100
	29	HUW24	Definir Disponibilidad Inicial del Empleado	61
	30	HUW25	Asignar Empleado a un Departamento	38
	31	HUW26	Cargar foto del empleado	14
	32	HUW27	Notificar al Empleado sobre su Registro	23
	33	HUW28	Editar empleados	23
	34	HUW29	Búsqueda de empleados	38
	35	HUW30	Grilla de empleados	61
	36	HUW31	Validación de Datos del Empleado	38
IV Gestión de la Estructura Organizativa y Turnos	37	HUW37	Gestión de contratos	61
	38	HUW38	Gestionar Áreas del negocio	61
	39	HUW39	Gestionar cargos	61
	40	HUW40	Gestión de tipos de turno	38
	41	HUW41	Validación de Maestros Duplicados	100
	42	HUW42	Gestión de profesiones	61

	43	HUW43	Programar turnos	38
	44	HUW44	Gestionar turnos	23
	45	HUW45	Solicitar aprobación de turnos	61
	46	HUW46	Visualizar programación de turnos	38
	47	HUW47	Exportar programación de turnos	100
	48	HUW48	Aprobar programación de turnos	100

Fuente: Elaboración propia.

4.1.5 TAREAS DE LAS ITERACIONES

Se asignaron tareas para cada historia de usuario en cada iteración y se estimó el tiempo de desarrollo de cada una de ellas.

Tabla 8

Tareas de la iteración I: Autenticación, Seguridad y Pantalla Principal

ID	HISTORIA	ESTIMACIÓN (HORAS)
HUW01	Inicio de Sesión Básico	
I1T01	Estructurar directorios para aplicación Web y diseñar la plantilla general	6
I1T02	Implementar el código fuente para la validación de credenciales	2
HUW02	Error de Credenciales	
I1T03	Diseñar la interfaz de error de credenciales.	2
I1T04	Implementar el código fuente para mostrar mensajes de error	3
HUW03	Recuperación de Contraseña	

I1T05	Diseñar la interfaz de recuperación de contraseña	4
I1T06	Implementar el código fuente para el envío de correos electrónicos.	6
HUW04	Cierre de Sesión	
I1T07	Implementar el código fuente para el cierre de sesión	4
HUW05	Sesión Expirada	
I1T08	Implementar el código fuente para la expiración automática de la sesión	6
HUW06	Bloqueo de Cuenta por Intentos Fallidos	
I1T09	Diseñar la lógica de bloqueo de cuenta	3
I1T10	Implementar el código fuente para el bloqueo de cuenta	4
HUW32	Diseñar la Pantalla Principal	
I1T11	Diseñar la interfaz de la pantalla principal	6
I1T12	Implementar el código fuente para la pantalla principal	6
HUW33	Acceder a Funciones desde el Menú	
I1T13	Diseñar la interfaz del menú de navegación	4
I1T14	Implementar el código fuente para la navegación del menú	4
HUW34	Visualizar Calendario de Turnos	
I1T15	Diseñar la interfaz del calendario de turnos	5
I1T16	Implementar el código fuente para la visualización del calendario de turnos	5
HUW35	Indicador de Sesión Activa	
I1T17	Implementar el código fuente para el indicador de sesión activa	5

HUW36	Acceso a la Configuración de Cuenta desde el Menú	
I1T18	Implementar el código fuente para el acceso a la configuración de cuenta	6

Fuente: Elaboración propia.

Tabla 9

Tareas de la iteración II: Gestión de Roles y Permisos

ID	HISTORIA	ESTIMACIÓN (HORAS)
HUW07	Crear Roles de Usuario	
I2T01	Diseñar la interfaz para la creación de roles	4
I2T02	Implementar el código fuente para la creación de roles	4
HUW08	Asignar Roles a Usuarios	
I2T03	Diseñar la interfaz para la asignación de roles	3
I2T04	Implementar el código fuente para la asignación de roles	3
HUW09	Modificar Roles Existentes	
I2T05	Diseñar la interfaz para la modificación de roles	3
I2T06	Implementar el código fuente para la modificación de roles	4
HUW10	Eliminar Roles	
I2T07	Diseñar la interfaz para la eliminación de roles	2
I2T08	Implementar el código fuente para la eliminación de roles	3

HUW11	Visualizar Roles de Usuario	
I2T09	Diseñar la interfaz para la visualización de roles	3
I2T10	Implementar el código fuente para la visualización de roles	3
HUW12	Visualizar Roles Asignados	
I2T11	Diseñar la interfaz para la visualización de roles asignados	3
I2T12	Implementar el código fuente para la visualización de roles asignados	3
HUW13	Definir Permisos por Rol	
I2T13	Diseñar la interfaz para la definición de permisos por rol	4
I2T14	Implementar el código fuente para la definición de permisos por rol	4
HUW14	Asignación Automática de Roles	
I2T15	Diseñar la lógica para la asignación automática de roles	5
I2T16	Implementar el código fuente para la asignación automática de roles	5
HUW15	Roles Basados en Grupos	
I2T17	Diseñar la interfaz para la creación de roles basados en grupos	4
I2T18	Implementar el código fuente para la creación de roles basados en grupos	4
HUW16	Asignación de Roles por Departamento	
I2T19	Diseñar la interfaz para la asignación de roles por	3

	departamento	
I2T20	Implementar el código fuente para la asignación de roles por departamento	4
HUW17	Roles de Auditoría	
I2T21	Diseñar la interfaz para la creación de roles de auditoría	3
I2T22	Implementar el código fuente para la creación de roles de auditoría	3
HUW18	Roles para Gestión de Turnos	
I2T23	Diseñar la interfaz para la creación de roles para gestión de turnos	4
I2T24	Implementar el código fuente para la creación de roles para gestión de turnos	4
HUW19	Asignación Dinámica de Roles	
I2T25	Diseñar la lógica para la asignación dinámica de roles	5
I2T26	Implementar el código fuente para la asignación dinámica de roles	5

Fuente: Elaboración propia.

Tabla 10

Tareas de la iteración III: Gestión de Empleados

ID	HISTORIA	ESTIMACIÓN (HORAS)
HUW20	Grilla de roles	
I3T01	Diseñar la interfaz de la grilla de roles	4
I3T02	Implementar el código fuente para la grilla de roles	4
HUW21	Agregar Empleado	

I3T03	Diseñar la interfaz para agregar empleados	3
I3T04	Implementar el código fuente para agregar empleados	3
HUW22	Asignar Roles al Empleado	
I3T05	Diseñar la interfaz para asignar roles a empleados	3
I3T06	Implementar el código fuente para asignar roles a empleados	4
HUW23	Agregar Información de Contacto del Empleado	
I3T07	Diseñar la interfaz para agregar información de contacto del empleado	2
I3T08	Implementar el código fuente para agregar información de contacto del empleado	3
HUW24	Definir Disponibilidad Inicial del Empleado	
I3T09	Diseñar la interfaz para definir disponibilidad inicial del empleado	3
I3T10	Implementar el código fuente para definir disponibilidad inicial del empleado	3
HUW25	Asignar Empleado a un Departamento	
I3T11	Diseñar la interfaz para asignar empleados a un departamento	2
I3T12	Implementar el código fuente para asignar empleados a un departamento	3
HUW26	Cargar foto del empleado	
I3T13	Diseñar la interfaz para cargar foto del empleado	2
I3T14	Implementar el código fuente para cargar foto del empleado	2

HUW27	Notificar al Empleado sobre su Registro	
I3T15	Diseñar la lógica para notificar al empleado sobre su registro	3
I3T16	Implementar el código fuente para notificar al empleado sobre su registro	3
HUW28	Editar empleados	
I3T17	Diseñar la interfaz para editar empleados	3
I3T18	Implementar el código fuente para editar empleados	4
HUW29	Búsqueda de empleados	
I3T19	Diseñar la interfaz para la búsqueda de empleados	3
I3T20	Implementar el código fuente para la búsqueda de empleados	3
HUW30	Grilla de empleados	
I3T21	Diseñar la interfaz de la grilla de empleados	4
I3T22	Implementar el código fuente para la grilla de empleados	4
HUW31	Validación de Datos del Empleado	
I3T23	Diseñar la lógica para la validación de datos del empleado	3
I3T24	Implementar el código fuente para la validación de datos del empleado	4

Fuente: Elaboración propia.

Tabla 11

Tareas de la iteración IV: Gestión de la Estructura Organizativa y Turnos

ID	HISTORIA	ESTIMACIÓN (HORAS)
HUW37	Gestión de contratos	
I4T01	Diseñar la interfaz para la gestión de contratos	5
I4T02	Implementar el código fuente para la gestión de contratos	5
HUW38	Gestionar Áreas del negocio	
I4T03	Diseñar la interfaz para gestionar áreas del negocio	4
I4T04	Implementar el código fuente para gestionar áreas del negocio	4
HUW39	Gestionar cargos	
I4T05	Diseñar la interfaz para gestionar cargos	3
I4T06	Implementar el código fuente para gestionar cargos	4
HUW40	Gestión de tipos de turno	
I4T07	Diseñar la interfaz para la gestión de tipos de turno	3
I4T08	Implementar el código fuente para la gestión de tipos de turno	3
HUW41	Validación de Maestros Duplicados	
I4T09	Diseñar la lógica para la validación de maestros duplicados	2
I4T10	Implementar el código fuente para la validación de maestros duplicados	3
HUW42	Gestión de profesiones	
I4T11	Diseñar la interfaz para la gestión de profesiones	3
I4T12	Implementar el código fuente para la gestión de	4

	profesiones	
HUW43	Programar turnos	
I4T13	Diseñar la interfaz para programar turnos	4
I4T14	Implementar el código fuente para programar turnos	4
HUW44	Gestionar turnos	
I4T15	Diseñar la interfaz para gestionar turnos	4
I4T16	Implementar el código fuente para gestionar turnos	4
HUW45	Solicitar aprobación de turnos	
I4T17	Diseñar la lógica para solicitar aprobación de turnos	3
I4T18	Implementar el código fuente para solicitar aprobación de turnos	3
HUW46	Visualizar programación de turnos	
I4T19	Diseñar la interfaz para visualizar la programación de turnos	3
I4T20	Implementar el código fuente para visualizar la programación de turnos	4
HUW47	Exportar programación de turnos	
I4T21	Diseñar la lógica para exportar la programación de turnos	3
I4T22	Implementar el código fuente para exportar la programación de turnos	3
HUW48	Aprobar programación de turnos	
I4T23	Diseñar la lógica para aprobar la programación de turnos	3
I4T24	Implementar el código fuente para aprobar la	4

	programación de turnos	
--	------------------------	--

Fuente: Elaboración propia.

4.1.6 GRÁFICO DE TRABAJO PENDIENTE

Según los procesos desarrollados en la tabla N° 7 del capítulo III, para el ciclo de vida de Scrum, la fase V de lanzamiento, según las actividades de Scrum descritos en el capítulo II, título 2.2.4, sección D. Permite el envío de entregables funcionales, determinar métodos de lanzamiento del producto y la retrospectiva del proyecto.

Realizado las tablas anteriores que calculan las horas para cada iteración, calcularemos los días totales y las horas totales para el desarrollo del proyecto.

Tabla 12

Totalidad de horas para las iteraciones

ITERACION	HORAS
I	81
II	95
III	75
IV	85
Total	336

Fuente: Elaboración propia.

El proyecto está siendo desarrollado por el Bach. X, se estimas las horas requeridas para cumplir cada iteración en función a las horas disponibles a diario, según la cantidad de días previstos para la culminación del proyecto cumpliendo los requisitos del sistema.

Se estima para nuestra investigación una totalidad de 04 semanas considerando 8 horas diarias por 5 días a la semana.

Tabla 13

Estimación de tiempo para el desarrollo del equipo de trabajo

DESARROLLADOR	HORAS
ACTIVIDADES SCRUM	12
PERMISOS	8
OTROS	10
TOTAL	30
Iteracion de 4 semanas (160- total)	160
Capacidad (90%)	144
Capacidad diaria (Capacidad/dias de interacion)	7

Fuente: Elaboración propia.

Según la tabla se calcula para el equipo de trabajo una capacidad de desarrollo de 7 horas diarias.

Tabla 14

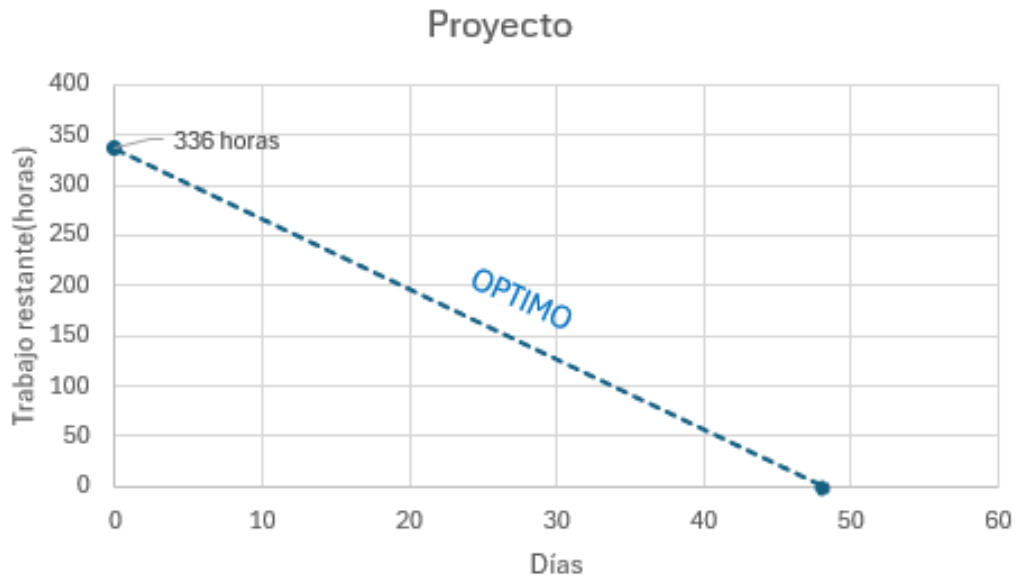
Tiempo del desarrollo del proyecto en días

TOTAL DE HORAS	336
CAPACIDAD DIARIA	7
TOTAL DE DIAS DE DESARROLLO	$336/7= 48$ Días

Fuente: Elaboración propia.

Figura 1

Gráfico de trabajo pendiente optima del proyecto.



Fuente: Elaboración propia.

Figura 2

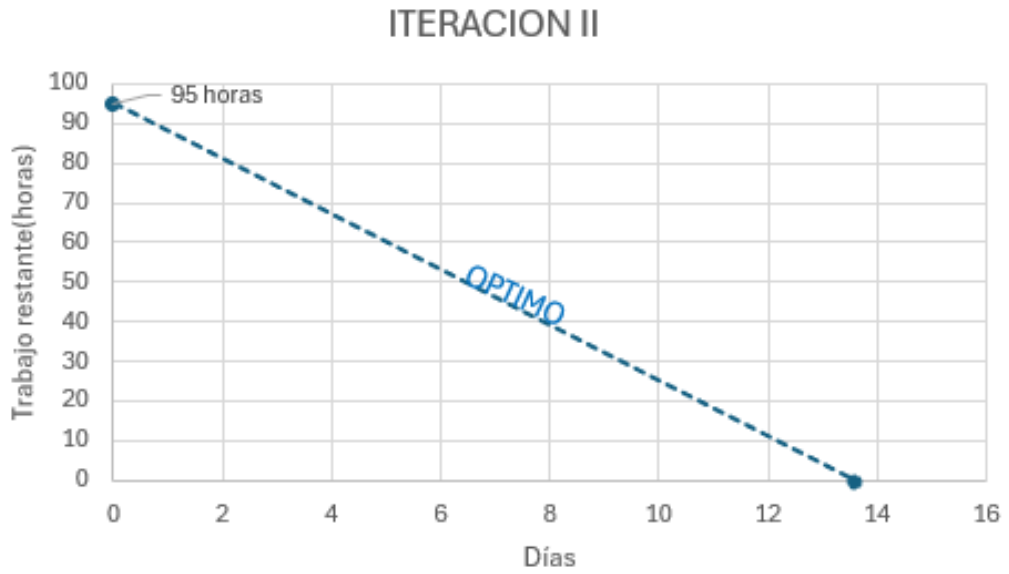
Gráfico de trabajo pendiente de Iteración I



Fuente: Elaboración propia.

Figura 3

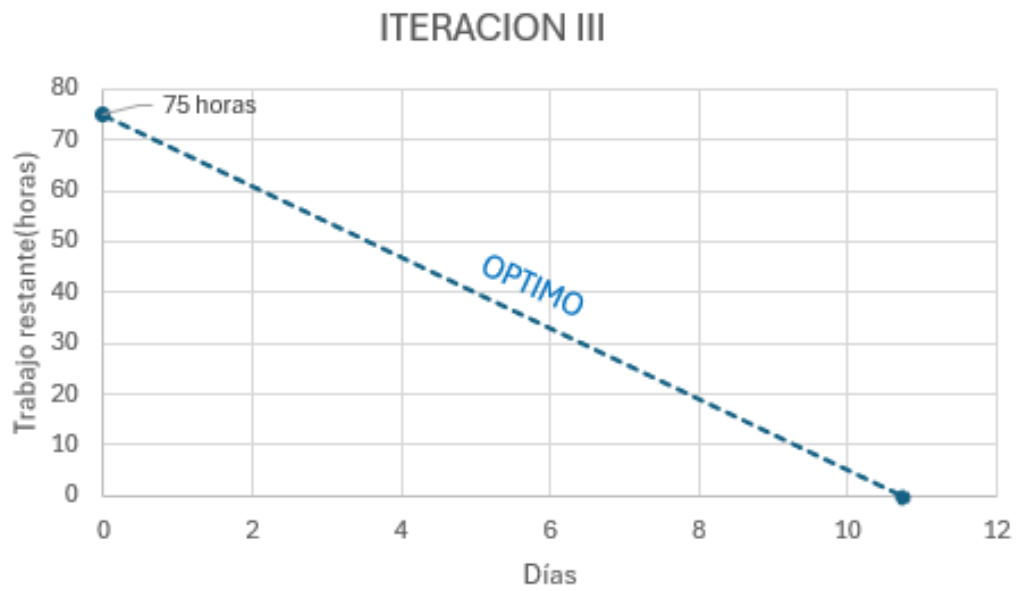
Gráfico de trabajo pendiente de Iteración II



Fuente: Elaboración propia.

Figura 4

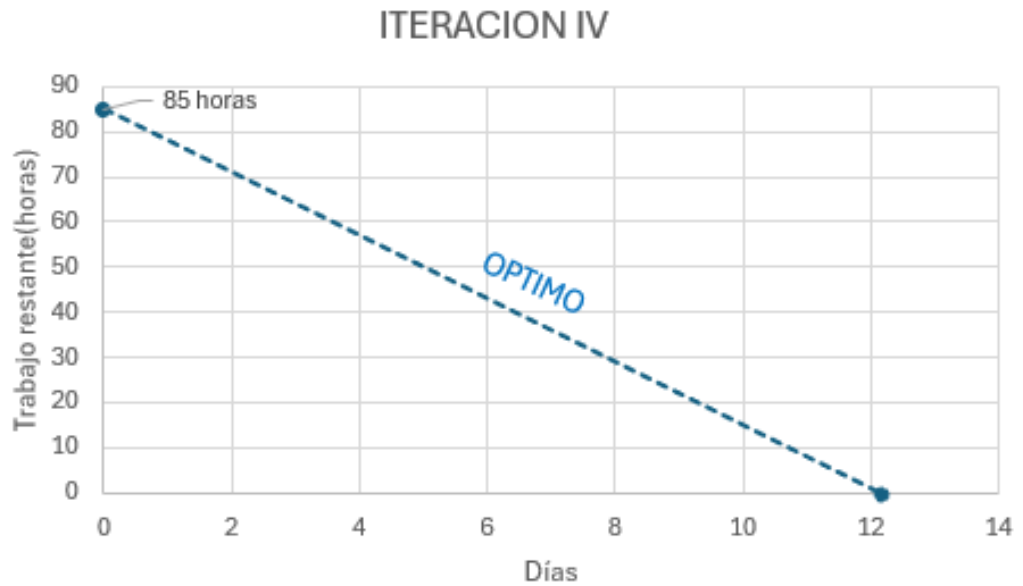
Gráfico de trabajo pendiente de Iteración III



Fuente: Elaboración propia.

Figura 5

Gráfico de trabajo pendiente de Iteración IV



Fuente: Elaboración propia.

4.2 ENTREGABLES SCRUM EN EL CICLO DE DESARROLLO

4.2.1 EJECUCIÓN DE ITERACIONES

4.2.1.1 TAREAS DE LA ITERACIÓN I: AUTENTICACIÓN, SEGURIDAD Y PANTALLA PRINCIPAL

En la primera iteración del desarrollo del sistema, se abordarán principalmente las funcionalidades de autenticación, seguridad y la pantalla principal del sistema web. Estas tareas permitirán que el usuario pueda iniciar sesión de manera segura, manejar errores de credenciales, recuperar contraseñas, cerrar sesión, gestionar la expiración automática de la sesión, así como bloquear cuentas por intentos fallidos. Además, se desarrollará la pantalla principal del sistema y se implementará un calendario de turnos. A continuación, se detallan las tareas específicas que componen esta iteración.

A. Estructurar directorios para aplicación Web y diseñar la plantilla general

Organizar los directorios de la aplicación web en secciones de front-end y back-

end. Diseñar una plantilla base que sirva de estructura común para todas las páginas de la aplicación, incluyendo cabeceras, pie de página y estilos generales.

Figura 6

Estructuración de directorios para la aplicación web y plantilla general

Id	Tipo	Periodo	Acciones
18177	Contratado	11-2021	
17193	Contratado	10-2021	
13439	Contratado	9-2021	
12478	Contratado	8-2021	
11529	Contratado	7-2021	
10577	Contratado	6-2021	

Filas por página: 8 1-6 de 6

Fuente: Elaboración propia.

B. Implementar el código fuente para la validación de credenciales

Desarrollar la lógica que permita la validación de las credenciales ingresadas por los usuarios, cotejando la información con la base de datos y asegurando que el acceso sea concedido únicamente con credenciales válidas.

Figura 7

Validación de credenciales de usuario

Logo: HOSPITAL REGIONAL DE AYACUCHO "MIGUEL A. MARISCAL LLERENA" 50 años

Usuario o Correo Este campo es requerido

Contraseña Este campo es requerido

LOGIN

EXAMENES CITAS DOCUMENTOS

Fuente: Elaboración propia.

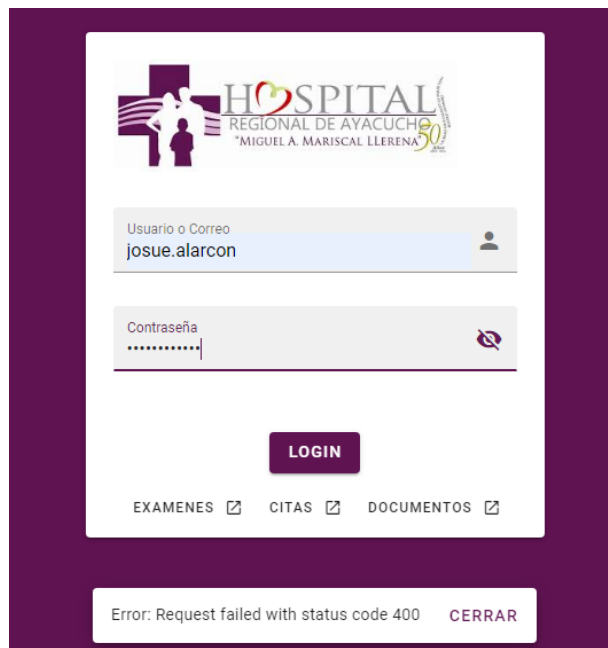
C. Diseñar la interfaz de error de credenciales

Crear una interfaz que informe al usuario cuando haya ingresado credenciales

incorrectas, mostrando un mensaje claro y específico sobre el error para facilitar la corrección.

Figura 8

Interfaz de error por credenciales incorrectas



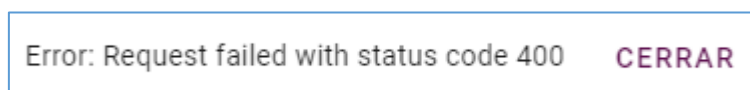
Fuente: Elaboración propia.

D. Implementar el código fuente para mostrar mensajes de error

Codificar la funcionalidad que permita desplegar los mensajes de error en la pantalla de inicio de sesión, activándose en caso de credenciales incorrectas o problemas de conexión con la base de datos.

Figura 9

Mensaje de error desplegado en la pantalla de inicio de sesión



Fuente: Elaboración propia.

E. Diseñar la interfaz de recuperación de contraseña

Crear una pantalla intuitiva donde el usuario pueda solicitar la recuperación de

su contraseña, permitiendo el ingreso de su correo electrónico registrado.

Figura 10

Interfaz de recuperación de contraseña



The screenshot shows a web interface for password recovery. At the top, there is a logo for 'HOSPITAL REGIONAL DE AYACUCHO MIGUEL A. MARISCAL LLERENA'. Below the logo, there are two input fields: 'Usuario o Correo' with the value 'josue.alarcon' and 'Contraseña' with masked characters. A link '¿Has olvidado tu contraseña?' is positioned below the password field. A purple 'LOGIN' button is centered below the link. At the bottom, there are three links: 'EXAMENES', 'CITAS', and 'DOCUMENTOS', each with an external link icon.

Fuente: Elaboración propia.

F. Implementar el código fuente para el envío de correos electrónicos

Desarrollar la lógica que permita enviar correos electrónicos automatizados para la recuperación de contraseñas, utilizando un servicio SMTP y asegurando que los correos lleguen correctamente al destinatario.

Figura 11

Envío de correo electrónico para la recuperación de contraseña



The screenshot shows a dialog box titled 'Recuperación de contraseña'. It contains a message: 'Se envió un correo con un código de confirmación para la recuperación de contraseña.' Below the message, there are two input fields: 'Tu correo electrónico' and 'Código de verificación'. A purple button labeled 'Confirmar cambio de contraseña' is located at the bottom of the dialog box.

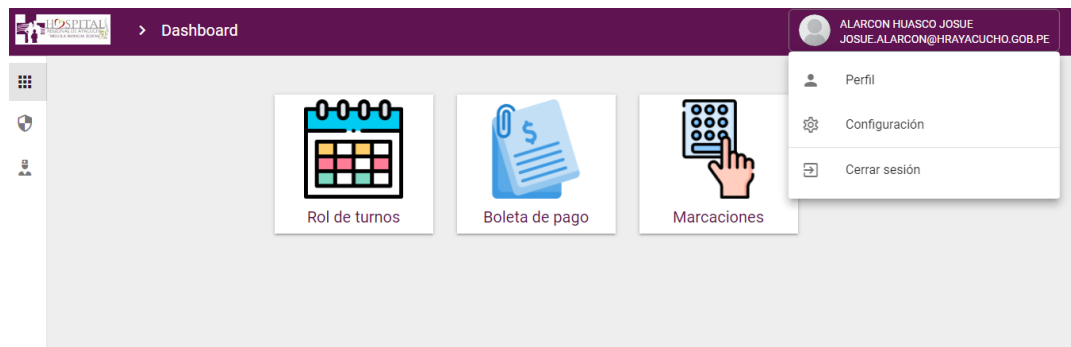
Fuente: Elaboración propia.

G. Implementar el código fuente para el cierre de sesión

Codificar la funcionalidad que permita al usuario cerrar su sesión de manera segura desde cualquier parte del sistema, eliminando los tokens de autenticación de manera efectiva.

Figura 12

Funcionalidad de cierre de sesión



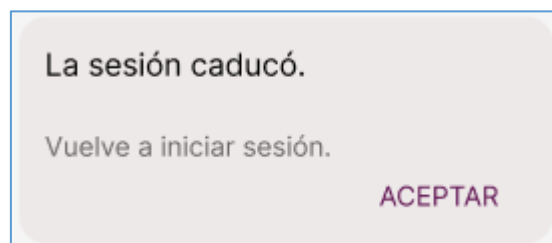
Fuente: Elaboración propia.

H. Implementar el código fuente para la expiración automática de la sesión

Desarrollar el mecanismo que cierra automáticamente la sesión del usuario después de un período de inactividad, garantizando la seguridad de la cuenta en caso de descuido o abandono de la sesión abierta.

Figura 13

Expiración automática de sesión



Fuente: Elaboración propia.

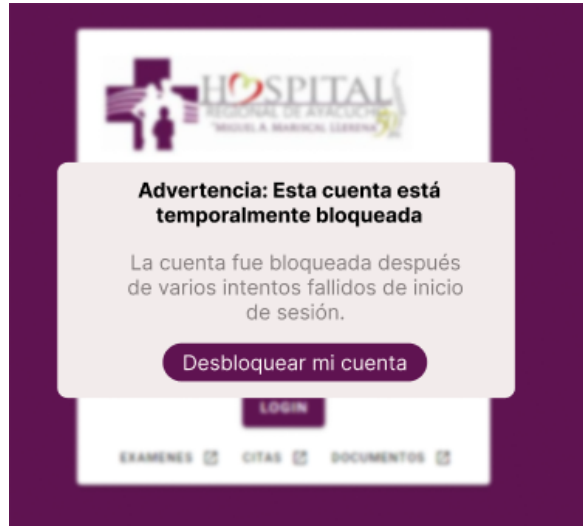
I. Diseñar la lógica de bloqueo de cuenta

Diseñar la lógica del sistema que permitirá bloquear la cuenta del usuario después

de varios intentos fallidos de inicio de sesión, evitando accesos no autorizados.

Figura 14

Lógica de bloqueo de cuenta por intentos fallidos



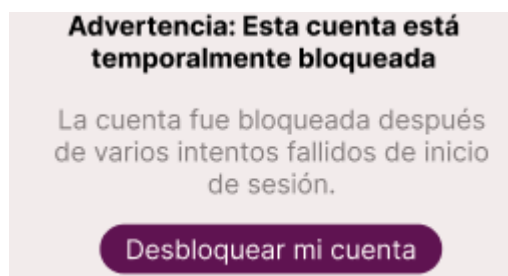
Fuente: Elaboración propia.

J. Implementar el código fuente para el bloqueo de cuenta

Codificar la funcionalidad que implementa la lógica de bloqueo de cuentas, impidiendo el acceso al sistema después de una serie de intentos fallidos y notificando al usuario sobre la situación.

Figura 15

Interfaz de bloqueo de cuenta por intentos fallidos



Fuente: Elaboración propia.

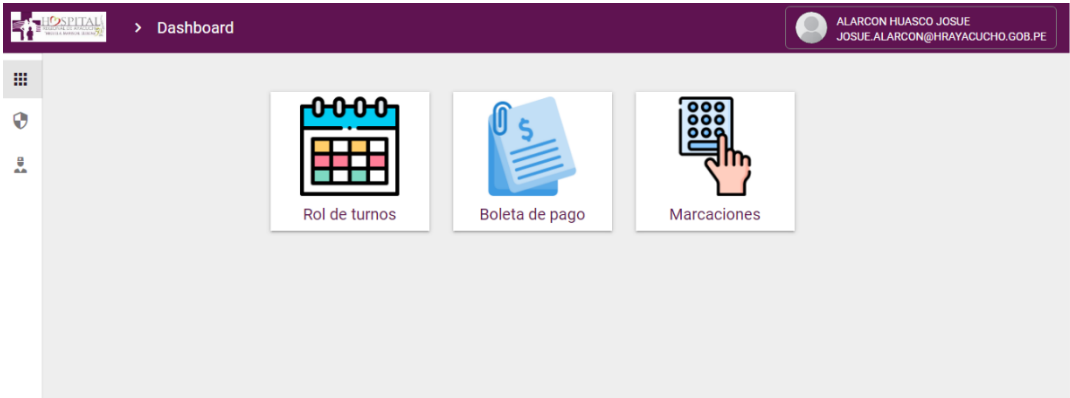
K. Diseñar la interfaz de la pantalla principal

Crear la interfaz gráfica de la pantalla principal del sistema, que será el punto de

entrada para que el usuario acceda a todas las funcionalidades, con un diseño claro y accesible.

Figura 16

Interfaz de pantalla principal del sistema



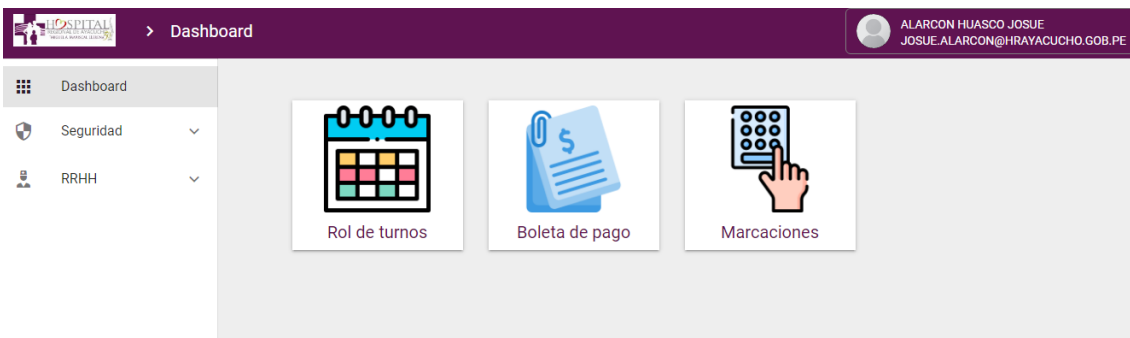
Fuente: Elaboración propia.

L. Diseñar la interfaz del menú de navegación

Crear el diseño del menú de navegación que permita al usuario moverse fácilmente entre las distintas secciones de la aplicación, asegurando una experiencia de usuario coherente.

Figura 17

Diseño de la interfaz del menú de navegación



Fuente: Elaboración propia.

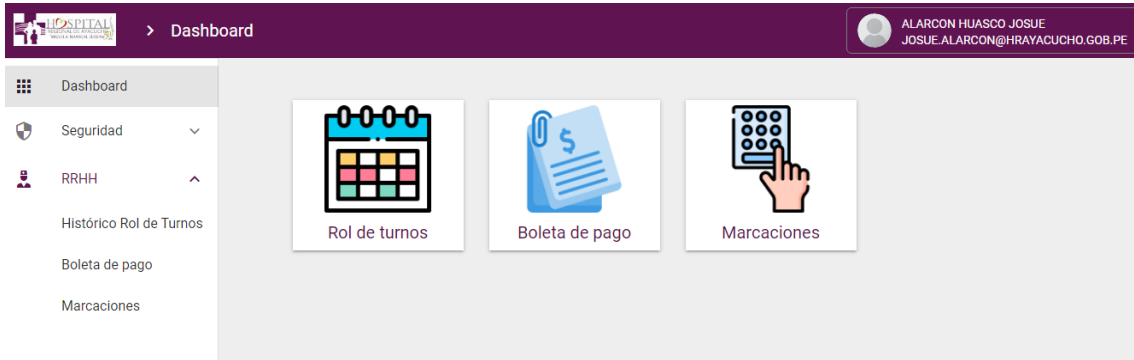
M. Implementar el código fuente para la navegación del menú

Codificar la lógica que hará funcionar el menú de navegación, asegurando que

las diferentes secciones del sistema sean accesibles según los permisos del usuario.

Figura 18

Implementación de la navegación del menú



Fuente: Elaboración propia.

N. Diseñar la interfaz del calendario de turnos

Desarrollar el diseño gráfico de un calendario interactivo donde los usuarios puedan visualizar los turnos asignados, mostrando la información de manera clara.

Figura 19

Diseño de la interfaz del calendario de turnos

LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO

Fuente: Elaboración propia.

O. Implementar el código fuente para la visualización del calendario de turnos

Codificar la funcionalidad que permita visualizar el calendario de turnos dentro

de la pantalla principal, mostrando los horarios y sus asignaciones de turnos.

Figura 20

Visualización del calendario de turnos en la pantalla principal

LUNES	MARTES	MÉRCOLES	JUEVES	VIERNES
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
TARDE	TARDE	TARDE	TARDE	TARDE

Fuente: Elaboración propia.

P. Implementar el código fuente para el indicador de sesión activa

Desarrollar una funcionalidad que muestre al usuario que su sesión está activa, indicando el tiempo de conexión o su estado de autenticación.

Figura 21

Indicador de sesión activa



Fuente: Elaboración propia.

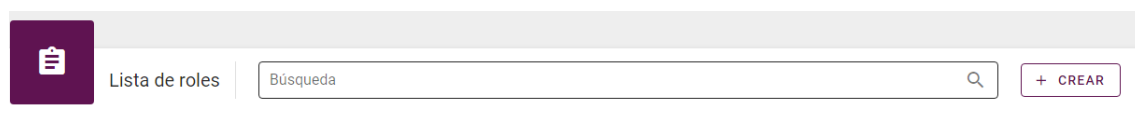
4.2.1.2 TAREAS DE LA ITERACIÓN II: GESTION DE ROLES Y TURNOS

A. Diseñar la interfaz para la creación de roles

Crear la interfaz gráfica de la ventana para la creación de roles, que será la ventana encargada para que el usuario del sistema acceda a una interfaz especializada para la creación del rol con el diseño accesible y específico.

Figura 22

Diseño de la interfaz para la creación de roles



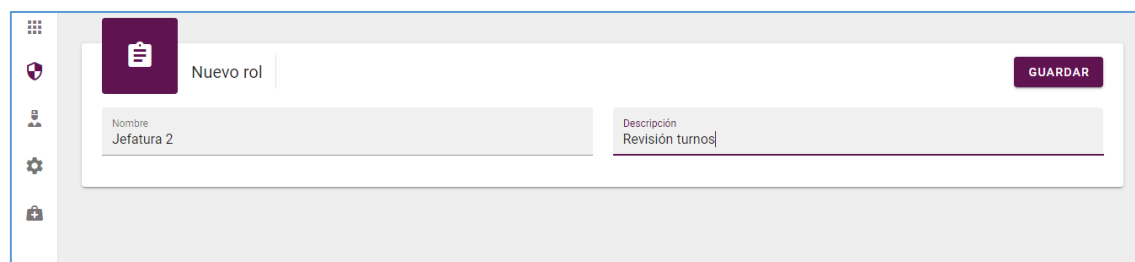
Fuente: Elaboración propia.

B. Implementar el código fuente para la creación de roles

Codificar la lógica, las inserciones y las validaciones para la creación de un rol dentro del sistema.

Figura 23

Implementación de la creación de roles



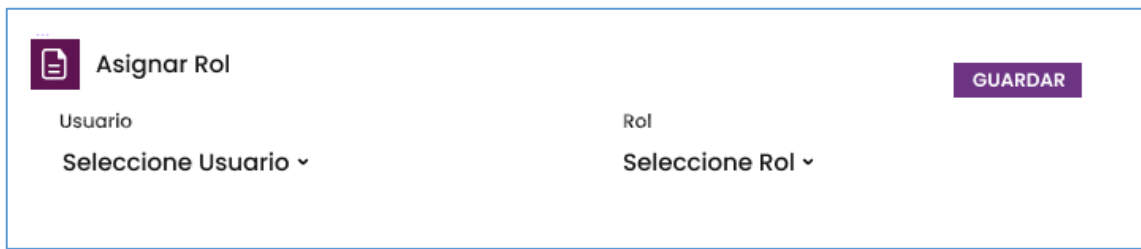
Fuente: Elaboración propia.

C. Diseñar la interfaz para la asignación de roles.

Crear la interfaz gráfica para la ventana de la asignación de roles al usuario, solamente para usuarios permitidos o asignados, con un diseño accesible, referenciado y entendible.

Figura 24

Diseño de la interfaz para la asignación de roles



Asignar Rol

GUARDAR

Usuario
Seleccione Usuario ▾

Rol
Seleccione Rol ▾

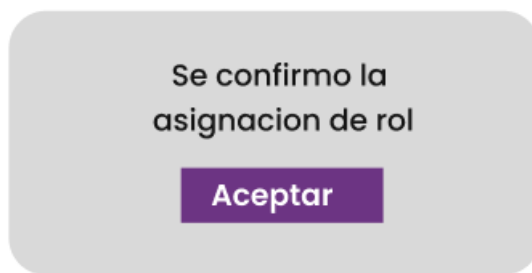
Fuente: Elaboración propia.

D. Implementar el código fuente para la asignación de roles

Codificar la lógica, la actualización y la validación para la actualización en la base de datos del sistema para la asignación de los roles solicitados.

Figura 25

Implementación de la asignación de roles



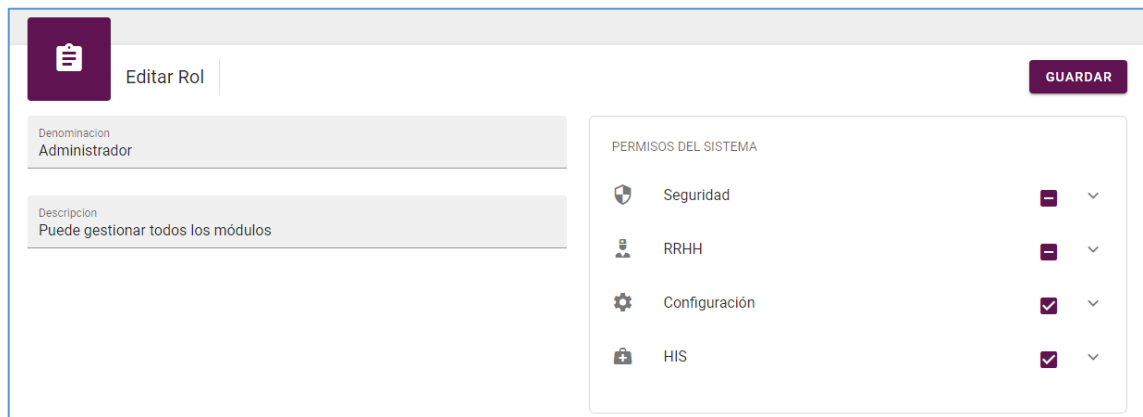
Fuente: Elaboración propia.

E. Diseñar la interfaz para la modificación de roles

Crear la interfaz gráfica e implementar componentes necesarios para la modificación de roles, ya sea con los permisos y accesos permitidos dentro del sistema.

Figura 26

Diseño de la interfaz para la modificación de roles



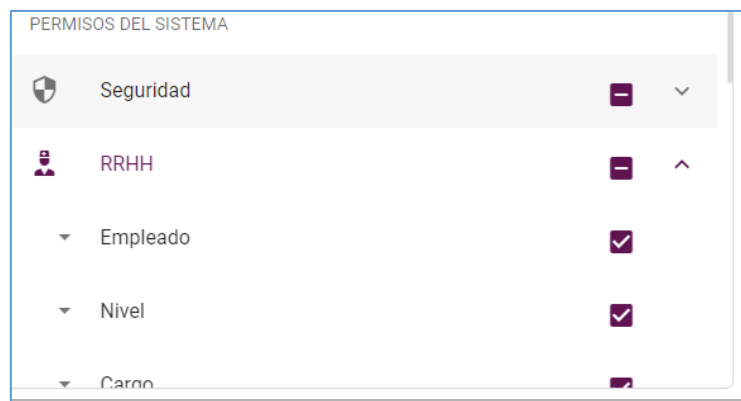
Fuente: Elaboración propia.

F. Implementar el código fuente para la modificación de roles

Codificar la lógica para permitir al usuario permitido la modificación de un rol creado modificando los accesos y características ya definidos, permitiendo las validaciones para una correcta inserción.

Figura 27

Implementación de la modificación de roles



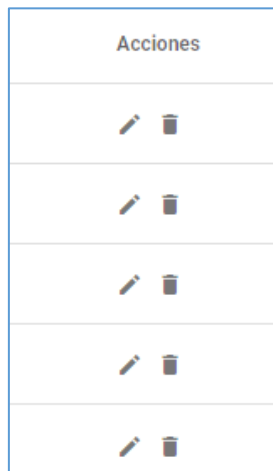
Fuente: Elaboración propia.

G. Diseñar la interfaz para la eliminación de roles

Crear la interfaz para eliminar el rol que no se desee, siempre asegurando una verificación por si sucede alguna equivocación, además de implementar una interfaz sencilla para la eliminación.

Figura 28

Diseño de la interfaz para la eliminación de roles



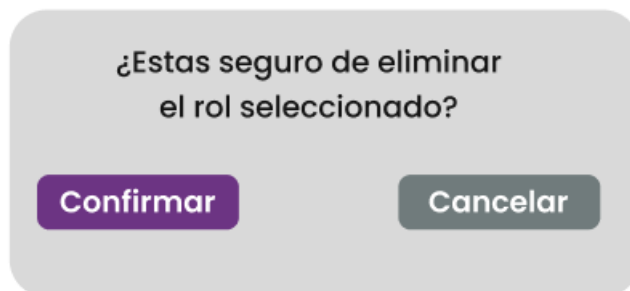
Fuente: Elaboración propia.

H. Implementar el código fuente para la eliminación de roles

Desarrollar un código que permita la eliminación correcta de un rol no requerido para la actualización en el sistema mediante una validación de seguridad.

Figura 29

Implementación de la eliminación de roles



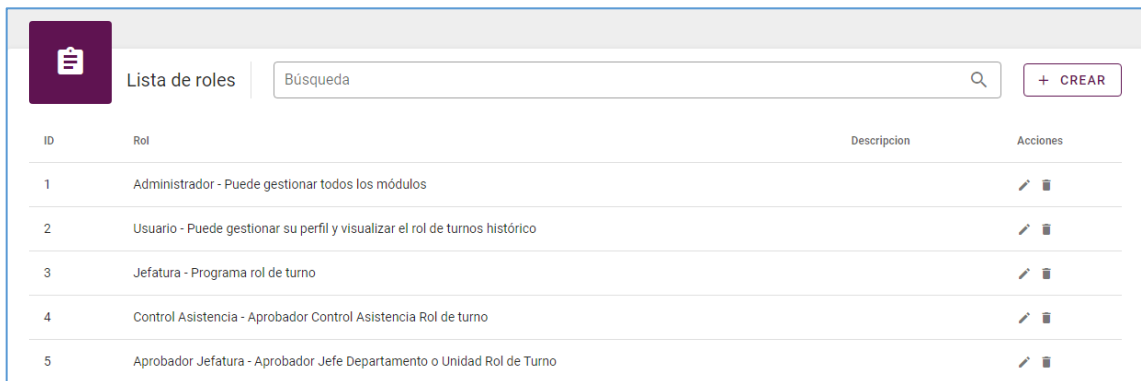
Fuente: Elaboración propia.











I. Diseñar la interfaz para la visualización de roles

Diseñar una interfaz para que el usuario pueda ver los roles creados en el sistema de manera actualizada e interactiva para poder navegar entre ellas, brindando la información necesaria para su conocimiento.

Figura 30

Diseño de la interfaz para la visualización de roles



ID	Rol	Descripción	Acciones
1	Administrador	- Puede gestionar todos los módulos	 
2	Usuario	- Puede gestionar su perfil y visualizar el rol de turnos histórico	 
3	Jefatura	- Programa rol de turno	 
4	Control Asistencia	- Aprobador Control Asistencia Rol de turno	 
5	Aprobador Jefatura	- Aprobador Jefe Departamento o Unidad Rol de Turno	 

Fuente: Elaboración propia.

J. Implementar el código fuente para la visualización de roles

Implementar un código para la visualización y la navegación de los roles creados en la base de datos para su vista y su interacción con accesos a operaciones externas.

Figura 31

Implementación de la visualización de roles

ID	Rol ↓
1	Administrador - Puede gestionar todos los módulos
2	Usuario - Puede gestionar su perfil y visualizar el rol de turnos histórico
3	Jefatura - Programa rol de turno
4	Control Asistencia - Aprobador Control Asistencia Rol de turno
5	Aprobador Jefatura - Aprobador Jefe Departamento o Unidad Rol de Turno

Fuente: Elaboración propia.

K. Diseñar la interfaz para la visualización de roles asignados

Crear la interfaz gráfica que permita a los usuarios ver los roles que han sido

asignados a los diferentes usuarios del sistema, brindando un diseño claro y comprensible que facilite la visualización de los roles asociados.

Figura 32

Diseño de la interfaz para la visualización de roles asignados



ID	Usuario	Rol
1	Tomas Morales	Administrador
2	Carlos Martinez	Usuario
3	Pedro Fernandez	Jefatura
4	Pablo Lopez	Control Asistencia

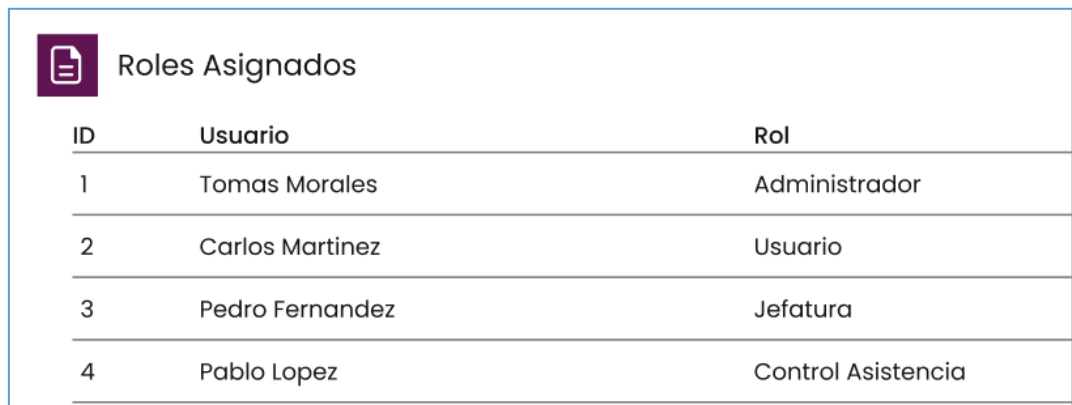
Fuente: Elaboración propia.

L. Implementar el código fuente para la visualización de roles asignados

Codificar la lógica que permita la visualización de los roles asignados a los usuarios en el sistema, incluyendo la recuperación de los datos desde la base de datos y la presentación de la información de manera organizada.

Figura 33

Implementación de la visualización de roles asignados



ID	Usuario	Rol
1	Tomas Morales	Administrador
2	Carlos Martinez	Usuario
3	Pedro Fernandez	Jefatura
4	Pablo Lopez	Control Asistencia

Fuente: Elaboración propia.

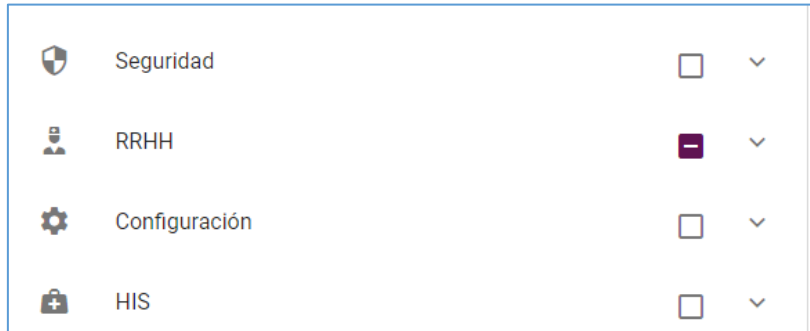
M. Diseñar la interfaz para la definición de permisos por rol

Diseñar una interfaz gráfica que permita a los usuarios definir y administrar los

permisos asociados a cada rol, asegurando que sea fácil de usar y proporcione un control adecuado de los niveles de acceso.

Figura 34

Diseño de la interfaz para la definición de permisos por rol



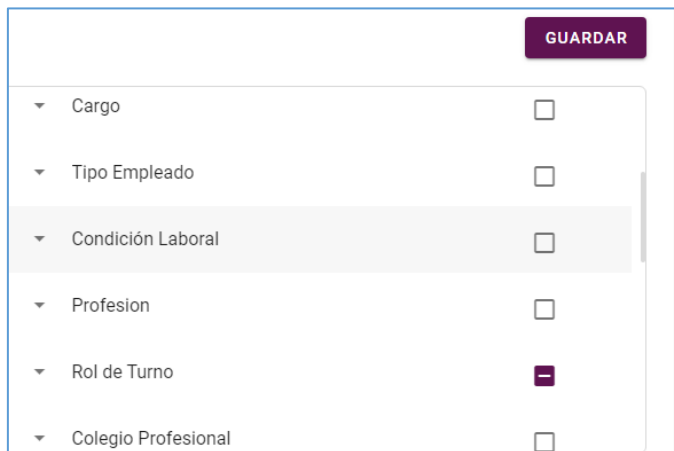
Fuente: Elaboración propia.

N. Implementar el código fuente para la definición de permisos por rol

Codificar la lógica y validación necesaria para que los permisos puedan ser asignados o modificados para cada rol, asegurando la correcta actualización de la base de datos y la validez de los cambios.

Figura 35

Implementación de la definición de permisos por rol




Fuente: Elaboración propia.

O. Diseñar la interfaz para la creación de roles basados en grupos

Crear una interfaz que permita a los usuarios crear roles que estén asociados a grupos específicos, facilitando la creación de roles colectivos.

Figura 36

Diseño de la interfaz para la creación de roles basados en grupos



The screenshot shows a web form titled "Nuevo Rol por Grupos". It features a purple document icon on the left and a purple "GUARDAR" button on the right. Below the title, there is a text input field labeled "Rol" and a dropdown menu labeled "Grupo" with the text "Seleccione Grupo" and a downward arrow.

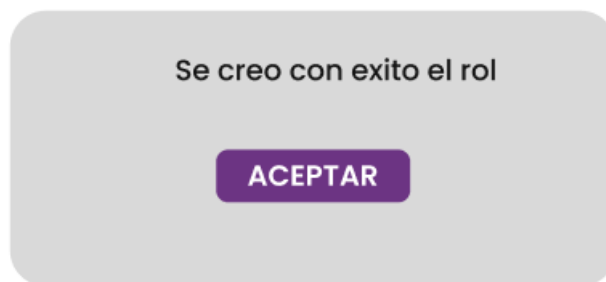
Fuente: Elaboración propia.

P. Implementar el código fuente para la creación de roles basados en grupos

Codificar la lógica que permita definir roles basados en grupos, incluyendo la creación, validación e inserción en la base de datos de dichos roles, garantizando la correcta configuración.

Figura 37

Implementación de la creación de roles basados en grupos



Fuente: Elaboración propia.

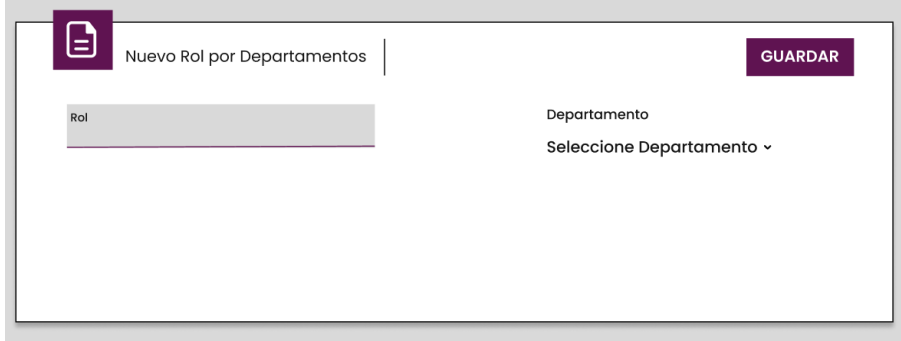
Q. Diseñar la interfaz para la asignación de roles por departamento

Crear la interfaz gráfica que permita la asignación de roles a usuarios con base

en el departamento al que pertenecen, asegurando una presentación clara y opciones de selección adecuadas.

Figura 38

Diseño de la interfaz para la asignación de roles por departamento



The screenshot shows a web form titled "Nuevo Rol por Departamentos". It features a purple document icon on the left and a purple "GUARDAR" button on the right. Below the title, there is a text input field labeled "Rol" and a dropdown menu labeled "Departamento" with the text "Seleccione Departamento" and a downward arrow.

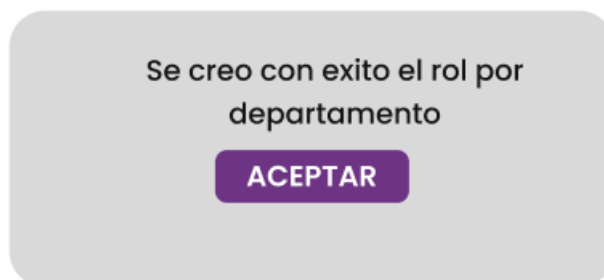
Fuente: Elaboración propia.

R. Implementar el código fuente para la asignación de roles por departamento

Codificar la lógica que permita asignar roles a usuarios en función de su departamento, actualizando la base de datos y asegurando que la asignación sea consistente y segura.

Figura 39

Implementación de la asignación de roles por departamento



Fuente: Elaboración propia.

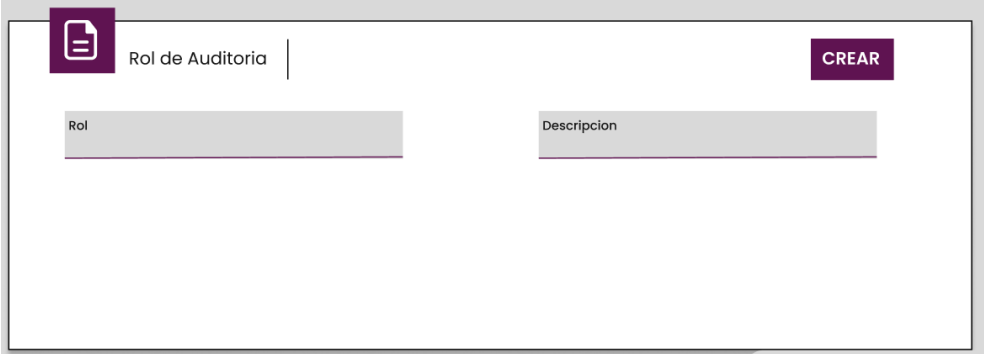
S. Diseñar la interfaz para la creación de roles de auditoría

Diseñar una interfaz que permita crear roles específicos para auditoría, con los

permisos adecuados para realizar funciones de supervisión y aseguramiento del cumplimiento dentro del sistema.

Figura 40

Diseño de la interfaz para la creación de roles de auditoría



The image shows a web form titled 'Rol de Auditoria'. At the top left is a purple icon of a document. To its right is the title 'Rol de Auditoria'. At the top right is a purple button labeled 'CREAR'. Below the title, there are two input fields: 'Rol' and 'Descripcion'. The form is enclosed in a light gray border.

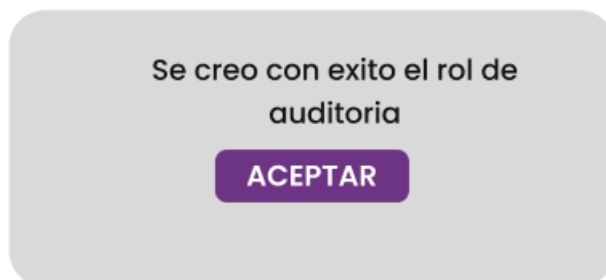
Fuente: Elaboración propia.

T. Implementar el código fuente para la creación de roles de auditoría

Codificar la lógica necesaria para crear roles de auditoría, incluyendo las validaciones que garanticen que estos roles solo tengan acceso a funciones de supervisión y no de modificación.

Figura 41

Implementación de la creación de roles de auditoría



Fuente: Elaboración propia.

U. Diseñar la interfaz para la creación de roles para gestión de turnos

Crear una interfaz que facilite la creación de roles específicos para la gestión de

turnos, proporcionando un diseño intuitivo que permita asignar roles relacionados con la programación y supervisión de turnos de trabajo.

Figura 42

Diseño de la interfaz para la creación de roles para gestión de turnos

Menu Seguridad	Seleccione submenu Rol
Acción GetRole	Ruta /role
Nombre Rol	Descripción Listar Rol

Fuente: Elaboración propia.

V. Implementar el código fuente para la creación de roles para gestión de turnos

Codificar la lógica que permita la creación de roles para la gestión de turnos, asegurando que los permisos sean configurados correctamente y que se inserten adecuadamente en la base de datos.

Figura 43

Implementación de la creación de roles para gestión de turnos

▲ Rol de Turno	▬
Información Rol de Turnos	☑
Enviar Rol Turnos y/o Horas Complementarias a revi...	☑
Histórico Rol de Turnos	☑
Visualizar Rol Turnos	☑

Fuente: Elaboración propia.

4.2.1.3 TAREAS DE LA ITERACIÓN III: GESTIÓN DE EMPLEADOS

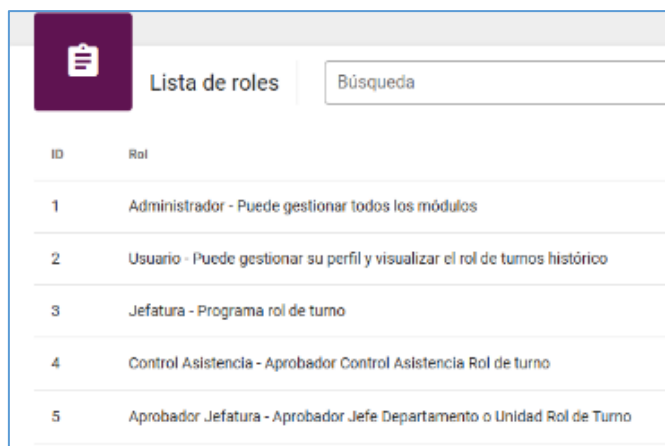
En esta iteración, el equipo se centrará en la gestión de empleados dentro del sistema. Las actividades incluirán la implementación de funcionalidades para agregar, editar, buscar y validar empleados, así como la asignación de roles y departamentos. Estas tareas son cruciales para mantener una base de datos de empleados actualizada y precisa, facilitando la gestión de recursos humanos.

A. Diseñar la interfaz de la grilla de roles

Crear una interfaz que muestre una grilla con todos los roles disponibles en el sistema, permitiendo a los administradores visualizar y gestionar los roles de manera eficiente. El diseño debe ser claro y fácil de usar, proporcionando una visión general de los roles y sus permisos asociados.

Figura 44

Diseño de la interfaz de la grilla de roles



ID	Rol
1	Administrador - Puede gestionar todos los módulos
2	Usuario - Puede gestionar su perfil y visualizar el rol de turnos histórico
3	Jefatura - Programa rol de turno
4	Control Asistencia - Aprobador Control Asistencia Rol de turno
5	Aprobador Jefatura - Aprobador Jefe Departamento o Unidad Rol de Turno

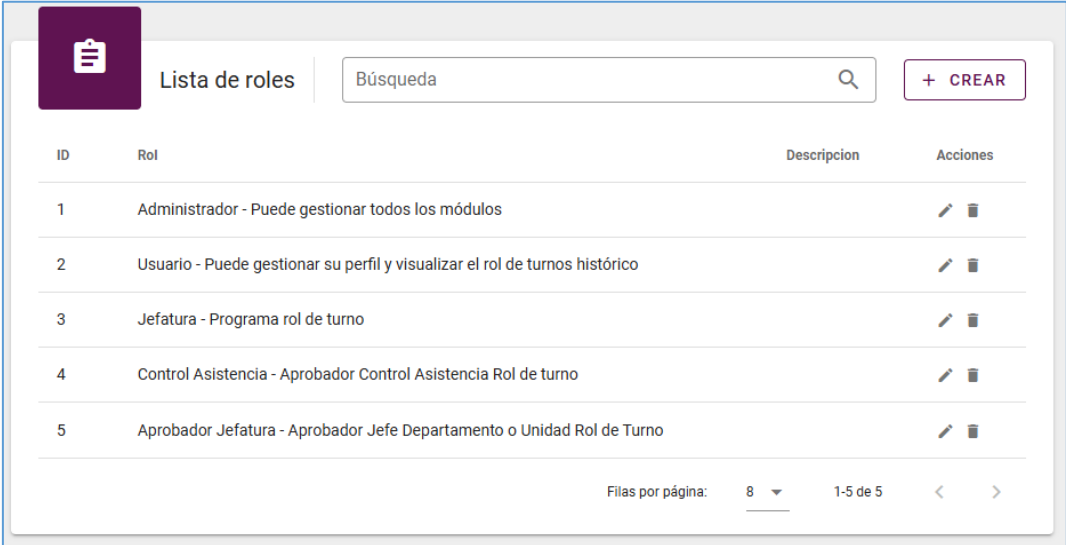
Fuente: Elaboración propia.











B. Implementar el código fuente para la grilla de roles

Desarrollar la funcionalidad que permitirá mostrar y gestionar la grilla de roles en la interfaz de usuario. Esto incluye la integración de la lógica de backend para recuperar y mostrar los datos de los roles de manera dinámica.

Figura 45

Implementación de la grilla de roles



ID	Rol	Descripción	Acciones
1	Administrador	- Puede gestionar todos los módulos	 
2	Usuario	- Puede gestionar su perfil y visualizar el rol de turnos histórico	 
3	Jefatura	- Programa rol de turno	 
4	Control Asistencia	- Aprobador Control Asistencia Rol de turno	 
5	Aprobador Jefatura	- Aprobador Jefe Departamento o Unidad Rol de Turno	 

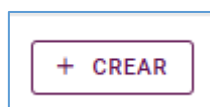
Fuente: Elaboración propia.

C. Diseñar la interfaz para agregar empleados

Crear una interfaz de usuario que permita agregar nuevos empleados al sistema. El diseño debe ser intuitivo y guiar al usuario a través del proceso de ingreso de información personal y profesional de manera clara y eficiente.

Figura 46

Diseño de la interfaz para agregar empleados



Fuente: Elaboración propia.

D. Implementar el código fuente para agregar empleados

Desarrollar la funcionalidad que permitirá agregar nuevos empleados al sistema, incluyendo la validación de datos y la integración con la base de datos para almacenar la información de los empleados.

Figura 47

Implementación para agregar empleados

Crear

GUARDAR

1 Información Pública* — 2 Información Laboral* — 3 Datos del Usuario* — 4 Información Personal* — 5 Cuentas Bancarias

Tipo de Documento* Numero de documento* Nacionalidad

Apellido Paterno* Apellido Materno* Nombres*

CONTINUAR

Fuente: Elaboración propia

E. Diseñar la interfaz para asignar roles a empleados

Crear una interfaz que permita asignar roles específicos a los empleados. El diseño debe facilitar la selección de roles y proporcionar una visión clara de los permisos asociados a cada rol.

Figura 48

Diseño de la interfaz para asignar roles a empleados

Crear

GUARDAR

1 Información Pública* — 2 Información Laboral* — 3 Datos del Usuario*

Correo de la Institucion*

Roles Sistema*

CONTINUAR

Fuente:

Elaboración propia

F. Implementar el código fuente para asignar roles a empleados

Desarrollar la funcionalidad que permitirá asignar roles a los empleados, asegurando que los permisos se actualicen correctamente en la base de datos y que los empleados tengan acceso a las funcionalidades adecuadas.

Figura 49

Implementación para asignar roles a empleados

Crear | GUARDAR

1 Información Pública* — 2 Información Laboral* — 3 Datos del Usuario* — 4 Información Personal* — 5 Cuentas Bancarias

Correo de la Institucion*

- Administrador - Puede gestionar todos los módulos
- Usuario - Puede gestionar su perfil y visualizar el rol de turnos histórico
- Jefatura - Programa rol de turno
- Control Asistencia - Aprobador Control Asistencia Rol de turno
- Aprobador Jefatura - Aprobador Jefe Departamento o Unidad Rol de Turno

CONTINUAR

Activar Windows
Vé a Configuración para activar Windows

Fuente: Elaboración Propia

G. Diseñar la interfaz para agregar información de contacto del empleado

Crear una interfaz que permita agregar y actualizar la información de contacto de los empleados, incluyendo teléfono, correo electrónico y dirección. El diseño debe ser sencillo y fácil de usar.

Figura 50

Diseño de la interfaz para agregar información de contacto del empleado

4 Información Personal*

Fuente: Elaboración propia

H. Implementar el código fuente para agregar información de contacto del empleado

Desarrollar la funcionalidad que permitirá agregar y actualizar la información de

contacto de los empleados en la base de datos, asegurando que los datos sean precisos y estén actualizados.

Figura 51

Implementación para agregar información de contacto del empleado

Crear | GUARDAR

1 Información Pública* — 2 Información Laboral* — 3 Datos del Usuario* — 4 Información Personal* — 5 Cuentas Bancarias

Fecha Nacimiento* | Sexo* | Estado Civil | Teléfonos

Correos | Dirección

CONTINUAR

Fuente: Elaboración propia

I. Diseñar la interfaz para definir disponibilidad inicial del empleado

Crear una interfaz que permita definir la disponibilidad inicial de los empleados, facilitando la programación de turnos y tareas. El diseño debe ser claro y permitir una fácil selección de horarios disponibles.

Figura 52

Diseño de la interfaz para definir disponibilidad inicial del empleado

N° documento	Apellidos y Nombres	1	2	3	4	5	6
Soporte Técnico		MT					
46154018	CARPIO QUISPE SHEYLA CYNTHIA	M	Elegir	Elegir	Elegir	Elegir	Elegir
42454724	BERROCAL LLOCCLA LEONCIO	L	Elegir	Elegir	Elegir	Elegir	Elegir
41901791	ZAPATA CASAVARDE RICHARD	V	Elegir	Elegir	Elegir	Elegir	Elegir

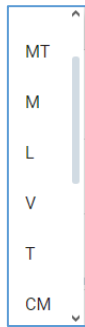
Fuente: Elaboración propia

J. Implementar el código fuente para definir disponibilidad inicial del empleado

Desarrollar la funcionalidad que permitirá definir y almacenar la disponibilidad inicial de los empleados en la base de datos, asegurando que la información sea accesible para la programación de turnos.

Figura 53

Implementación para definir disponibilidad inicial del empleado



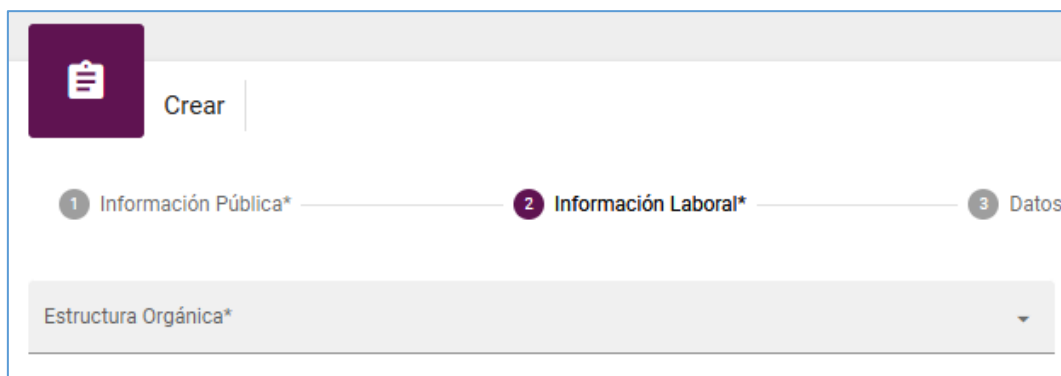
Fuente: Elaboración propia

K. Diseñar la interfaz para asignar empleados a un departamento

Crear una interfaz que permita asignar empleados a departamentos específicos, asegurando una organización clara y eficiente. El diseño debe facilitar la selección de departamentos y mostrar la estructura organizativa.

Figura 54

Diseño de la interfaz para asignar empleados a un departamento



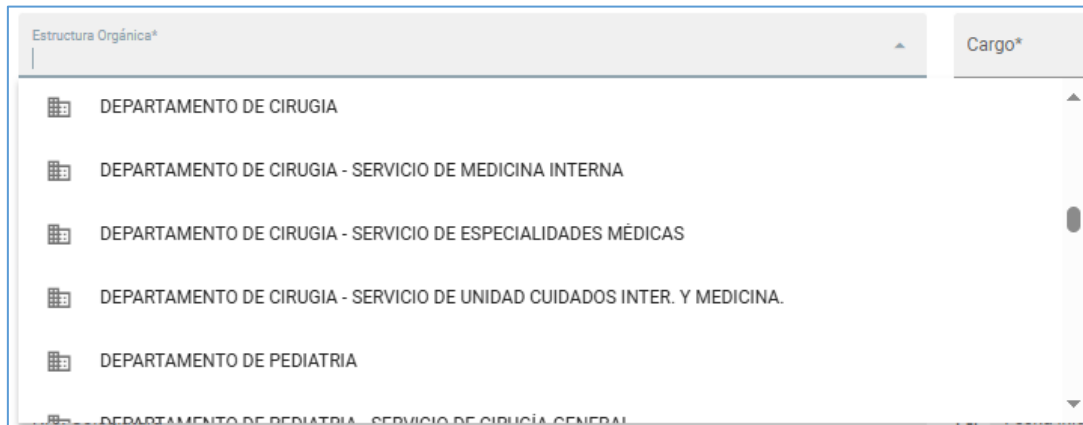
Fuente: Elaboración propia

L. Implementar el código fuente para asignar empleados a un departamento

Desarrollar la funcionalidad que permitirá asignar empleados a departamentos en la base de datos, asegurando que la información sea precisa y esté actualizada.

Figura 55

Implementación para asignar empleados a un departamento



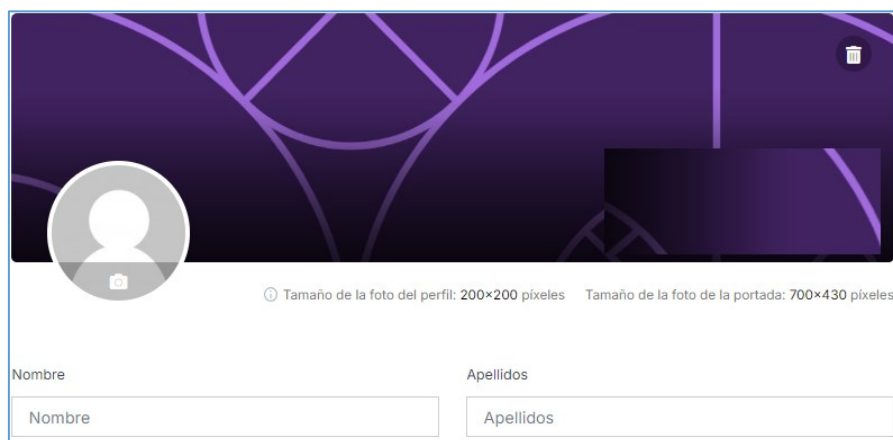
Fuente: Elaboración propia

M. Diseñar la interfaz para cargar foto del empleado

Crear una interfaz que permita cargar y actualizar la foto de perfil de los empleados, mejorando la identificación visual dentro del sistema. El diseño debe ser sencillo y permitir una fácil carga de imágenes.

Figura 56

Diseño de la interfaz para cargar foto del empleado



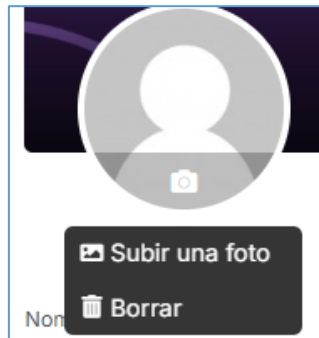
Fuente: Elaboración propia

N. Implementar el código fuente para cargar foto del empleado

Desarrollar la funcionalidad que permitirá cargar y almacenar la foto de perfil de los empleados en la base de datos, asegurando que las imágenes sean accesibles y se muestren correctamente en la interfaz de usuario.

Figura 57

Implementación para cargar foto del empleado



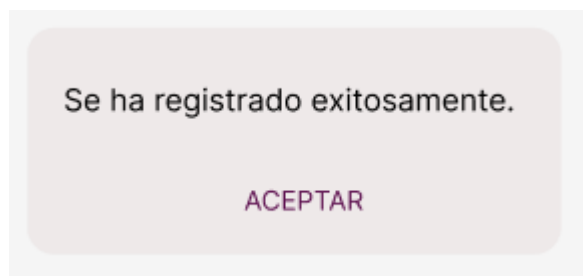
Fuente: Elaboración propia

O. Diseñar la lógica para notificar al empleado sobre su registro

Crear la lógica que enviará notificaciones automáticas a los empleados cuando sean registrados en el sistema, incluyendo detalles sobre su rol y departamento. La lógica debe ser clara y asegurar que las notificaciones se envíen correctamente.

Figura 58

Diseño de la lógica para notificar al empleado sobre su registro



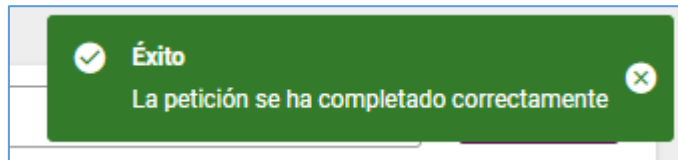
Fuente: Elaboración propia

P. Implementar el código fuente para notificar al empleado sobre su registro

Desarrollar la funcionalidad que enviará notificaciones automáticas a los empleados registrados, asegurando que reciban la información necesaria sobre su registro y roles asignados.

Figura 59

Implementación para notificar al empleado sobre su registro



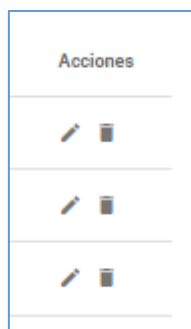
Fuente: Elaboración propia

Q. Diseñar la interfaz para editar empleados

Crear una interfaz que permita editar la información de los empleados, incluyendo datos personales, roles y departamentos. El diseño debe ser intuitivo y permitir una fácil actualización de la información.

Figura 60

Diseño de la interfaz para editar empleados



Fuente: Elaboración propia

R. Implementar el código fuente para editar empleados

Desarrollar la funcionalidad que permitirá editar la información de los empleados en la base de datos, asegurando que los cambios se guarden correctamente y se reflejen

en la interfaz de usuario.

Figura 61

Implementación para editar empleados

The screenshot shows a web interface for editing employee information. At the top left is a purple icon of a document with a pencil, followed by the text 'Editar'. On the top right is a purple button labeled 'GUARDAR'. Below this is a progress indicator with five steps: 1. Información Pública*, 2. Información Laboral*, 3. Datos del Usuario*, 4. Información Personal*, and 5. Cuentas Bancarias. The form contains several input fields: 'Tipo de Documento*' with a dropdown menu showing 'DOCUMENTO NACIONAL DE IDENTI...', 'Numero de documento*' with the value '40945036', 'Nacionalidad' with a dropdown menu, 'Apellido Paterno*' with the value 'ACUÑA', 'Apellido Materno*' with the value 'MEDRANO', and 'Nombres*' with the value 'ISABEL'. At the bottom right is a purple button labeled 'CONTINUAR'.

Fuente: Elaboración Propia

S. Diseñar la interfaz para la búsqueda de empleados

Crear una interfaz que permita buscar empleados en el sistema utilizando diferentes criterios, como nombre, rol o departamento. El diseño debe ser claro y permitir una búsqueda rápida y eficiente.

Figura 62

Diseño de la interfaz para la búsqueda de empleados

The screenshot shows a search interface. On the left is a purple icon of a document with a pencil, followed by the text 'Lista de empleados'. To the right is a search input field with the placeholder text 'Búsqueda' and a magnifying glass icon on the right side.

Fuente: Elaboración propia

T. Implementar el código fuente para la búsqueda de empleados

Desarrollar la funcionalidad que permitirá buscar empleados en la base de datos utilizando diferentes criterios, asegurando que los resultados se muestren correctamente

en la interfaz de usuario.

Figura 63

Implementación para la búsqueda de empleados



DNI	Nombre	Estructura	Cargo	Acciones
28267416	GUILLEN ALCA GLADYS	SERVICIO DE EMERGENCIA	ASISTENTE EN SERVICIO DE SALUD I (ASISTENTE/A PROFESIONAL I)	 
71402317	ALCA PAQUIYAURI DIEGO ALEXIS		ASISTENTE EN SERVICIO DE SALUD I (ASISTENTE/A PROFESIONAL I)	 
41036840	ALCA ORE JACKELINE KARINA	UNIDAD DE SEGUROS	ENFERMERA/O ESPECIALISTA	 

Filas por página: 8 1-3 de 3

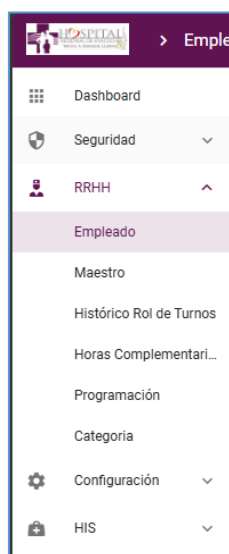
Fuente: Elaboración propia

U. Diseñar la interfaz de la grilla de empleados

Crear una interfaz que muestre una grilla con todos los empleados registrados en el sistema, permitiendo una gestión eficiente de la información. El diseño debe ser claro y fácil de usar, dando una visión general de los empleados y sus detalles.

Figura 64

Diseño de la interfaz de la grilla de empleados



Fuente: Elaboración propia

V. Implementar el código fuente para la grilla de empleados

Desarrollar la funcionalidad que permitirá mostrar y gestionar la grilla de empleados en la interfaz de usuario, incluyendo la integración de la lógica de backend para recuperar y mostrar los datos de los empleados de manera dinámica.

Figura 65

Implementación de la grilla de empleados



DNI	Nombre	Estructura	Cargo	Acciones
70447847	ALVA BENDEZU TTHAIRY MARGHIT		ASISTENTE EN SERVICIO DE SALUD I (ASISTENTE/A PROFESIONAL I)	 
44750793	AMIQUERO MERCADO XIOMARA	SERVICIO DE ENFERMERÍA EN CONSULTA EXTERNA	ASISTENTE EN SERVICIO DE SALUD I (ASISTENTE/A PROFESIONAL I)	 
28290699	AMORIN MARTINEZ ROSARIO	SERVICIO DE ENFERMERÍA EN PEDIATRÍA	ENFERMERA/O ESPECIALISTA	 
42980374	ANANCUSI ILLANES LISBETH ESTHER	UNIDAD DE SEGUROS	ASISTENTE EN SERVICIO DE SALUD I (ASISTENTE/A PROFESIONAL I)	 
70251966	ANAYA BARRON CARMEN MARILY	SERVICIO DE ENFERMERÍA EN CIRUGÍA GENERAL Y ESPECIALIDADES	ASISTENTE EN SERVICIO DE SALUD I (ASISTENTE/A PROFESIONAL I)	 
42056943	APAICO DE LA CRUZ ROCIO		ASISTENTE EN SERVICIO DE SALUD I (ASISTENTE/A PROFESIONAL I)	 
46623893	APAICO HUAMANI BANESSA	SERVICIO DE PATOLOGÍA CLÍNICA	ASISTENTE EN SERVICIO DE SALUD I (ASISTENTE/A PROFESIONAL I)	 
70050175	APAICO ROMERO EVELYN		ASISTENTE EN SERVICIO DE SALUD I (ASISTENTE/A PROFESIONAL I)	 

Filas por página: 8 9-16 de 1856

Fuente: Elaboración propia

W. Implementar el código fuente para la validación de datos del empleado

Desarrollar la funcionalidad que valide la información ingresada para los empleados en la base de datos, asegurando que los datos sean precisos y completos antes de ser guardados.

Figura 66

Implementación para la validación de datos del empleado

The image shows a registration form with a progress indicator at the top consisting of five numbered steps: 1. Información Pública*, 2. Información Laboral*, 3. Datos del Usuario* (current step), 4. Información Personal*, and 5. Cuentas Bancarias. The form contains six input fields arranged in two rows. The first row includes 'Tipo de Documento*' (dropdown), 'Numero de documento*' (text), and 'Nacionalidad' (dropdown). The second row includes 'Apellido Paterno*' (text), 'Apellido Materno*' (text), and 'Nombres*' (text). Each of the six fields has a red underline and the text 'Este campo es requerido' below it. A purple 'CONTINUAR' button is located in the bottom right corner of the form area.

Fuente: Elaboración propia

4.2.1.4 TAREAS DE LA ITERACIÓN IV

GESTIÓN DE LA ESTRUCTURA ORGANIZATIVA Y TURNOS

En la cuarta iteración, se desarrollarán las funcionalidades relacionadas con la gestión de la estructura organizativa y la administración de los turnos de los empleados. Esto incluye la gestión de contratos, áreas del negocio, cargos, y tipos de turno. A través de estas funcionalidades, el sistema ofrecerá a los usuarios una manera eficiente de organizar y gestionar la estructura interna del hospital, facilitando el control sobre el personal y sus turnos de manera clara y organizada.

A. Diseñar la interfaz para la gestión de contratos

Crear una interfaz gráfica amigable que permita a los usuarios gestionar los contratos del personal de manera clara, incluyendo opciones para crear, modificar o eliminar contratos.

Figura 67

Diseño de la interfaz para la gestión de contratos

ID	Denominación	Acciones
CL01	Nombrado	
CL02	Destacado	
CL03	Contratado 276 - Plazo fijo	
CL05	Contrato CAS	
CL06	Servicio de terceros / locación de servicios	
CL07	Residente	
CL15	Contrato Municipal CAS	
CL16	Contrato municipal por servicio de terceros / por locación de servicios	

Fuente: Elaboración propia

B. Implementar el código fuente para la gestión de contratos

Codificar la funcionalidad que permitirá a los usuarios gestionar contratos en el sistema, ofreciendo las opciones necesarias para registrar nuevos contratos, editarlos o eliminarlos según sea necesario.

Figura 68

Implementación para la gestión de contratos

Crear GUARDAR

Código: CL20

Denominación: Contrato Municipal

Fuente: Elaboración propia

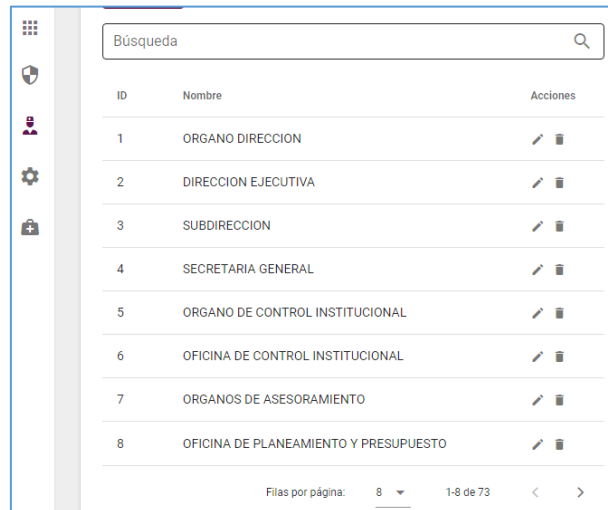
C. Diseñar la interfaz para gestionar áreas del negocio

















Diseñar una interfaz intuitiva que permita a los administradores gestionar las

distintas áreas del hospital, facilitando la creación, modificación o eliminación de áreas de trabajo dentro del sistema.

Figura 69

Diseño de la interfaz para gestionar áreas del negocio



ID	Nombre	Acciones
1	ORGANO DIRECCION	 
2	DIRECCION EJECUTIVA	 
3	SUBDIRECCION	 
4	SECRETARIA GENERAL	 
5	ORGANO DE CONTROL INSTITUCIONAL	 
6	OFICINA DE CONTROL INSTITUCIONAL	 
7	ORGANOS DE ASESORAMIENTO	 
8	OFICINA DE PLANEAMIENTO Y PRESUPUESTO	 

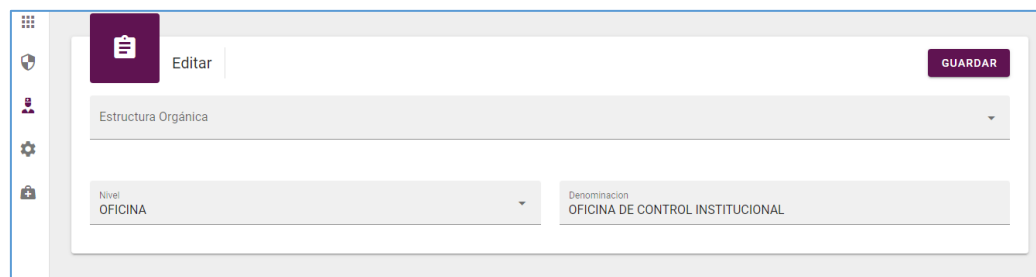
Fuente: Elaboración propia

D. Implementar el código fuente para gestionar áreas del negocio

Desarrollar la funcionalidad que permitirá gestionar las áreas del negocio, permitiendo a los administradores organizar las distintas secciones operativas del hospital dentro del sistema.

Figura 70

Implementación para gestionar áreas del negocio



Editar

GUARDAR

Estructura Orgánica

Nivel: OFICINA

Denominación: OFICINA DE CONTROL INSTITUCIONAL

Fuente: Elaboración propia

E. Diseñar la interfaz para gestionar cargos

Crear una interfaz clara y accesible para la gestión de cargos, permitiendo a los

usuarios asignar y modificar cargos específicos dentro de las áreas del hospital.

Figura 71

Diseño de la interfaz para gestionar cargos



ID	Denominación	Acciones
CA014	DIRECTOR/A EJECUTIVO	 
CA016	DIRECTOR/A ADJUNTO	 
CA018	DIRECTOR/A DE HOSPITAL II (mediana complejidad)	 
CA023	SUPERVISOR/A II	 
CA024	SUPERVISOR/A ADMINISTRATIVO/A	 
CA029	SUPERVISOR/A I	 
CA031	JEFE/A DE UNIDAD	 

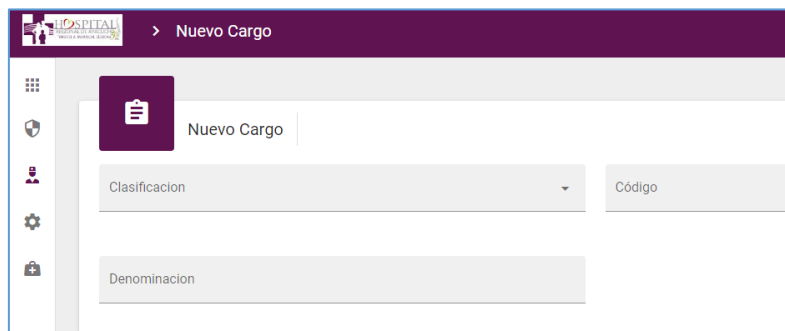
Fuente: Elaboración propia

F. Implementar el código fuente para gestionar cargos

Codificar la lógica que permitirá la gestión de cargos en el sistema, asegurando que los usuarios puedan añadir, modificar o eliminar cargos.

Figura 72

Implementación para gestionar cargos



HOSPITAL
HOSPITAL DE ESPECIALIDADES

> Nuevo Cargo

Nuevo Cargo

Clasificación Código

Denominación

Fuente: Elaboración propia

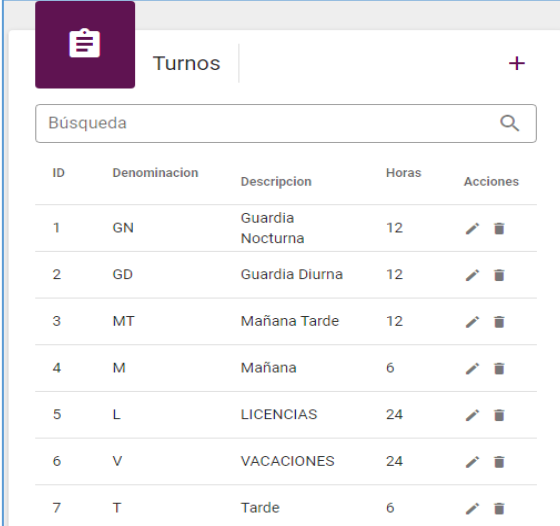
G. Diseñar la interfaz para la gestión de tipos de turno

Diseñar una interfaz que permita a los usuarios visualizar y gestionar los

diferentes tipos de turnos disponibles, con opciones para crear, modificar y eliminar turnos según los requerimientos del hospital.

Figura 73

Diseño de la interfaz para la gestión de tipos de turno



ID	Denominación	Descripción	Horas	Acciones
1	GN	Guardia Nocturna	12	
2	GD	Guardia Diurna	12	
3	MT	Mañana Tarde	12	
4	M	Mañana	6	
5	L	LICENCIAS	24	
6	V	VACACIONES	24	
7	T	Tarde	6	

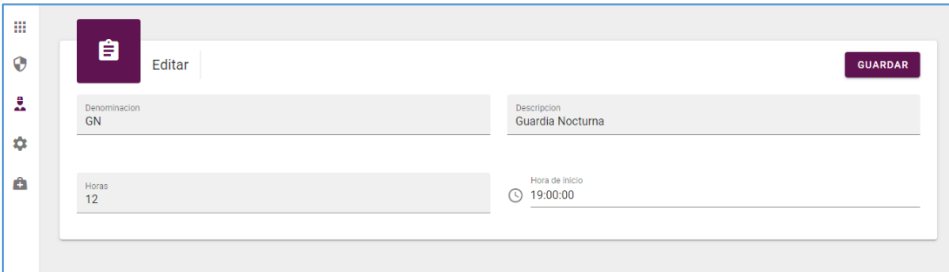
Fuente: Elaboración propia

H. Implementar el código fuente para la gestión de tipos de turno

Desarrollar el código que permitirá la gestión de los tipos de turno en el sistema, asegurando que el personal pueda ser asignado a los turnos correctos.

Figura 74

Implementación para la gestión de tipos de turno



Denominación GN	Descripción Guardia Nocturna
Horas 12	Hora de Inicio 19:00:00

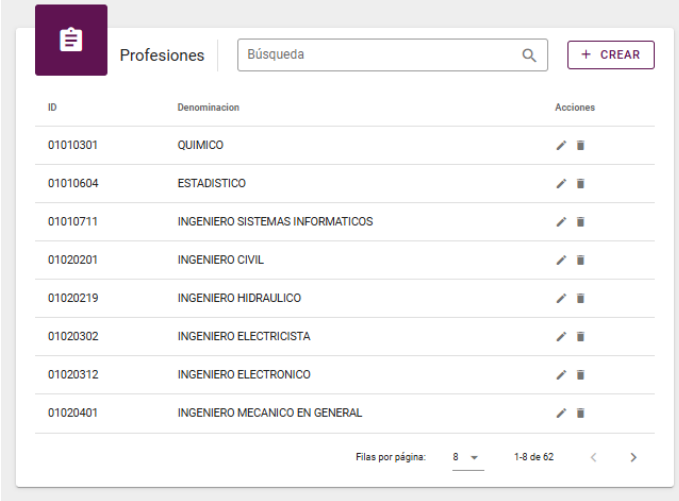
Fuente: Elaboración propia

















I. Diseñar la interfaz para la gestión de profesiones

Crear una interfaz que permita gestionar las diferentes profesiones dentro del sistema, incluyendo la creación y modificación de profesiones. El diseño debe ser claro y fácil de usar, proporcionando una visión general de las profesiones y sus detalles.

Figura 75

Diseño de la interfaz para la gestión de profesiones



ID	Denominación	Acciones
01010301	QUIMICO	 
01010604	ESTADISTICO	 
01010711	INGENIERO SISTEMAS INFORMATICOS	 
01020201	INGENIERO CIVIL	 
01020219	INGENIERO HIDRAULICO	 
01020302	INGENIERO ELECTRICISTA	 
01020312	INGENIERO ELECTRONICO	 
01020401	INGENIERO MECANICO EN GENERAL	 

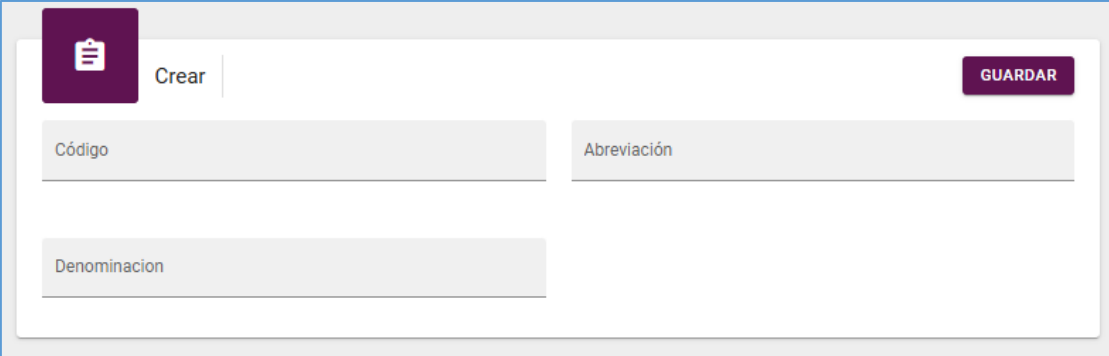
Fuente: Elaboración propia

J. Implementar el código fuente para la gestión de profesiones

Desarrollar la funcionalidad que permitirá gestionar las profesiones en la base de datos, asegurando que la información sea precisa y esté actualizada.

Figura 76

Implementación para la gestión de profesiones



Crear GUARDAR

Código

Abreviación

Denominación

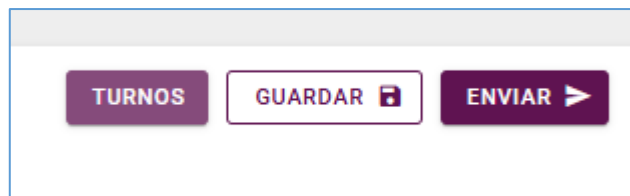
Fuente: Elaboración propia

K. Diseñar la interfaz para programar turnos

Crear una interfaz que permita programar turnos para los empleados, asegurando una planificación eficiente y equitativa. El diseño debe ser claro y fácil de usar, proporcionando una visión general de los turnos programados.

Figura 77

Diseño de la interfaz para programar turnos



Fuente: Elaboración propia

L. Implementar el código fuente para programar turnos

Desarrollar la funcionalidad que permitirá programar turnos en la base de datos, asegurando que la información sea precisa y esté actualizada.

Figura 78

Implementación para programar turnos

Denominacion	Descripción	Hora Inicio	Hora Fin	Total Horas
GN	Guardia Nocturna	7:00:00 p. m.	7:00:00	12
GD	Guardia Diurna	7:00:00 a. m.	19:00:00	12
MT	Mañana Tarde	7:00:00 a. m.	19:00:00	12
M	Mañana	7:00:00 a. m.	13:00:00	6
L	LICENCIAS			24
V	VACACIONES			24
T	Tarde	1:00:00 p. m.	19:00:00	6
CM	Consultorio Mañana	7:00:00 a. m.	13:00:00	6
CT	Consultorio Tarde	1:00:00 p. m.	19:00:00	6

Fuente: Elaboración propia

M. Diseñar la interfaz para gestionar turnos

Crear una interfaz que permita gestionar los turnos programados, incluyendo la modificación y cancelación de turnos. El diseño debe ser claro y fácil de usar, proporcionando una visión general de los turnos y sus detalles.

Figura 79

Diseño de la interfaz para gestionar turnos

ID	Denominacion	Descripcion	Horas	Acciones
1	GN	Guardia Nocturna	12	
2	GD	Guardia Diurna	12	
3	MT	Mañana Tarde	12	
4	M	Mañana	6	
5	L	LICENCIAS	24	
6	V	VACACIONES	24	
7	T	Tarde	6	
8	CM	Consultorio Mañana	6	

Filas por página: 8 1-8 de 14

Fuente: Elaboración propia

N. Implementar el código fuente para gestionar turnos

Desarrollar la funcionalidad que permitirá gestionar los turnos en la base de datos, asegurando que la información sea precisa y esté actualizada.

Figura 80:

Implementación para gestionar turnos

Crear GUARDAR

Denominacion

Descripcion

Horas

Hora de inicio

Fuente: Elaboración propia

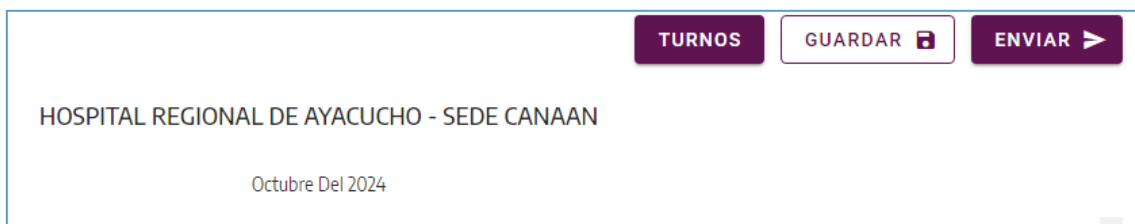
O. Diseñar la lógica para solicitar aprobación de turnos

Crear una lógica que permita a los empleados solicitar la aprobación de turnos,

donde los usuarios puedan gestionar dichas solicitudes. El diseño debe ser claro y eficiente, permitiendo a los empleados enviar solicitudes de turnos. Los usuarios deberán tener la capacidad de aprobar, rechazar o solicitar más información sobre los turnos, y deben recibir notificaciones sobre el estado de su solicitud.

Figura 81

Diseño de la lógica para solicitar aprobación de turnos



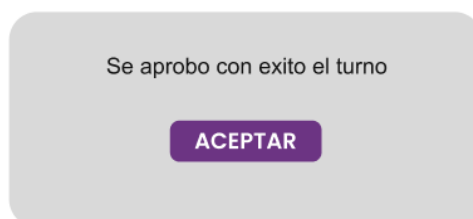
Fuente: Elaboración propia

P. Implementar el código fuente para solicitar aprobación de turnos

Desarrollar el código fuente que permita la implementación del flujo de solicitud de turnos y su aprobación. El sistema deberá gestionar estas solicitudes, enviándolas al administrador correspondiente, quien tendrá opciones para gestionar esta. El código debe trabajar con la base de datos para gestionar de manera correcta y actualizada la información del sistema.

Figura 82

Implementación para solicitar aprobación de turnos



Fuente: Elaboración propia

Q. Diseñar la interfaz para visualizar la programación de turnos

Crear una interfaz que permita a los usuarios visualizar de manera clara y

organizada la programación de turnos. El diseño debe mostrar los turnos asignados a cada empleado, incluyendo detalles como la fecha, duración y estado del turno. La interfaz debe ser intuitiva, permitiendo una navegación fácil y rápida para usuarios.

Figura 83

Diseño de la interfaz para visualizar la programación de turnos

ID	Estructura Orgánica	Mes	Año	Fecha	Estado	Acciones
1002	AREA DE INFORMATICA	Octubre		29/9/2024, 11:59:13	Enviado	

Fuente: Elaboración propia

R. Implementar el código fuente para visualizar la programación de turnos

Desarrollar el código fuente necesario para implementar una interfaz que permita visualizar la programación de turnos de manera clara y eficiente. Además, la interfaz debe actualizarse en tiempo real cuando se modifique la programación, garantizando que los usuarios tengan acceso a la información más actualizada.

Figura 84

Implementación para visualizar la programación de turnos

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
N° documento	Apellidos y Nombres																	
Soporte Tecnico																		
46154018	CARPIO QUISPE SHEYLA CYNTHIA	GD	M	L	MT	MT	M	L	V	M	MT	MT	GD	MT	M	GD	GD	GE

Fuente: Elaboración propia

S. Diseñar la lógica para exportar la programación de turnos

U. Diseñar la lógica para aprobar la programación de turnos

Desarrollar la lógica que permita a los administradores aprobar la programación de turnos de manera eficiente. Además, debe estar conectado con la base de datos para manejar una gestión de información eficiente y actualizado para su visualización.

Figura 87

Diseño de la lógica para aprobar la programación de turnos.

N° documento	Apellidos y Nombres	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
46154018	CARPIO QUISPE SHEYLA CYNTHIA	GD	M	L	MT	MT	M	L	V	M	MT	MT	GD	MT	M	GD	GD	GD

Fuente: Elaboración propia

V. Implementar el código fuente para aprobar la programación de turnos

Desarrollar el código fuente que permita a los supervisores aprobar, rechazar o solicitar modificaciones en la programación de turnos. El sistema debe proporcionar a los supervisores una vista clara de los turnos solicitados. El código debe permitir a los supervisores tomar decisiones sobre los turnos y registrar automáticamente dichas acciones en el sistema.

Figura 88

Implementación para aprobar la programación de turnos

¿Está seguro de aprobar el rol de turnos del/de la **AREA DE INFORMATICA** para el periodo **octubre del 2024**?

CANCELAR **GUARDAR**

Fuente: Elaboración propia

4.2.2 SCRUM DIARIO

4.2.2.1 ITERACIÓN I: Autenticación, Seguridad y Pantalla Principal

A. ¿Qué hice ayer?

- Se estructura y gestiona la infraestructura del proyecto, archivos y directorios, para la implementación del código del proyecto.
- Se diseñó las interfaces generales del proyecto.
- Se implementó la lógica de la validación de credenciales

B. ¿Qué voy a hacer hoy?

- Diseño de la interfaz siguiendo la lógica para la validación de los errores de las credenciales.
- Implementación de los mensajes de error según la lógica diseñada.

C. ¿Tengo algún impedimento?

- Dificultad en la modularización de las interfaces.
- Problemas de sincronización entre el manejo de sesiones y el bloqueo de cuentas

4.2.2.2 ITERACIÓN II: Gestión de Roles y Permisos

A. ¿Qué hice ayer?

- Se diseñó las interfaces y se implementó la lógica para crear, modificar, asignar y eliminar roles
- Se implementó los roles basados en grupos y departamentos.

B. ¿Qué voy a hacer hoy?

- Completar la asignación dinámica de roles y la validación de esta.
- Implementar las interfaces y lógica para la visualización de los roles de usuario y asignados.

C. ¿Tengo algún impedimento?

- Dificultad para definir permisos específicos para cada rol en la interfaz debido a requerimientos complejos.
- Integración de roles de auditoria con la gestión de turnos.

4.2.2.3 ITERACIÓN III: Gestión de Empleados

A. ¿Qué hice ayer?

- Diseñé la interfaz para agregar empleados y asignar roles.
- Implementé el código fuente para agregar empleados y la validación de datos.

B. ¿Qué voy a hacer hoy?

- Concluir la implementación de las funcionalidades relacionados con los empleados.
- Diseñar la interfaz para asignar roles a empleados.
- Implementar el código fuente para asignar roles a empleados.
- Implementar las funcionalidades de búsqueda y edición de empleados.

C. ¿Tengo algún impedimento?

- El equipo informático presenta problemas de rendimiento, lo que retrasa el desarrollo del código.

4.2.2.4 ITERACIÓN IV: Gestión de la Estructura Organizativa y Turnos

A. ¿Qué hice ayer?

- Diseñé la interfaz para la gestión de contratos y areas del negocio.
- Implementé el código fuente para la gestión de contratos.

B. ¿Qué voy a hacer hoy?

- Diseñar la interfaz para gestionar áreas del negocio.
- Implementar el código fuente para gestionar áreas del negocio.

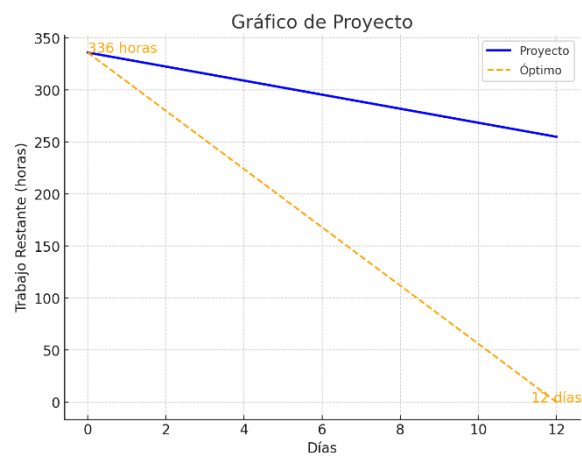
C. ¿Tengo algún impedimento?

- Problemas de integración con las bases de datos para la gestión de contratos y áreas del negocio.

4.2.3 GRÁFICAS DE TRABAJO PENDIENTE

Figura 89

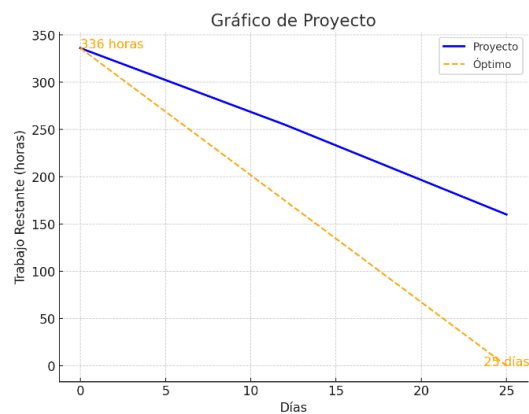
Gráfico de trabajo pendiente de iteración I y del proyecto al finalizar la iteración



Fuente: Elaboración propia

Figura 90

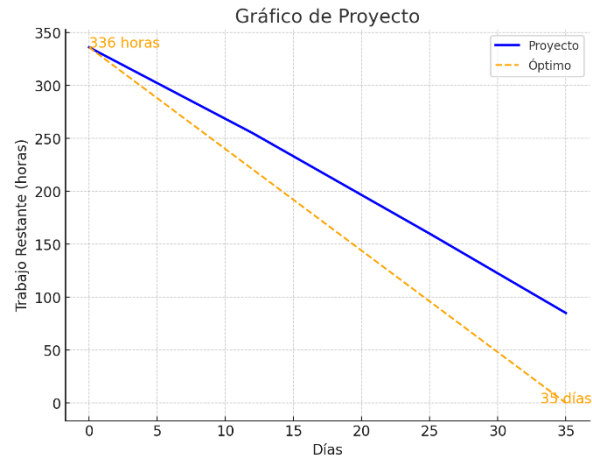
Gráfico de trabajo pendiente de iteración II y proyecto al finalizar la iteración.



Fuente: Elaboración propia

Figura 91

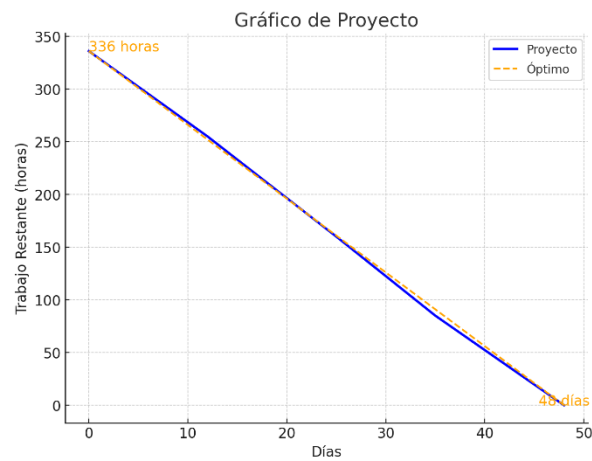
Gráfico de trabajo pendiente de iteración III y del proyecto al finalizar la iteración



Fuente: Elaboración propia

Figura 92

Gráfico de trabajo pendiente de iteración IV y del proyecto al finalizar la iteración



Fuente: Elaboración propia

4.2.3.1 REVISIÓN DE ITERACIONES

Tras culminar el proyecto, se llevó a cabo la revisión de las iteraciones con el objetivo de verificar el correcto funcionamiento de las iteraciones, todo a cargo del propietario y lo asociados al proyecto.

En estas reuniones se corrobora el funcionamiento exitoso de cada iteración al propietario del producto y al facilitador, los resultados fueron aprobados tras la relación de los requisitos para el funcionamiento de cada iteración.

4.2.3.2 RETROSPECTIVA DE ITERACIÓN

En esta sección, revisaremos mediante una evaluación sobre las iteraciones para analizar el desempeño del Equipo Scrum y examinar para la mejora de próximos proyectos, para ellos se plantean tres preguntas:

A. ¿Que se hizo bien?

- ✓ Se completaron las historias de usuario dentro del plazo visualizado
- ✓ La integración de las funcionalidades necesarias fue exitosa
- ✓ Buen plan para el desarrollo de proyecto
- ✓ El código fue modularizado correctamente

B. ¿Qué se debe mejorar?

- ✓ Mejorar la estimación de tiempo en iteraciones futuras.
- ✓ Optimizar la ejecución de tareas relacionadas con la integración.
- ✓ Automatizar procesos ya repetitivos y mejorar el entorno de pruebas
- ✓ Mejorar la coordinación en ejecución de tareas

C. ¿Qué se debe dejar de hacer?

- ✓ Evitar postergar algunas tareas que se pueden ejecutar en corto plazo.
- ✓ No omitir documentación del código
- ✓ Dejar de ignorar las tareas relacionadas con la validación de datos.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- a. La aplicación web desarrollada para la gestión de turnos del personal en el Hospital Regional de Ayacucho ha sido útil para optimizar la asignación y seguimiento de turnos, permitiendo al equipo operativo organizarse mejor, realizar la entrega de roles de turno en las fechas establecidas y reducir los errores humanos en la planificación.
- b. La automatización del proceso de programación de turnos ha automatizado el control de la entrega de roles de turno por parte de los departamentos y servicios, ha disminuido considerablemente el tiempo requerido para asignar horarios, lo que ha contribuido a la eficiencia del personal administrativo en el hospital.
- c. Se automatizó la atención de roles de turno generando bandejas de atención para las aprobaciones por las jefaturas de los departamentos, recursos humanos administración y dirección. Adicionalmente se gestionó las observaciones e identificando los roles con estados de su atención, evitando de esta manera la burocracia de los documentos físicos y los tiempos para su aprobación.
- d. La automatización de la gestión de roles de turno mejoró la calidad de servicio de uno de los procesos críticos, el cual es utilizado para el control de asistencia y pago del personal asistencial de horas por contrato y horas complementarias. Con esta automatización la programación se efectúa dentro de las fechas establecidas por norma y al estar dentro del centro de datos y verificados por sistema, se evita la impresión en papel de los roles de turno del personal. La integración efectiva del personal en el proceso de desarrollo ha permitido realizar ajustes importantes durante las iteraciones, asegurando que el sistema cumpla con las necesidades reales del hospital

- e. La metodología Scrum ha demostrado ser efectiva para organizar el desarrollo del software, aunque la ejecución de varias tareas y roles por un único individuo ha representado un reto considerable en términos de tiempo y esfuerzo.

5.2 RECOMENDACIONES

- a. Se recomienda capacitar al personal administrativo del hospital sobre el uso del sistema de gestión de turnos, especialmente para aquellos menos familiarizados con herramientas digitales, con el fin de garantizar una adopción plena y eficiente.
- b. Se sugiere implementar un sistema de soporte continuo que permita resolver rápidamente los inconvenientes técnicos que puedan surgir, asegurando que el personal pueda usar el sistema sin interrupciones.
- c. Para aumentar la seguridad del sistema, sería recomendable incluir autenticación multifactorial (MFA) en futuras versiones, reduciendo aún más los riesgos de accesos no autorizados.
- d. Considerar la implementación de reportes automáticos de turnos generados en tiempo real que puedan ser enviados a los supervisores del hospital para facilitar la supervisión y ajustes en los horarios.
- e. Para futuros proyectos, se sugiere contar con un equipo dedicado a cada rol dentro de Scrum, lo que permitiría una mejor distribución del trabajo y un cumplimiento más ágil de los plazos.

REFERENCIAS BIBLIOGRÁFICAS

- Bisquerra, R. (1989). METODOLOGÍA DE LA INVESTIGACIÓN EDUCATIVA RAFAEL BISQUERRA.pdf. Metodología de La Investigación Educativa, 58(58), 459. https://www.academia.edu/38170554/METODOLOG%C3%8DA_DE_LA_INVESTIGACION_EDUCATIVA_RAFAEL_BISQUERRA_pdf
- Díaz, M., & Hernández, R. (2010). administración, economía, humanidades y ciencias sociales.
- Hernández Sampieri, R., Fernández Collado, C., María del Pilar Baptista Lucio, D., & Méndez Valencia Christian Paulina Mendoza Torres, S. (2014).
- Supo, F., Hugo, C., & Cavero, N. (2014). FUNDAMENTOS TEÓRICOS Y PROCEDIMENTALES DE LA INVESTIGACIÓN CIENTÍFICA EN CIENCIAS SOCIALES COMO DISEÑAR Y FORMULAR TESIS DE MAESTRÍA Y DOCTORADO.
- Batthyany, K., & Cabrera, M. (2011). Metodología de la investigación en Ciencias Sociales. Apuntes para un curso inicial. https://www.researchgate.net/publication/354099679_Metodologia_de_la_investigacion_en_Ciencias_Sociales_Apuntes_para_un_curso_inicial
- Zheng, C., & Huang, X. (2017). Learning as Changing Identity Investment in Social Practice, Discourse Analysis of a Peer Feedback Activity. *Advances in Applied Sociology*, 07(02), 64–82. <https://doi.org/10.4236/AASOCI.2017.72004>
- Bernal, C. (2016). Metodología de la investigación.
- Ochoa, A. (2018). Framework para implementar aplicaciones web en diferentes lenguajes de programación orientado a objetos, 2017 (tesis de grado). Universidad Nacional de San Cristóbal de Huamanga. <http://repositorio.unsch.edu.pe/handle/UNSCH/3294>

- Guayán Castillo, J. C. F. (2018). Sistema Web de gestión hospitalaria para mejorar la programación de citas médicas en consultorios externos del Hospital Regional Docente de Trujillo [Tesis de licenciatura, Universidad César Vallejo]. Repositorio Institucional UCV. <https://hdl.handle.net/20.500.12692/34977>
- Molina Segovia, J. E. (2016). Desarrollo e implantación del sistema de gestión de turnos para consulta externa del Hospital de Especialidades Eugenio Espejo [Tesis de licenciatura, Escuela Politécnica Nacional]. Repositorio Latinoamericano. <https://repositorioslatinoamericanos.uchile.cl/handle/2250/1054386>
- Martínez Gutiérrez, A. (2019). Implementación de un sistema web para la gestión de turnos de personal en el Hospital Universitario de Quito [Tesis de maestría, Universidad de Quito]. Repositorio Institucional UdeQ.
- Mite Labre, K. D., & Sánchez Sinche, J. E. (2020). Diseño e implementación de un sistema informático para el control y seguimiento de cuentas de pacientes para el Hospital Clínica Panamericana (CLIMESA S.A.) en Guayaquil [Tesis de licenciatura, Universidad Politécnica Salesiana]. Repositorio Institucional UPS. <https://dspace.ups.edu.ec/handle/123456789/19584>
- Minsa RM 242. (2024). Resolución Ministerial N° 242-2024/Minsa. Directiva administrativa para la programación de turnos de trabajo del profesional de la salud del ministerio Salud. <https://cdn.www.gob.pe/uploads/document/file/6152717/5431133-resolucion-ministerial-n-242-2024-minsa.pdf>
- Andrade Ruiz, E. A., & Tufiño Estévez, L. E. (2011). Análisis, diseño, implementación e implantación del sistema WEB para manejo de hospitalización, consulta externa y centro quirúrgico en la nueva Clínica Bolívar [Tesis de licenciatura, Universidad Politécnica Salesiana]. Repositorio Institucional UPS. <https://dspace.ups.edu.ec/handle/123456789/1843>

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The definitive guide to Scrum: The rules of the game*. Scrum.org. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>

Rubin, K. S. (2013). *Essential Scrum: A practical guide to the most popular agile process*. Addison-Wesley.

Martín, F. J., & Rodríguez, P. (2018). *Sistemas de gestión de turnos en entornos hospitalarios: Mejora de la eficiencia y satisfacción del personal*. Editorial Académica Española.

López, M., & González, S. (2019). *Implementación de sistemas informáticos para la gestión de turnos del personal asistencial en hospitales*. *Revista de Ingeniería de Sistemas*, 14(3), 45-52. <https://doi.org/10.1234/rev.ing.sys.2019.45>

Esposito, D. (2018). *Modern Web Development with ASP.NET Core 2: Building an end-to-end application from scratch using ASP.NET Core, React, and TypeScript*. Packt Publishing.

Microsoft. (2023). *Introduction to .NET Framework*. Microsoft Docs. <https://learn.microsoft.com/en-us/dotnet/framework/>

ANEXOS

Anexo 1. Matriz de consistencia

Título: Sistema web spa para la gestión de rol de turnos del personal asistencial, hospital regional de Ayacucho, 2024.

PROBLEMAS	OBJETIVOS	VARIABLES	MÉTODO
<p>PROBLEMA GENERAL</p> <p>¿Cómo automatizar la gestión de rol de turnos del personal asistencial en el Hospital Regional de Ayacucho, 2024?</p> <p>PROBLEMAS ESPECÍFICOS</p> <p>a. ¿De qué manera automatizar el control de rol de turnos del personal asistencial</p>	<p>OBJETIVO GENERAL</p> <p>Desarrollar un sistema web SPA para la gestión de rol de turnos del personal asistencial en el Hospital Regional de Ayacucho</p> <p>OBJETIVOS ESPECÍFICOS</p> <p>a. Explorar, planificar e iterar para automatizar el control de rol de turnos del personal asistencial del Hospital Regional de Ayacucho, 2024.</p>	<p>VARIABLE DE INTERÉS</p> <p>X: Gestión de rol de turnos</p> <p>VARIABLES DESCRIPTIVAS</p> <p>X1: Control de rol de turnos</p> <p>X2: Atención de rol de turnos</p> <p>X3: Calidad del servicio</p>	<p>TIPO DE INVESTIGACIÓN</p> <p>Observacional, prospectivo, transversal y descriptivo</p> <p>NIVEL DE INVESTIGACIÓN</p> <p>Descriptivo</p> <p>DISEÑO</p> <p>No experimental, prospectivo, transversal</p>

<p>del Hospital Regional de Ayacucho?</p> <p>b. ¿De qué manera automatizar la atención de rol de turnos del personal asistencial del Hospital Regional de Ayacucho?</p> <p>c. ¿Cómo mejorar la calidad del servicio al personal asistencial del Hospital Regional de Ayacucho?</p>	<p>b. Explorar, planificar e iterar para automatizar la atención de rol de turnos del personal asistencial del Hospital Regional de Ayacucho, 2024.</p> <p>c. Explorar, planificar e iterar para brindar calidad de servicio al personal asistencial del Hospital Regional de Ayacucho, 2024.</p>		<p>POBLACIÓN</p> <p>La población está compuesta por un total de 1,500 miembros del personal asistencial del Hospital Regional de Ayacucho.</p> <p>MUESTRA</p> <p>Se seleccionó una muestra representativa de 306 miembros del personal asistencial.</p> <p>TÉCNICA</p> <p>Entrevista.</p> <p>INSTRUMENTO</p> <p>Guía de entrevista.</p>
--	---	--	---

Anexo 2. Instrumento de recolección para la gestión de turnos del personal operativo

El presente instrumento permite registrar datos para la gestión de roles de turnos del personal operativo en el Hospital Regional de Ayacucho. Las variables han sido definidas para evaluar el control de roles, la atención y calidad del servicio.

Variable: Gestión de rol de turnos	Pregunta
Dimensión: Control de rol de turnos	¿Cuántos roles de turno se crean en una semana?
	¿Cuántos roles de turno son recibidos y confirmados por el personal operativo?
	¿Cuál es la cantidad de roles de turno distribuidos correctamente entre el personal?
Dimensión: Atención de rol de turnos	¿Cuánto tiempo toma la creación de un rol de turno desde la solicitud hasta la generación?
	¿Cuánto tiempo pasa entre la creación y la aprobación del rol de turno?
	¿Cuánto tiempo toma resolver cualquier problema identificado en los roles de turno?
Dimensión: Calidad del servicio	¿Qué tan satisfecho está el personal operativo con la gestión de sus turnos?

Anexo 3. Certificado de validez de contenido del instrumento que mide la gestión de roles de turnos del personal operativo

Señor juez o experto, se requiere su colaboración para determinar la validez de contenido del instrumento de recolección de datos sobre la gestión de roles de turnos del personal operativo. Usted puede sugerir retirar ítems, incorporar ítems o cambiar el contenido de un ítem. Se agradece por su colaboración.

N°	DIMENSIONES / Ítems	Relevancia ¹		Representatividad ²		Claridad ³		Sugerencias / Observaciones
		Sí	No	Sí	No	Sí	No	
	Dimensión: Control de rol de turnos							
1	¿Cuántos roles de turno se crean en una semana?							
2	¿Cuántos roles de turno son recibidos y confirmados por el personal operativo?							
3	¿Cuál es la cantidad de roles de turno distribuidos correctamente entre el personal?							
	Dimensión: Atención de rol de turnos							
4	¿Cuánto tiempo toma la creación de un rol de turno							

	desde la solicitud hasta la generación?							
5	¿Cuánto tiempo pasa entre la creación y la aprobación del rol de turno?							
6	¿Cuánto tiempo toma resolver cualquier problema identificado en los roles de turno?							
Dimensión: Calidad del servicio								
7	¿Qué tan satisfecho está el personal operativo con la gestión de sus turnos?							

¹**Relevancia:** El ítem corresponde al concepto teórico formulado.

²**Representatividad:** El ítem es apropiado para representar al componente o dimensión específica del constructo.

³**Claridad:** Se entiende sin dificultad alguna el enunciado del ítem, es conciso, exacto y directo.

Nota: Suficiencia, se dice suficiencia cuando los ítems planteados son suficientes para medir la dimensión.



ACTA DE SUSTENTACIÓN DE TESIS N° 025-2025-FIMGC

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

En la Universidad Nacional de San Cristóbal de Huamanga de la ciudad de Ayacucho, en cumplimiento a la **RESOLUCIÓN DECANAL No 201-2025-FIMGC-D**, a los **once días del mes de agosto 2025**, siendo las **4:00 p.m.**, reunidos en el **Auditorio de la Escuela Profesional de Ingeniería de Minas**, bajo la presidencia del **MSc. Ing. José Ernesto Estrada Cárdenas** y **Mg. Edith Felicitas Guevara Morote**, y los miembros: **Mg. Hubner JANAMPA PATILLA** y **Mg. Richard ZAPATA CASAVERDE** actuando como secretario docente el **MSc. Ing. Saul Walter RETAMOZO FERNÁNDEZ**, para proceder a la sustentación de tesis para optar el **Título Profesional de Ingeniero de Sistemas**, del bachiller:

CRISTIAN POZO ANCHO

Quien presentó la tesis denominada:

SISTEMA WEB SPA PARA LA GESTIÓN DE ROL DE TURNOS DEL PERSONAL ASISTENCIAL, HOSPITAL REGIONAL DE AYACUCHO, 2024

Los señores miembros del jurado, luego de expuesta la tesis y absueltas las preguntas, deliberaron y declararon:

Aprobado con dieciséis (16)

Siendo las **5:00 p.m.** del día **11 de agosto de 2025**, culmina el acto de sustentación de tesis, y en conformidad con lo actuado, los miembros del jurado firman al pie del presente.

MSc. Ing. José Ernesto Estrada Cárdenas
Presidente

Mg. Edith Felicitas Guevara Morote
Presidenta

Mg. Hubner JANAMPA PATILLA
Miembro

Mg. Richard ZAPATA CASAVERDE
Miembro - Asesor

MSc. Saul Walter RETAMOZO FERNANDEZ
Secretario docente de la FIMGC



CONSTANCIA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN

CONSTANCIA N° 024-2025-KPS-FIMGC/UNSCH

El que suscribe; responsable verificador de originalidad de trabajos de tesis de pregrado con el software Turnitin, en segunda instancia para las **Escuelas Profesionales** de la **Facultad de Ingeniería de Minas, Geología y Civil**; en cumplimiento a la **Resolución de Consejo Universitario N° 039-2021-UNSCH-CU**, Reglamento de Originalidad de Trabajos de Investigación de la Universidad Nacional San Cristóbal de Huamanga y **Resolución Decanal N° 697-2024-FIMGC-D**, deja constancia de originalidad de trabajo de investigación, que el/la Sr./Srta.

Nombres y Apellidos : Cristian Pozo Ancho
Escuela Profesional : INGENIERÍA DE SISTEMAS
Título de la Tesis : Sistema web spa para la gestión de rol de turnos del personal asistencial, hospital regional de Ayacucho, 2024
Evaluación de la Originalidad : 0% Índice de Similitud
Identificador de la entrega : 2751896824

Por tanto, según los Artículos 12, 13 y 17 del Reglamento de Originalidad de Trabajos de Investigación, es **PROCEDENTE** otorgar la **Constancia de Originalidad** para los fines que crea conveniente.

En señal de conformidad y verificación se firma la presente constancia

Ayacucho, 24 de setiembre de 2025



Firmado digitalmente por:
PERALTA SOTOMAYOR Karel
FAU 20143680754 soft
Motivo: Soy el autor del documento
Fecha: 24/09/2025 08:12:39-0500

SISTEMA WEB SPA PARA LA GESTIÓN DE ROL DE TURNOS DEL PERSONAL ASISTENCIAL, HOSPITAL REGIONAL DE AYACUCHO, 2024

por Cristian Pozo Ancho

Fecha de entrega: 15-sept-2025 12:54p. m. (UTC-0500)

Identificador de la entrega: 2751896824

Nombre del archivo: MEMORANDO_N_497-2025-CERTIFICADO_DE_ORIGINALIDAD-
_CRISTIAN_POZO_ANCHO.pdf (3.89M)

Total de palabras: 24021

Total de caracteres: 135599

SISTEMA WEB SPA PARA LA GESTIÓN DE ROL DE TURNOS DEL PERSONAL ASISTENCIAL, HOSPITAL REGIONAL DE AYACUCHO, 2024

INFORME DE ORIGINALIDAD

0%

INDICE DE SIMILITUD

0%

FUENTES DE INTERNET

0%

PUBLICACIONES

0%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

Excluir citas

Activo

Excluir coincidencias < 30%

Excluir bibliografía

Activo