



# Implementación de Sistema de Gestión de Equipos Informáticos para la Gobernación de Sucre

Jose Alfonso Mercado Samur

Trabajo de grado para optar al título de Ingeniero Electrónico  
Modalidad de Practica Cursada  
Semestre de Industria

Maria Katherine Plazas Olaya, Maestría en Ingeniería.  
Carlos Mauricio Diaz Royerth, Ingeniero de Sistemas.

Universidad de Antioquia  
Facultad de Ingeniería  
Ingeniería Electrónica  
Medellín  
2024

---

<b>Cita</b>	(Mercado Samur, 2024)
<b>Referencia</b>	(Mercado Samur, J. A. 2024). <i>Implementación de Sistema de Gestión de Equipos Informáticos para la Gobernación de Sucre</i> . Semestre de Industria. Universidad de Antioquia, Medellín.
<b>Estilo APA 7 (2020)</b>	

---



Centro de Documentación de Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - [www.udea.edu.co](http://www.udea.edu.co)

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

## Agradecimientos

Con el corazón lleno de gratitud y alegría, quiero mencionar a todas las personas que han iluminado mi camino, guiando mis pensamientos hasta este maravilloso momento.

Gracias a quienes me han dado el don de la autocrítica y la habilidad de desenvolverme en el mundo con confianza. A toda mi familia, Keren Mendoza, Joseph Mercado, quienes son la fuente de mi fuerza y felicidad: mi madre y mi padre, que han entregado su vida para ver florecer mis sueños; mi hermano, mi faro desde la infancia, cuyo ejemplo siempre he seguido con admiración.

A la Universidad de Antioquia, un espacio mágico donde se abrieron las puertas de mi crecimiento académico, ¡gracias por acogerme en tu cálido abrazo! A todos los profesores que llenaron de sabiduría cada rincón de mi mente, especialmente aquellos que, con su pasión y entrega, dejaron una huella imborrable en mi formación. Profesor Norman Cesar Mercado, quien fue un foco en medio de una tormenta emocional que zarpaba en su momento, le agradezco su apoyo y comprensión en esos momentos difíciles. Profesor Álvaro Octavio Gaviria, de quien aprendí que la humildad es el verdadero camino del conocimiento, cuánta razón hay en Descartes, solo sabemos que no sabemos nada.

Todos mis amigos que conocí durante mi permanencia en la Universidad, David Vélez, Yeiner Pájaro, amigos de vida y compañeros de trabajo que recordaré por siempre.

Todas las personas que me acompañaron durante mi estancia en la Gobernación de Sucre, tiempo en el que me sentía honrado de poder servir a mi departamento a través de mi colaboración. Asesor externo Carlos Diaz, Isaac Paternina, David Berrio, su humanidad y deseo de ayudar me inspiran.

A mi querida universidad, que tanto admiro, y al invaluable apoyo del profesor y coordinador Adrián Montoya Lince, así como de la profesora y asesora interna María Katherine Plazas, ¡gracias por acompañarme y brindarme su inagotable apoyo!

# Índice general

<b>1. Resumen</b>	<b>9</b>
1.1. Introducción . . . . .	9
1.2. Objetivos . . . . .	10
1.3. Marco Teórico . . . . .	11
1.3.1. Gestión de Activos de TI . . . . .	11
1.3.2. Mockups y Diseño de Interfaz de Usuario . . . . .	11
1.3.3. Seguridad Informática . . . . .	12
1.4. Metodología . . . . .	13
<b>2. Análisis de Requerimientos</b>	<b>14</b>
2.1. Objetivos del Sistema . . . . .	14
2.2. Requerimientos Funcionales y No Funcionales . . . . .	14
2.2.1. Requerimientos Funcionales . . . . .	14
2.2.2. Requerimientos No Funcionales . . . . .	15
<b>3. Diseño del Sistema</b>	<b>17</b>
3.1. Arquitectura del Software . . . . .	17
3.2. Backend . . . . .	17
3.2.1. controllers . . . . .	18
3.2.1.1. accountcontroller . . . . .	18
3.2.1.2. invpficocontroller . . . . .	20
3.2.1.3. invppalcontroller . . . . .	20
3.2.1.4. permisoscontroller . . . . .	20
3.2.1.5. pficohistorialcontroller . . . . .	20
3.2.1.6. ppalhistorialcontroller . . . . .	25
3.2.1.7. regfuncontroller . . . . .	25
3.2.1.8. reghardppalcontroller . . . . .	27
3.2.1.9. regpfricocontroller . . . . .	27
3.2.1.10. secretariascontroller . . . . .	27
3.2.2. database . . . . .	27
3.2.3. handlerMe . . . . .	30
3.2.3.1. accounthandler . . . . .	30
3.2.3.2. imghandler . . . . .	32

3.2.3.3.	invpficohandler . . . . .	32
3.2.3.4.	invppalhandler . . . . .	32
3.2.3.5.	permisoshandler . . . . .	32
3.2.3.6.	pficohistorialhandler . . . . .	33
3.2.3.7.	ppalhistorialhandler . . . . .	33
3.2.3.8.	regfunhandler . . . . .	37
3.2.3.9.	reghardppalhandler . . . . .	37
3.2.3.10.	regpfricohandler . . . . .	41
3.2.3.11.	secretariashandler . . . . .	42
3.2.4.	migracion . . . . .	42
3.2.5.	models . . . . .	43
3.2.6.	plantillaDeDatos . . . . .	46
3.2.7.	repository . . . . .	46
3.2.8.	src . . . . .	47
3.2.9.	utils . . . . .	47
3.2.10.	main.go . . . . .	49
3.3.	FrontEnd . . . . .	50
3.3.1.	database . . . . .	58
3.3.2.	form . . . . .	60
3.3.3.	register . . . . .	60
3.3.4.	userpermits . . . . .	62
3.3.5.	usuario . . . . .	62
3.4.	Diagramas de Flujo . . . . .	62
3.4.1.	Permisos de Usuario . . . . .	62
3.4.2.	Registro de Usuarios . . . . .	65
3.4.3.	Registro de Funcionarios, Hardware y Asignaciones . . . . .	66
3.5.	Mockup de la Interfaz de Usuario . . . . .	67
<b>4.</b>	<b>Desarrollo e Implementación</b>	<b>69</b>
4.1.	Tecnologías Utilizadas . . . . .	69
4.2.	Configuración Inicial . . . . .	71
4.2.1.	Cuenta Administradora . . . . .	71
4.2.2.	Recomendación de Seguridad . . . . .	71
4.3.	Página de Inicio de Sesión . . . . .	72
4.4.	Página de Registro de Usuarios . . . . .	72
4.5.	Página Principal . . . . .	74
4.5.1.	Perfil . . . . .	74
4.5.2.	Registrar . . . . .	75
4.5.2.1.	Registrar Funcionario . . . . .	75
4.5.2.2.	Registrar Hardware Principal . . . . .	76
4.5.2.3.	Registrar Hardware Periférico . . . . .	76
4.5.3.	Asignaciones . . . . .	77
4.5.3.1.	Asignación de Hardware Principal . . . . .	78

4.5.3.2.	Asignación de Hardware Periférico . . . . .	78
4.5.4.	Consultar . . . . .	80
4.5.4.1.	Consultar funcionario . . . . .	80
4.5.4.2.	Consultar Hardware principal . . . . .	80
4.5.4.3.	Consultar Hardware Periférico . . . . .	81
4.5.4.4.	Base de Datos . . . . .	82
<b>5.</b>	<b>Pruebas y Ajustes</b>	<b>84</b>
5.1.	Proceso de Pruebas . . . . .	84
5.1.1.	Preparación de Casos de Prueba . . . . .	84
5.1.2.	Ejecución de Pruebas . . . . .	84
5.1.2.1.	Creación de Permisos de Cuenta de Usuario . . . . .	84
5.1.2.2.	Registro de una Cuenta de Usuario . . . . .	85
5.1.2.3.	Inicio de Sesión con Cuenta de Usuario . . . . .	86
5.1.2.4.	Consulta de Funcionario . . . . .	87
5.2.	Errores Encontrados . . . . .	87
5.2.1.	Errores Críticos . . . . .	87
5.2.2.	Errores Menores . . . . .	87
5.3.	Correcciones Realizadas . . . . .	88
5.3.1.	Corrección de Errores Críticos . . . . .	88
5.3.2.	Corrección de Errores Menores . . . . .	88
5.4.	Preparación para el Despliegue . . . . .	88
5.4.1.	Validación Final . . . . .	88
5.4.2.	Documentación . . . . .	88
5.4.3.	Despliegue . . . . .	89
<b>6.</b>	<b>Despliegue</b>	<b>90</b>
6.1.	Configuración del Entorno de Producción . . . . .	90
6.2.	Fase de Pruebas Realizadas por Usuarios Finales . . . . .	91
6.2.1.	Selección de Usuarios Finales . . . . .	91
6.2.2.	Ejecución de Pruebas . . . . .	91
6.2.3.	Metodología de Pruebas . . . . .	91
6.3.	Informe de Retroalimentación . . . . .	91
6.3.1.	Recopilación de Retroalimentación . . . . .	91
6.3.2.	Análisis de Retroalimentación . . . . .	92
6.3.3.	Correcciones y Mejoras Realizadas . . . . .	92
	<b>Resultados y Análisis</b>	<b>93</b>
	<b>Conclusiones</b>	<b>95</b>
	<b>Referencias</b>	<b>97</b>

# Índice de figuras

3.1. Esquema general de la arquitectura . . . . .	18
3.2. Módulos de controllers. . . . .	20
3.3. Modulo database. . . . .	27
3.4. Tablas de la base de datos. . . . .	27
3.5. Estructura del módulo handlerMe . . . . .	32
3.6. Estructura del módulo migracion . . . . .	42
3.7. Estructura del módulo models . . . . .	43
3.8. Estructura y tipos de datos de permisos de usuario . . . . .	43
3.9. Estructura y tipos de datos de un usuario . . . . .	44
3.10. Estructura y tipos de datos de secretarias . . . . .	44
3.11. Estructura y tipos de datos de un funcionario . . . . .	44
3.12. Estructura y tipos de datos de un hardware principal . . . . .	45
3.13. Estructura y tipos de datos de un periférico . . . . .	45
3.14. Estructura y tipos de datos de un hardware principal asignado . . . . .	46
3.15. Estructura y tipos de datos de un historial de hardware principal . . . . .	46
3.16. Plantilla de datos en formato Excel para generar documento . . . . .	46
3.17. Documento generado después de una asignación . . . . .	47
3.18. Directorio src con los archivos CSS, HTML y JavaScript . . . . .	48
3.19. Modulo utils . . . . .	49
3.20. Estructura general del Frontend . . . . .	58
3.21. Estructura general del archivo <code>form.js</code> . . . . .	60
3.22. Diagrama de flujo para un permiso de usuario . . . . .	65
3.23. Diagrama de flujo para un registro de usuario . . . . .	66
3.24. Diagrama de flujo para un registro de funcionario, hardware principal, periférico o asignación . . . . .	67
3.25. Mockup de la interfaz de la página principal . . . . .	68
4.1. Interfaz del registro de permisos de usuario . . . . .	72
4.2. Menú de una cuenta administradora y cuenta de usuario . . . . .	73
4.3. Página de inicio de sesión . . . . .	73
4.4. Interfaz para el registro de usuarios . . . . .	74
4.5. Interfaz de la página principal . . . . .	74
4.6. Página de actualización de información de cuentas de usuario . . . . .	75
4.7. Formulario para el registro de funcionarios . . . . .	76

4.8. Formulario para el registro de hardware principal . . . . .	77
4.9. Formulario para el registro de hardware periférico . . . . .	78
4.10. Formulario para la asignación de hardware principal . . . . .	79
4.11. Formulario para la asignación de hardware periférico . . . . .	79
4.12. Formulario para consultar un funcionario . . . . .	81
4.13. Formulario para consultar un hardware principal . . . . .	81
4.14. Formulario para consultar un hardware periférico . . . . .	82
4.15. Consulta de base de datos y filtrado de la información . . . . .	83
4.16. Descarga de tabla empleados de la base de datos . . . . .	83
5.1. Registro de permisos de una cuenta de usuario . . . . .	85
5.2. Registro de una cuenta de usuario . . . . .	86
5.3. Sesión de una cuenta en su rol usuario . . . . .	86
5.4. Consulta de funcionario con asignaciones de diferentes tipos de hardware	87

# Índice de tablas

3.1. Funciones principales modulo accountcontroller . . . . .	19
3.2. Funciones principales modulo invpficocontroller . . . . .	21
3.3. Funciones principales modulo invppalcontroller . . . . .	22
3.4. Funciones principales modulo permisoscontroller . . . . .	23
3.5. Funciones principales modulo pficohistorialcontroller . . . . .	24
3.6. Funciones principales modulo ppalhistorialcontroller . . . . .	25
3.7. Funciones principales modulo regfuncontroller . . . . .	26
3.8. Funciones principales modulo reghardppalcontroller . . . . .	28
3.9. Funciones principales modulo regpfricocontroller . . . . .	29
3.10. Funciones principales modulo secretariascontroller . . . . .	30
3.11. Funciones principales módulo accounthandler . . . . .	31
3.12. Funciones principales módulo inghandler . . . . .	33
3.13. Funciones principales módulo pficohandler . . . . .	34
3.14. Funciones principales módulo invppalhandler . . . . .	35
3.15. Funciones principales módulo permisoshandler . . . . .	36
3.16. Funciones principales módulo pficohistorialhandler . . . . .	37
3.17. Funciones principales módulo ppalhistorialhandler . . . . .	38
3.18. Funciones principales módulo regfunhandler . . . . .	39
3.19. Funciones principales módulo reghardppalhandler . . . . .	40
3.20. Funciones principales módulo regpfricohandler . . . . .	41
3.21. Funciones principales módulo secretariashandler . . . . .	42
3.22. Funciones principales módulo repository . . . . .	48
3.23. Funciones principales módulo utils . . . . .	49
3.24. Endpoint estáticos . . . . .	50
3.25. Endpoint para servir IP del servidor . . . . .	50
3.26. Endpoints para el inicio de sesión . . . . .	51
3.27. Endpoints de manejo de usuarios . . . . .	52
3.28. Endpoints para los permisos de usuario . . . . .	53
3.29. Endpoints para secretarías . . . . .	53
3.30. Endpoints relacionados a funcionarios . . . . .	54
3.31. Endpoints de hardware principales . . . . .	54
3.32. Endpoints de asignación de hardware principal . . . . .	55
3.33. Endpoints asignaciones de periféricos . . . . .	56

3.34. Endpoints de exportación de base de datos . . . . .	57
3.35. Funciones principales de database.js . . . . .	59
3.36. Funciones principales de form.js . . . . .	60
3.37. Funciones principales de register.js . . . . .	61
3.38. Funciones principales de userpermits.js . . . . .	63
3.39. Funciones principales de usuario.js . . . . .	64

# Capítulo 1

## Resumen

El proyecto se centró en la implementación de un Sistema de Gestión de Equipos Informáticos para la Gobernación de Sucre. El problema identificado era la falta de un sistema centralizado para la administración de los recursos tecnológicos, lo que causaba demoras y falta de accesibilidad a la información. El sistema desarrollado permitió gestionar en tiempo real los inventarios de equipos, asignaciones a funcionarios y la gestión de información crítica como IPs estáticas de dispositivos clave. Se emplearon lenguajes como HTML, CSS, JavaScript y Go, junto con bases de datos relacionales. Los resultados mostraron una notable mejora en la eficiencia, la coordinación del Área de Sistemas y la seguridad de la información, logrando una administración más organizada de los recursos tecnológicos.

### 1.1. Introducción

La tecnología desempeña un papel esencial en el funcionamiento de las organizaciones públicas, siendo los equipos informáticos uno de los activos clave para asegurar la eficiencia operativa. En la Gobernación de Sucre, la administración de los equipos tecnológicos ha sido un desafío creciente debido al aumento del número de dispositivos y usuarios a lo largo del tiempo. Hasta hace poco, la gestión de estos equipos se realizaba mediante una plantilla Excel, una solución limitada que no permitía un control eficiente ni una adecuada protección de la información. Este enfoque manual dificultaba la actualización en tiempo real del inventario, generando problemas como la falta de acceso rápido a la información, errores en la asignación de equipos y una mayor vulnerabilidad en la seguridad de los datos críticos.

Con el crecimiento constante de la infraestructura tecnológica y el número de funcionarios, se volvió indispensable un sistema que permitiera la centralización y automatización de la gestión de los recursos informáticos. En este contexto, se planteó el desarrollo de un Sistema de Gestión de Equipos Informáticos, una solución que no solo permitiría mejorar la organización y el control de los activos de TI, sino que también facilitaría la asignación de equipos a los funcionarios, el seguimiento de incidencias, y el

almacenamiento seguro de información sensible como las IPs estáticas de dispositivos clave.

El presente proyecto se enfocó en desarrollar una plataforma web [8] que permita a la Gobernación de Sucre contar con un sistema integrado y centralizado, diseñado específicamente para mejorar la eficiencia en la gestión de los equipos informáticos. Se utilizó una combinación de tecnologías modernas, como HTML, CSS, JavaScript y Go, junto con bases de datos relacionales para almacenar y procesar la información de manera segura y estructurada. El desarrollo del sistema incluyó desde la fase de análisis y recolección de requerimientos hasta el diseño de la arquitectura del software y la interfaz de usuario, con el objetivo de crear una herramienta intuitiva y funcional.

El impacto de este sistema en la Gobernación se evidencia en una gestión más organizada, un acceso rápido a los datos de inventario y asignaciones, y una mejora significativa en la seguridad de la información. La automatización de procesos como el registro de equipos y la generación de reportes ha reducido el tiempo de respuesta del área de sistemas ante solicitudes de los usuarios, mejorando la calidad del servicio y la eficiencia operativa de la organización.

## 1.2. Objetivos

- **Objetivo general:** Desarrollar un sistema de gestión de equipos informáticos que permita una asignación y control eficiente de los recursos tecnológicos de la Gobernación de Sucre, usando bases de datos relacionales para el almacenamiento de la información y lenguajes informáticos HTML, CSS, JavaScript y Golang para la interfaz de usuario.
- **Objetivos específicos:**
  1. Identificar y definir los requisitos necesarios para la gestión de equipos informáticos en la Gobernación de Sucre.
  2. Establecer un diseño detallado y funcional del sistema de gestión de equipos informáticos, que incluya la arquitectura del software, el mockup de la interfaz de usuario y funcionalidades que aborden los requisitos identificados.
  3. Desarrollar las funcionalidades definidas en el diseño del sistema de gestión de equipos, utilizando bases de datos relacionales y una interfaz web en lenguajes informáticos HTML, CSS, JavaScript y Golang.
  4. Realizar pruebas del sistema para identificar posibles errores o problemas de funcionamiento y ejecutar los correctivos que sean necesarios para lograr el cumplimiento de los requerimientos previamente establecidos.
  5. Desplegar el sistema en el entorno de producción de la Gobernación de Sucre y hacer pruebas con usuarios finales capacitados para evaluar el funcionamiento de la gestión de recursos.

## 1.3. Marco Teórico

El presente proyecto se fundamenta en tres pilares conceptuales: la gestión de activos de TI, el diseño de interfaces mediante mockups y la seguridad informática. A continuación, se describen en profundidad estos conceptos y su relevancia para el desarrollo del Sistema de Gestión de Equipos Informáticos para la Gobernación de Sucre.

### 1.3.1. Gestión de Activos de TI

La gestión de activos de TI (Information Technology Asset Management, ITAM) es un conjunto de prácticas enfocadas en optimizar la administración del ciclo de vida de los recursos tecnológicos de una organización. Estos activos incluyen tanto hardware como software, y su gestión abarca desde la adquisición, implementación y mantenimiento, hasta su disposición final. El objetivo principal de la gestión de activos de TI es maximizar el valor de los recursos tecnológicos, asegurando que se utilicen de manera eficiente y que estén alineados con los objetivos organizacionales.

La correcta gestión de activos de TI permite tener visibilidad en tiempo real del estado y ubicación de los equipos, lo que reduce los riesgos asociados a la pérdida o mal uso de estos. Además, un sistema de este tipo facilita la planeación de futuras adquisiciones, la asignación de recursos, y el seguimiento del rendimiento de los equipos, lo que resulta crucial para garantizar la continuidad operativa de una organización. En el caso específico de la Gobernación de Sucre, la implementación de un sistema de gestión de equipos informáticos optimiza la supervisión y el control de sus recursos, permitiendo a los responsables del área de sistemas tomar decisiones informadas basadas en datos actualizados y precisos.

La International Association of IT Asset Managers (IAITAM) establece que una gestión de activos eficiente debe contemplar la planificación, adquisición, despliegue, mantenimiento y disposición final de los recursos, siempre garantizando el cumplimiento de normativas y regulaciones de la industria [5]. Este proceso sistemático asegura que los recursos tecnológicos sean utilizados de forma óptima, generando un retorno de inversión medible para la organización [2].

### 1.3.2. Mockups y Diseño de Interfaz de Usuario

Un mockup es una representación visual de la interfaz de un sistema o aplicación, que muestra su estructura y disposición de elementos antes de su desarrollo. Los mockups son herramientas clave dentro del proceso de diseño de interfaces de usuario (UI), ya que permiten a los desarrolladores y partes interesadas visualizar y validar la apariencia y funcionalidad del sistema antes de escribir código. El uso de mockups reduce el riesgo de errores costosos en fases posteriores del desarrollo, facilitando una retroalimentación temprana sobre aspectos como la usabilidad, la disposición de los componentes visuales y la coherencia con los requerimientos del usuario [6].

En el contexto del desarrollo del sistema para la Gobernación de Sucre, los mockups ayudaron a definir el flujo de interacción de los usuarios con la plataforma, optimizando la disposición de los elementos clave como la gestión de inventarios y la asignación de equipos a funcionarios. El diseño de interfaces debe enfocarse en ofrecer una experiencia intuitiva y accesible para el usuario, asegurando que las funcionalidades del sistema sean fácilmente navegables y entendibles. Por lo tanto, los mockups no solo permitieron prever la disposición de las pantallas, sino también facilitaron la validación de la experiencia del usuario, lo que es crucial para que los usuarios finales (en este caso, el personal de la Gobernación) adopten la plataforma de manera eficaz.

Además, el uso de mockups en el proceso de diseño minimiza la brecha entre las expectativas de los usuarios y el producto final, ya que permite realizar ajustes visuales o funcionales basados en las necesidades detectadas durante las fases iniciales del proyecto. En este sentido, el enfoque iterativo basado en prototipos permite un desarrollo más ágil y centrado en el usuario, aspecto crucial en proyectos de tecnología.

### **1.3.3. Seguridad Informática**

La seguridad informática es un campo que se encarga de proteger la integridad, confidencialidad y disponibilidad de la información, especialmente en sistemas digitales. Con el incremento de las amenazas cibernéticas, asegurar la información es un requisito fundamental para cualquier organización, y aún más en el sector público, donde se manejan datos sensibles y críticos [3]. En el desarrollo del Sistema de Gestión de Equipos Informáticos, la seguridad es uno de los aspectos prioritarios, tanto a nivel de acceso como de almacenamiento de la información.

Uno de los componentes esenciales de la seguridad informática es el control de acceso basado en roles (RBAC, Role-Based Access Control). Este modelo permite restringir el acceso a los recursos del sistema en función del rol del usuario, garantizando que solo los usuarios autorizados puedan acceder a cierta información o realizar determinadas acciones [7]. Para la Gobernación de Sucre, esto es particularmente importante dado que los datos sobre los equipos asignados, así como la información crítica de las IPs estáticas de dispositivos, deben estar restringidos a personal autorizado para evitar usos indebidos o modificaciones no autorizadas.

Otro aspecto clave en la seguridad del sistema es la implementación de mecanismos de encriptado para proteger la información sensible, como las contraseñas. El algoritmo bcrypt para el hash de las contraseñas, garantiza que estas no puedan ser recuperadas en caso de un acceso no autorizado a la base de datos [1]. Además, protocolos seguros en las comunicaciones entre el frontend y el backend, minimizan el riesgo de ataques como el man-in-the-middle [4].

## 1.4. Metodología

La implementación del sistema de gestión de equipos informáticos se desarrolló en varias etapas: análisis de los requisitos, diseño del sistema, desarrollo e implementación, pruebas y ajustes, y finalmente, capacitación y despliegue. A continuación, se describen cada una de estas fases:

- **Capítulo 2:** Se definieron los objetivos del sistema y se identificaron los requerimientos funcionales y no funcionales necesarios para su correcto funcionamiento.
- **Capítulo 3:** Se estableció la arquitectura del software, incluyendo la estructura del backend (con componentes como controllers, database, y modelos) y del frontend (formulario de registro, gestión de usuarios, etc.). Se crearon diagramas de flujo para ilustrar procesos clave como el registro de usuarios y la gestión de permisos. Se diseñaron mockups de la interfaz de usuario para visualizar la interacción del usuario con el sistema.
- **Capítulo 4:** Se configuró el entorno inicial y se desarrollaron las distintas páginas del sistema, incluyendo la página de inicio de sesión, registro de usuarios, y gestión de asignaciones de hardware. Se utilizaron tecnologías específicas para garantizar una implementación efectiva.
- **Capítulo 5:** Se prepararon y ejecutaron casos de prueba para validar el funcionamiento del sistema, identificando y corrigiendo errores críticos y menores. Se realizó una preparación final para el despliegue, incluyendo documentación y validación.
- **Capítulo 6:** Se configuró el entorno de producción y se llevaron a cabo pruebas con usuarios finales, recopilando y analizando su retroalimentación para realizar mejoras. Se confirmó la efectividad y eficiencia del sistema, asegurando que cumpliera con las expectativas establecidas. Cada una de estas fases fue crucial para el desarrollo de un sistema robusto y funcional, asegurando que se abordaran todas las necesidades y requerimientos del usuario.

# Capítulo 2

## Análisis de Requerimientos

### 2.1. Objetivos del Sistema

Desarrollar un sistema de gestión de equipos informáticos que permita asignar y controlar los recursos tecnológicos de la Gobernación de Sucre, con bases de datos relacionales y tecnologías web.

### 2.2. Requerimientos Funcionales y No Funcionales

#### 2.2.1. Requerimientos Funcionales

- **Gestión de Inventarios:**

- Los usuarios deben poder registrar equipos informáticos con detalles como número de serie, especificaciones técnicas, estado y ubicación.
- Visualización y búsqueda avanzada de equipos utilizando diferentes criterios de búsqueda como el nombre de la base de datos que se desea consultar, como también una palabra clave a buscar dentro de la misma.
- Los usuarios deben poder editar y eliminar los equipos registrados en función de sus permisos.

- **Asignación de Equipos:**

- Los administradores deben poder asignar y reasignar equipos a los funcionarios.
- Registro automático de un historial de asignaciones para cada equipo, con detalles de las fechas y los funcionarios involucrados.

- **Gestión de Usuarios:**

- Los usuarios deben poder registrarse, iniciar sesión y cerrar sesión, como también consultar o visualizar la información suministrada por las bases de datos, pero no la modificación, únicamente consultas.
- Los administradores deben poder agregar, eliminar y modificar usuarios en el sistema.

■ **Reportes y Análisis:**

- Generación de reportes personalizados sobre el estado actual de los equipos, las asignaciones realizadas, y el historial de uso.
- Análisis del rendimiento de los equipos en función del tiempo de uso y las asignaciones.

## 2.2.2. Requerimientos No Funcionales

■ **Rendimiento y Escalabilidad:**

- El sistema debe garantizar un rendimiento óptimo con tiempos de respuesta rápidos, incluso cuando el número de usuarios o equipos a gestionar aumente considerablemente. Se espera que el sistema maneje de forma eficiente consultas en la base de datos relacional y la generación de reportes en tiempo real, con capacidad para escalar a medida que la Gobernación de Sucre aumente su infraestructura tecnológica.

■ **Seguridad:**

- La seguridad de la información es crítica, dado que el sistema maneja datos sensibles como la información personal de cada funcionario, asignaciones de equipos, IP estáticas de algunos dispositivos. Por ello, se requiere un sistema de autenticación y autorización basado en roles (administradores y usuarios) para garantizar que solo personal autorizado tenga acceso a funciones específicas.

■ **Usabilidad y Accesibilidad:**

- El sistema debe ser intuitivo y fácil de usar para los funcionarios de la Gobernación, sin necesidad de conocimientos técnicos avanzados. Se requiere que la interfaz sea diseñada con HTML, CSS y JavaScript para ser accesible y navegable, proporcionando una experiencia de usuario fluida tanto en dispositivos móviles como en ordenadores de escritorio mediante un diseño responsivo.

■ **Mantenibilidad y Extensibilidad:**

- El código del sistema debe ser modular y estar bien documentado para facilitar su mantenimiento y futuras extensiones. El uso de tecnologías como Golang en el backend y PostgreSQL como sistema de gestión de bases de datos relacionales facilita la implementación de mejoras y nuevas funcionalidades sin comprometer el rendimiento.
- **Disponibilidad y Confiabilidad:**
- El sistema debe estar disponible en todo momento, especialmente durante las horas laborales de la Gobernación de Sucre.

# Capítulo 3

## Diseño del Sistema

Se definió la arquitectura del software, la estructura de la base de datos y se diseñó una interfaz de usuario amigable e intuitiva.

### 3.1. Arquitectura del Software

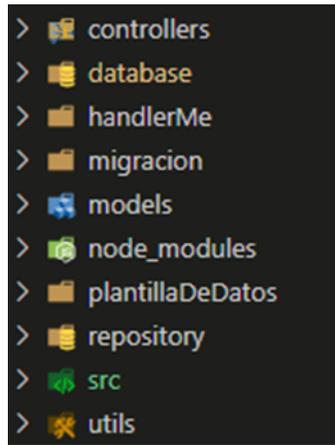
El sistema de gestión de equipos informáticos desarrollado para la Gobernación de Sucre sigue una arquitectura de software multicapa. Esta arquitectura permite una clara separación de responsabilidades, mejorando la mantenibilidad y escalabilidad del sistema. Las principales capas implementadas son:

- **Capa de Presentación:** Gestiona la interfaz de usuario, donde se interactúa con el sistema a través de un entorno web accesible y fácil de usar. Se desarrolló con tecnologías como HTML, CSS y JavaScript.
- **Capa de Lógica de Negocio:** Aquí se manejan todas las reglas de negocio y operaciones del sistema. Está implementada principalmente en Go, gestionando la lógica relacionada con la administración de inventarios, asignación de equipos y seguimiento de información crítica.
- **Capa de Acceso a Datos:** Se encarga de la interacción con la base de datos relacional, donde se almacenan todos los registros relacionados con los equipos y usuarios. Esta capa permite la centralización y seguridad de la información, asegurando la persistencia de los datos.

### 3.2. Backend

El backend del sistema fue estructurado de manera modular para facilitar la gestión y organización de las distintas responsabilidades del servidor. En la Figura 3.1 se pueden observar los diferentes directorios, cada uno encargado de tareas específicas. El directorio controllers maneja todas las operaciones relacionadas con la base de datos,

Figura 3.1: Esquema general de la arquitectura



incluyendo modificaciones y consultas. Además, genera mensajes internos en el servidor para indicar si las operaciones solicitadas por el cliente desde la interfaz de usuario se realizaron con éxito o si hubo algún error. Por su parte, el directorio handlerMe actúa como intermediario entre la interfaz de usuario y los controladores, ejecutando las acciones correspondientes en el backend y mostrando al usuario mensajes sobre el éxito o el fallo de las solicitudes, asegurando así una comunicación clara y eficiente entre ambas partes.

### 3.2.1. controllers

El directorio controllers agrupa todos los módulos encargados de ejecutar las operaciones en la base de datos y de gestionar los errores internos del servidor. En la Figura 3.2 se ilustra la estructura de este directorio, destacando cómo cada módulo cumple una función específica en el manejo de la lógica de negocio y las interacciones con la base de datos. A continuación, se describen en detalle cada uno de los módulos que componen este directorio, resaltando su propósito y las operaciones que realizan.

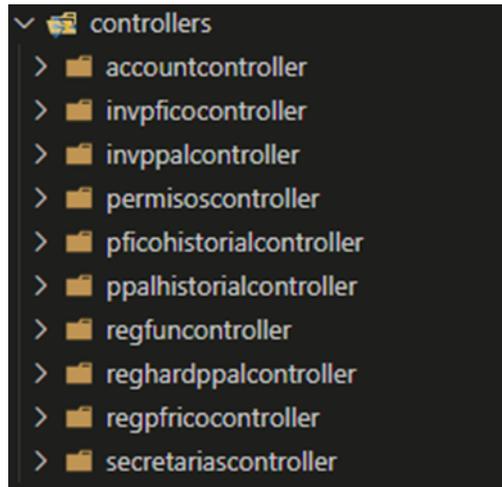
#### 3.2.1.1. accountcontroller

Este módulo gestiona el control de las cuentas de usuarios, permitiendo la creación, actualización, eliminación y consulta de los registros almacenados en la base de datos. La Tabla 3.1 muestra las principales funciones implementadas en este módulo, detallando las operaciones realizadas y su propósito. Además de interactuar directamente con la base de datos, este módulo cifra las contraseñas de los usuarios para garantizar la seguridad de los datos, y tiene la capacidad de exportar la información de las cuentas a un archivo Excel.

Tabla 3.1: Funciones principales modulo accountcontroller.

<b>Funciones</b>	<b>Descripción</b>
<b>LeerUsuarios</b>	Recupera todos los usuarios de la base de datos utilizando una consulta SQL, los convierte a formato JSON y los devuelve.
<b>LeerUnUsuario</b>	Recupera un solo usuario de la base de datos según el correo electrónico proporcionado, lo convierte a formato JSON y lo devuelve.
<b>RegistrarUsuario</b>	Registra un nuevo usuario en la base de datos después de verificar si ya existe, y cifra su contraseña antes de almacenarla.
<b>ActualizarUsuario</b>	Actualiza la información de un usuario existente en la base de datos. Si la contraseña es modificada, se cifra antes de actualizarla.
<b>construirUpdateQuery</b>	Construye una consulta SQL de actualización con los valores proporcionados.
<b>EliminarUsuario</b>	Elimina un usuario de la base de datos después de verificar su existencia. Si el usuario tiene una imagen de perfil asociada, también se elimina.
<b>ExportarUsuariosAExcel</b>	Exporta todos los usuarios de la base de datos a un archivo Excel, incluyendo encabezados y datos.

Figura 3.2: Módulos de controllers.



#### 3.2.1.2. invpficocontroller

Este módulo se encarga de gestionar las operaciones relacionadas con los periféricos asignados a los usuarios en el sistema de gestión de hardware. A través de sus funciones, es posible asignar nuevos periféricos, consultar las asignaciones existentes, eliminarlas y exportar la información de los periféricos asignados a un archivo Excel. En la Tabla 3.2 se detallan las funciones clave de este módulo, junto con sus descripciones.

#### 3.2.1.3. invppalcontroller

Este módulo gestiona las operaciones relacionadas con el hardware principal asignado a los usuarios dentro del sistema de gestión de hardware. Las funciones del módulo permiten asignar nuevos hardware principales, consultar los datos existentes, eliminar asignaciones y exportar la información a un archivo Excel. La Tabla 3.3 presenta una descripción detallada de cada función incluida en el módulo.

#### 3.2.1.4. permisoscontroller

Este módulo se encarga de gestionar las operaciones relacionadas con los permisos de los usuarios dentro del sistema. Las funciones proporcionadas permiten leer, registrar, actualizar, eliminar y exportar permisos asignados a los usuarios. La Tabla 3.4 ofrece una descripción detallada de cada una de las funciones que componen este módulo.

#### 3.2.1.5. pficohistorialcontroller

Este módulo gestiona las operaciones relacionadas con el historial de periféricos dentro del sistema. Incluye funciones para registrar eventos en el historial, recuperar registros, y exportar datos a Excel. La Tabla 3.5 proporciona una descripción detallada de cada una de las funciones que componen este módulo.

Tabla 3.2: Funciones principales modulo invpficocontroller.

<b>Funciones</b>	<b>Descripción</b>
<b>AsignaPeriferico</b>	Asigna un periférico a un usuario, creando una nueva entrada en la base de datos a partir de los datos proporcionados en el cuerpo de la solicitud.
<b>LeerPerifericosAsig</b>	Recupera todos los periféricos asignados de la base de datos, los convierte a formato JSON y los devuelve.
<b>RegistrarUsuario</b>	Registra un nuevo usuario en la base de datos después de verificar si ya existe, y cifra su contraseña antes de almacenarla.
<b>LeerSerialPerifericoAsignado</b>	Recupera un periférico específico de la base de datos basado en su número de serie, lo convierte a formato JSON y lo devuelve.
<b>EliminarPerifericoAsignado</b>	Elimina un periférico asignado de la base de datos usando el número de serie como identificador.
<b>LeerIdentificacionFunc</b>	Recupera todos los periféricos asignados a un funcionario específico según su identificación, los convierte a formato JSON y los devuelve.
<b>ExportarPerifericoAsignadosAExcel</b>	Exporta todos los periféricos asignados de la base de datos a un archivo Excel, incluyendo encabezados y datos en columnas como identificación, marca y modelo.

Tabla 3.3: Funciones principales modulo invppalcontroller.

<b>Funciones</b>	<b>Descripción</b>
<b>AsignarHardwarePpal</b>	Asigna un hardware principal a un usuario, creando una nueva entrada en la base de datos a partir de los datos proporcionados en el cuerpo de la solicitud.
<b>LeerHardPpalAsig</b>	Recupera todos los hardware principales asignados de la base de datos, los convierte a formato JSON y los devuelve.
<b>LeerSerialHardPpalAsignado</b>	Recupera un hardware principal específico de la base de datos basado en su número de serie, lo convierte a formato JSON y lo devuelve.
<b>EliminarHardwarePpalAsignado</b>	Elimina un hardware principal asignado de la base de datos usando el número de serie como identificador.
<b>LeerIdentificacionFunc</b>	Recupera todos los hardware principales asignados a un funcionario específico según su identificación, los convierte a formato JSON y los devuelve.
<b>ExportarHardPpalAsigAExcel</b>	Exporta todos los hardware principales asignados de la base de datos a un archivo Excel, incluyendo encabezados y datos en columnas como identificación, marca y modelo.

Tabla 3.4: Funciones principales modulo permisoscontroller.

Funciones	Descripción
<b>LeerPermisosDeUsuario</b>	Recupera los permisos de un usuario específico basándose en su correo electrónico, convierte los datos a formato JSON y los devuelve.
<b>RegPermisosDeUsuario</b>	Registra nuevos permisos para un usuario. Verifica si el usuario ya tiene permisos antes de crear una nueva entrada en la base de datos y devuelve el ID del nuevo permiso.
<b>LeerPermisos</b>	Recupera todos los permisos de usuario de la base de datos, los convierte a formato JSON y los devuelve.
<b>ActualizarPermisoDeUsuario</b>	Actualiza los permisos de un usuario específico con los nuevos valores proporcionados en el cuerpo de la solicitud.
<b>construirUpdateQuery</b>	Construye una consulta SQL para actualizar los permisos del usuario, basada en los nuevos valores proporcionados.
<b>EliminarPermisoDeUsuario</b>	Elimina los permisos de un usuario específico basándose en su correo electrónico, siempre y cuando existan permisos para ese usuario en la base de datos.
<b>ExportaPermisosDeUsuarioAExcel</b>	Exporta todos los permisos de usuario a un archivo Excel, incluyendo encabezados y datos en columnas como tipo y correo electrónico.

Tabla 3.5: Funciones principales modulo pficohistorialcontroller.

Funciones	Descripción
<b>RegHistorialPeriferico</b>	Registra un nuevo evento en el historial del periférico con los datos proporcionados en el cuerpo de la solicitud.
<b>LeerPficoHistorial</b>	Recupera todos los registros del historial de periféricos, convierte los datos a formato JSON y los devuelve.
<b>LeerSerialHardPfico</b>	Recupera todos los registros del historial de un periférico específico basándose en su número de serie, convierte los datos a formato JSON y los devuelve.
<b>ExportaHistorialPerifericoAExcel</b>	Exporta el historial de periféricos a un archivo Excel, incluyendo encabezados y datos en columnas como serial, identificación, apellidos, nombres, estado, fecha y observación.

### 3.2.1.6. ppalhistorialcontroller

Este módulo se encarga de gestionar el historial del hardware principal dentro del sistema. Administra funciones relacionadas con la creación de registros históricos, la recuperación de datos y la exportación a Excel. La Tabla 3.6 presenta una descripción detallada de las funciones que componen este módulo.

Tabla 3.6: Funciones principales modulo ppalhistorialcontroller.

Funciones	Descripción
<b>RegHistorialHardPpal</b>	Registra un nuevo evento en el historial del hardware principal con los datos proporcionados en el cuerpo de la solicitud.
<b>LeerHardPpalHistorial</b>	Recupera todos los registros del historial del hardware principal, convierte los datos a formato JSON y los devuelve.
<b>LeerSerialHardPpal</b>	Recupera todos los registros del historial de un hardware principal específico basándose en su número de serie, convierte los datos a formato JSON y los devuelve.
<b>ExportaHistorialHardPpalAExcel</b>	Exporta el historial del hardware principal a un archivo Excel, incluyendo encabezados y datos en columnas como serial, identificación, apellidos, nombres, estado, fecha y observación.

### 3.2.1.7. regfuncontroller

Este módulo se encarga de gestionar las operaciones relacionadas con los funcionarios en el sistema. Incluye funciones para la creación, lectura, actualización, eliminación y exportación de datos a Excel. La Tabla 3.7 detalla las funciones principales de este módulo.

Tabla 3.7: Funciones principales modulo regfuncontroller.

<b>Funciones</b>	<b>Descripción</b>
<b>RegFuncionario</b>	Registra un nuevo funcionario en la base de datos. Verifica si el funcionario ya existe antes de crear un nuevo registro. Retorna el ID del nuevo funcionario y su identificación, o un error.
<b>LeerFuncionarios</b>	Recupera todos los registros de funcionarios, convierte los datos a formato JSON y los devuelve.
<b>LeerIdentificacionFunc</b>	Recupera un funcionario específico basado en su identificación, convierte los datos a formato JSON y los devuelve.
<b>ActualizarUnFuncionario</b>	Actualiza la información de un funcionario existente basado en los nuevos valores proporcionados en el cuerpo de la solicitud. Maneja la eliminación de la imagen de perfil si es necesario. Retorna un mensaje de éxito o un error.
<b>construirUpdateQuery</b>	Construye una consulta SQL de actualización a partir de un mapa de nuevos valores. Utilizada internamente para generar consultas dinámicas de actualización.
<b>EliminarUnFuncionario</b>	Elimina un funcionario existente basado en su identificación. También maneja la eliminación de la imagen de perfil asociada, si existe. Retorna un mensaje de éxito o un error.
<b>ExportarEmpleadosAExcel</b>	Exporta todos los registros de empleados a un archivo Excel, incluyendo encabezados y datos.

### 3.2.1.8. reghardppalcontroller

Este módulo maneja las operaciones relacionadas con el registro, lectura, actualización, eliminación y exportación de datos de hardware principal. La Tabla 3.8 resume las funciones principales de este módulo.

### 3.2.1.9. regpfricocontroller

Este módulo gestiona operaciones relacionadas con el registro, lectura, actualización, eliminación y exportación de datos de periféricos. La Tabla 3.9 describe las funciones que pertenecen a este modulo en cuestión.

### 3.2.1.10. secretariascontroller

Este módulo gestiona operaciones relacionadas con la lectura y exportación de datos de secretarías y subsecretarías. La Tabla 3.10 describe todas las funciones principales.

## 3.2.2. database

En la Figura 3.3 se muestra el directorio que posee el almacenamiento de toda la base de datos de la aplicación.

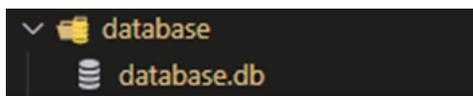
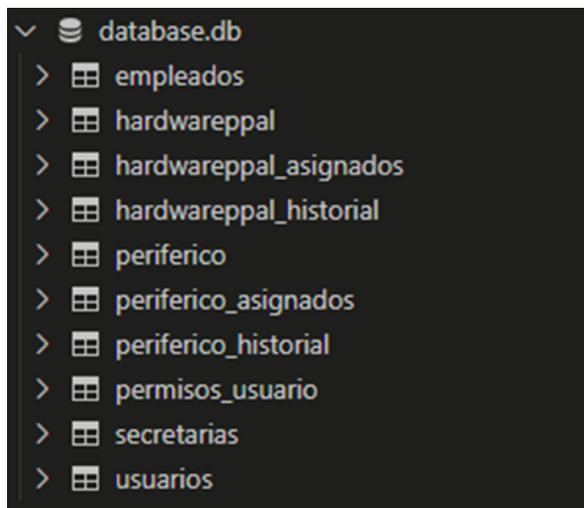


Figura 3.3: Modulo database.

Figura 3.4: Tablas de la base de datos.



En la Figura 3.4 se muestran todas las tablas que se incluyeron dentro de la base de datos, cada una tiene sus dependencias. Mas adelante, en este Capitulo 3, en 3.2.5 models que trata de los modelos de datos usados, se expone a detalle la estructura del sistema de datos, como también el tipo de datos que sigue cada tabla de la base de datos.

Tabla 3.8: Funciones principales modulo reghardppalcontroller.

<b>Funciones</b>	<b>Descripción</b>
<b>RegHardwarePpal</b>	Registra un nuevo hardware principal con los datos proporcionados en el cuerpo de la solicitud. Verifica si el hardware principal ya existe en la base de datos antes de crear un nuevo registro.
<b>LeerTiposDeHardware</b>	Recupera todos los tipos de hardware principal, convierte los datos a formato JSON y los devuelve.
<b>LeerHardwarePpal</b>	Recupera todos los registros de hardware principal, convierte los datos a formato JSON y los devuelve.
<b>LeerSerialHardwarePpal</b>	Recupera un registro de hardware principal específico basado en su serial, convierte los datos a formato JSON y los devuelve.
<b>ActualizarUnHardwarePpal</b>	Actualiza los datos de un hardware principal existente con la información proporcionada en el cuerpo de la solicitud.
<b>construirUpdateQuery</b>	Construye una consulta SQL de actualización a partir de un mapa de nuevos valores para un hardware principal.
<b>EliminarUnHardwarePpal</b>	Elimina un hardware principal existente basado en su serial.
<b>ExportarHardPpalAExcel</b>	Exporta la lista de hardware principal a un archivo Excel, incluyendo encabezados y datos de cada hardware en columnas.

Tabla 3.9: Funciones principales modulo regpficocontroller.

<b>Funciones</b>	<b>Descripción</b>
<b>RegPeriferico</b>	Registra un nuevo periférico con los datos proporcionados en el cuerpo de la solicitud.
<b>LeerTiposDeHardware</b>	Recupera todos los tipos de hardware periférico, convierte los datos a formato JSON y los devuelve.
<b>LeerPerifericos</b>	Recupera todos los registros de periféricos, convierte los datos a formato JSON y los devuelve.
<b>LeerSerialHardwarePfico</b>	Recupera un registro de periférico específico basado en su serial, convierte los datos a formato JSON y los devuelve.
<b>ActualizarUnHardwarePfico</b>	Actualiza los datos de un periférico existente con la información proporcionada en el cuerpo de la solicitud.
<b>construirUpdateQuery</b>	Construye una consulta SQL de actualización a partir de un mapa de nuevos valores para un periférico.
<b>EliminarUnHardwarePfico</b>	Elimina un periférico existente basado en su serial.
<b>ExportarPerifericosAExcel</b>	Exporta la lista de periféricos a un archivo Excel, incluyendo encabezados y datos de cada periférico en columnas.

Tabla 3.10: Funciones principales modulo secretariascontroller.

Funciones	Descripción
<b>LeerSecretarias</b>	Recupera todos los registros de secretarías, convierte los datos a formato JSON y los devuelve.
<b>LeerSubsecretariasFunc</b>	Recupera todos los registros de subsecretarías para una secretaría específica, convierte los datos a formato JSON y los devuelve.
<b>ExportaSecretariasAExcel</b>	Exporta la lista de secretarías a un archivo Excel, incluyendo encabezados y datos de cada secretaría y subsecretaría en columnas.

### 3.2.3. handlerMe

El directorio handlerMe alberga los módulos encargados de interactuar directamente con los controladores del sistema. Estos módulos tienen la responsabilidad de gestionar la comunicación entre el backend y la interfaz de usuario, proporcionando mensajes claros y legibles sobre el estado de las solicitudes realizadas.

En handlerMe, se centralizan las funciones que:

- **Llaman a los Controladores:** Dirigen las solicitudes del usuario a los módulos correspondientes en el directorio de controladores.
- **Proporcionan Mensajes al Usuario:** Generan alertas, notificaciones o mensajes de error que informan al usuario sobre el éxito o el fracaso de una solicitud. Esto incluye la presentación de mensajes que indican si una operación se realizó correctamente o si ocurrieron errores durante el proceso.

La Figura 3.5 muestra la estructura del módulo handlerMe, destacando cómo se organizan y relacionan sus componentes para garantizar una interacción fluida y eficiente con los usuarios del sistema.

#### 3.2.3.1. accounthandler

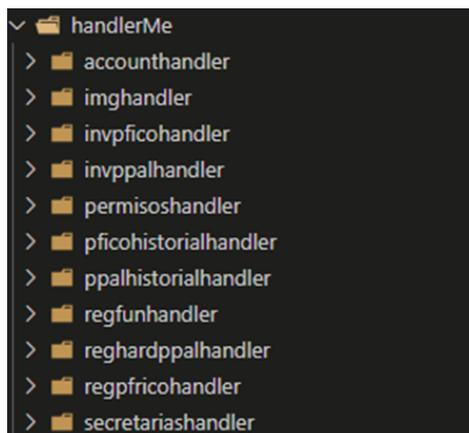
El módulo accounthandler gestiona las operaciones relacionadas con la autenticación de usuarios, el manejo de cuentas y la exportación de datos. Incluye funciones para verificar sesiones, manejar el inicio y cierre de sesión, y gestionar datos de usuarios.

La Tabla 3.11 muestra las funciones principales del módulo accounthandler y sus respectivas descripciones.

Tabla 3.11: Funciones principales módulo accounthandler.

<b>Funciones</b>	<b>Descripción</b>
<b>AuthMiddleware</b>	Verifica la sesión del usuario. Si no hay una sesión activa, redirige al usuario al inicio de sesión.
<b>Index</b>	Sirve la página de inicio (index.html).
<b>Login</b>	Maneja el inicio de sesión del usuario. Verifica las credenciales, guarda la sesión, y redirige según el resultado.
<b>ServirIP</b>	Devuelve la IP local del servidor en formato JSON.
<b>Welcome</b>	Sirve la página de bienvenida (form.html).
<b>GetUserData</b>	Devuelve los datos del usuario obtenidos de la sesión.
<b>Logout</b>	Maneja el cierre de sesión del usuario, eliminando la sesión y redirigiendo al usuario al inicio de sesión.
<b>Registrarse</b>	Sirve la página de registro (register.html).
<b>CuentaUsuario</b>	Sirve la página de cuenta de usuario (usuario.html).
<b>LeerUsuarios</b>	Obtiene todos los usuarios en formato JSON.
<b>LeerUnUsuario</b>	Obtiene los datos de un usuario específico por email y los devuelve en formato JSON.
<b>RegistrarUsuario</b>	Registra un nuevo usuario, manejando errores como usuarios duplicados. Devuelve un mensaje con el ID del nuevo usuario.
<b>ActualizarUsuario</b>	Actualiza los datos de un usuario existente por email. Actualiza la sesión si la actualización es exitosa.
<b>EliminarUsuario</b>	Elimina un usuario por email y devuelve un mensaje de éxito o error dependiendo del resultado.
<b>ExportarUserAExcel</b>	Exporta los datos de todos los usuarios a un archivo Excel y lo devuelve como respuesta para descarga.

Figura 3.5: Estructura del módulo handlerMe



### 3.2.3.2. **imghandler**

El módulo `imghandler` se encarga del manejo de imágenes asociadas a usuarios y funcionarios. Gestiona la carga y eliminación de imágenes, asegurando que se almacenen con nombres únicos y en ubicaciones específicas.

La Tabla 3.12 detalla las funciones principales del módulo `imghandler` y sus respectivas descripciones.

### 3.2.3.3. **invpficohandler**

El módulo `invpficohandler` se encarga de la gestión integral de los periféricos asignados en el sistema. Incluye funcionalidades para la asignación de periféricos, la lectura y eliminación de registros, así como la exportación de datos a archivos Excel. Estas funciones permiten una administración eficiente y completa de los periféricos asignados a funcionarios. La Tabla 3.13 detalla las funciones principales de este módulo.

### 3.2.3.4. **invppalhandler**

El módulo `invppalhandler` gestiona la asignación, consulta y eliminación de hardware principal en el sistema. Además, ofrece herramientas para generar y exportar documentos Excel con detalles sobre las asignaciones. Este módulo permite una administración efectiva del hardware principal asignado a los usuarios y facilita la exportación de datos para su análisis y registro. La Tabla 3.14 detalla las funciones principales de este módulo.

### 3.2.3.5. **permisoshandler**

El módulo `permisoshandler` gestiona los permisos de los usuarios, permitiendo registrar, actualizar, eliminar y exportar permisos, así como renderizar la página de registro de usuarios. Facilita la administración de permisos individuales y ofrece herramientas

Tabla 3.12: Funciones principales módulo imghandler.

Funciones	Descripción
<b>UploadImageUser</b>	Permite a un usuario subir una imagen. La imagen se guarda en un directorio específico (./src/img/users) con un nombre basado en el email del usuario. Devuelve la ruta de la imagen.
<b>DeleteImageUser</b>	Elimina la imagen asociada a un usuario basándose en el prefijo del nombre del archivo (el email del usuario). Busca en el directorio ./src/img/users.
<b>UploadImage</b>	Permite a un funcionario subir una imagen. La imagen se guarda en un directorio específico (./src/img/profiles) con un nombre basado en la identificación del funcionario. Devuelve la ruta de la imagen.
<b>DeleteImage</b>	Elimina la imagen asociada a un funcionario basándose en el prefijo del nombre del archivo (la identificación del funcionario). Busca en el directorio ./src/img/profiles.

para la exportación de datos a formatos Excel, asegurando una gestión eficiente y accesible de los permisos en el sistema. La Tabla 3.15 detalla las funciones principales de este módulo.

### 3.2.3.6. pficohistorialhandler

El módulo pficohistorialhandler gestiona el historial de los periféricos registrados en el sistema. Permite registrar nuevos eventos de historial, consultar todo el historial o uno específico basado en el número de serie de un periférico. También ofrece la opción de exportar el historial a un archivo Excel, facilitando la descarga y análisis de los datos históricos. La Tabla 3.16 detalla las funciones principales de este módulo.

### 3.2.3.7. ppalhistorialhandler

El módulo ppalhistorialhandler se encarga de la gestión del historial de los equipos de hardware principal del sistema. Ofrece funciones para registrar nuevos eventos de historial, consultar tanto el historial completo como el específico de un equipo usando su número de serie, y generar un archivo Excel descargable con el historial total. La Tabla 3.17 detalla las funciones principales de este módulo.

Tabla 3.13: Funciones principales módulo pficohandler.

Funciones	Descripción
<b>AsignaPeriferico</b>	Registra la asignación de un periférico, leyendo los datos del cuerpo de la solicitud y utilizando el controlador para guardar la asignación.
<b>LeerPerifericosAsig</b>	Lee todos los periféricos asignados desde la base de datos y responde con los datos en formato JSON.
<b>LeerSerialPerifericoAsignado</b>	Lee la información del periférico asignado usando el serial proporcionado en la URL. Devuelve la información en formato JSON si se encuentra el periférico.
<b>EliminarPerifericoAsignado</b>	Elimina el periférico asignado basado en el serial proporcionado. Devuelve un mensaje de confirmación o un error si ocurre un problema.
<b>LeerIdentificacionFunc</b>	Lee los periféricos asignados a un funcionario usando la identificación proporcionada en la URL. Devuelve la información en formato JSON si se encuentran periféricos asignados; en caso contrario, devuelve un error 404.
<b>GenerarAsigHardwarePpal</b>	Genera un documento Excel con la información detallada del periférico asignado y del funcionario asociado. Usa una plantilla de Excel para crear el documento, llena los datos del periférico, del funcionario, y del historial del periférico, y lo devuelve como un archivo descargable.
<b>ExportarPficoAsigAExcel</b>	Exporta todos los periféricos asignados a un archivo Excel.

Tabla 3.14: Funciones principales módulo invppalhandler.

<b>Funciones</b>	<b>Descripción</b>
<b>AsignarHardwarePpal</b>	Registra la asignación de hardware principal a un usuario.
<b>LeerHardPpalAsig</b>	Obtiene la información sobre todos los hardware principales asignados.
<b>LeerSerialHardPpalAsignado</b>	Consulta la asignación de hardware principal basada en su número de serie.
<b>EliminarUnHardwarePpal</b>	Elimina una asignación de hardware principal por número de serie.
<b>LeerIdentificacionFunc</b>	Consulta el hardware principal asignado a un funcionario utilizando su identificación.
<b>GenerarAsigHardwarePpal</b>	Genera y descarga un documento Excel con detalles sobre la asignación de un hardware principal.
<b>ExportarHardPpalAsigAExcel</b>	Exporta la información de hardware principal asignado a un archivo Excel para su descarga.

Tabla 3.15: Funciones principales módulo permisoshandler.

<b>Funciones</b>	<b>Descripción</b>
<b>Registrarse</b>	Maneja la solicitud para renderizar la página de registro de usuarios, cargando el archivo HTML <code>userpermits.html</code> . Responde con un error HTTP 500 si ocurre un problema al cargar el archivo.
<b>LeerPermisoDeUsuario</b>	Recupera los permisos de un usuario usando su correo electrónico. Devuelve los permisos en formato JSON o responde con un error HTTP 500.
<b>LeerPermisos</b>	Obtiene y devuelve una lista de todos los permisos almacenados en la base de datos en formato JSON. Responde con un error HTTP 404 si no se encuentran permisos.
<b>RegPermisosDeUsuario</b>	Registra o actualiza permisos para un usuario. Si el usuario ya existe, responde con error HTTP 409. Si es exitoso, devuelve un mensaje de confirmación con el ID del nuevo registro.
<b>ActualizarPermisoDeUsuario</b>	Actualiza los permisos de un usuario utilizando su correo y los datos proporcionados. Devuelve un mensaje de confirmación si es exitoso o un error HTTP 500 en caso de falla.
<b>EliminarPermisosDeUsuario</b>	Elimina los permisos de un usuario por su correo electrónico. Si no tiene permisos o hay un error, responde con error HTTP 409 o 500. Si la eliminación es exitosa, devuelve un mensaje de confirmación.
<b>ExportPermisoDeUserAExcel</b>	Genera un archivo Excel con los permisos de usuario almacenados y lo envía como una descarga.

Tabla 3.16: Funciones principales módulo pfcohistorialhandler.

Funciones	Descripción
<b>RegHistorialPeriferico</b>	Registra un nuevo historial para un periférico basado en la información recibida en el cuerpo de la solicitud. Devuelve el ID del nuevo registro si es exitoso.
<b>LeerPficoHistorial</b>	Recupera y devuelve todo el historial de los periféricos registrados en la base de datos <code>hardwareppal_historial</code> en formato JSON.
<b>LeerSerialHardPfico</b>	Recupera el historial de asignaciones para un periférico específico, identificado por su número de serie. Devuelve un mensaje si no hay historial registrado.
<b>ExportaHistorialPfricoAExcel</b>	Genera un archivo Excel con el historial de periféricos registrado en la base de datos y lo envía como una descarga para el cliente.

### 3.2.3.8. regfunhandler

El módulo regfunhandler se encarga de la gestión de la información de los funcionarios en el sistema. Permite registrar nuevos funcionarios, consultar la lista completa o buscar un funcionario específico por su identificación, actualizar la información de los funcionarios y eliminar funcionarios si es necesario. Además, ofrece la capacidad de exportar la lista de funcionarios a un archivo Excel descargable. La Tabla 3.18 detalla las funciones principales de este módulo.

### 3.2.3.9. reghardppalhandler

El módulo reghardppalhandler está diseñado para gestionar el ciclo de vida de los registros de hardware principal en el sistema. Sus funciones permiten registrar nuevo hardware, consultar la lista completa o información específica por número de serial, actualizar registros existentes, eliminar hardware según el serial, y exportar los datos a un archivo Excel descargable. Este módulo maneja posibles conflictos, como el registro de hardware duplicado, garantizando operaciones seguras y eficientes sobre los datos.

La Tabla 3.19 detalla las funciones principales de este módulo.

Tabla 3.17: Funciones principales módulo ppalhistorialhandler.

Funciones	Descripción
<b>RegHistorialHardPpal</b>	Registra un nuevo historial de hardware principal, procesando la información recibida en el cuerpo de la solicitud. Devuelve el ID del nuevo registro al finalizar exitosamente.
<b>LeerHardPpalHistorial</b>	Recupera y devuelve todo el historial registrado de hardware principal en la base de datos <code>hardwareppal_historial</code> en formato JSON.
<b>LeerSerialHardPpal</b>	Recupera el historial de asignaciones para un hardware principal específico, identificado por su número de serie. Indica si no hay historial registrado.
<b>ExportHistorialHardPpalAExcel</b>	Genera y envía un archivo Excel con todo el historial de hardware principal registrado en el sistema.

Tabla 3.18: Funciones principales módulo regfunhandler.

<b>Funciones</b>	<b>Descripción</b>
<b>RegFuncionario</b>	Registra un nuevo funcionario en el sistema. Procesa la información del cuerpo de la solicitud y verifica si ya existe un funcionario con la misma identificación. Devuelve el ID del nuevo funcionario.
<b>LeerFuncionarios</b>	Recupera y devuelve la lista completa de funcionarios registrados en la base de datos en formato JSON.
<b>LeerIdentificacionFunc</b>	Recupera la información de un funcionario específico según su número de identificación proporcionado en la solicitud.
<b>ActualizarUnFuncionario</b>	Actualiza la información de un funcionario existente según su identificación. La nueva información se envía en el cuerpo de la solicitud.
<b>EliminarUnFuncionario</b>	Elimina un funcionario específico de la base de datos utilizando su identificación. Si el funcionario no existe, devuelve un mensaje correspondiente.
<b>ExportarEmpleadosAExcel</b>	Genera y envía un archivo Excel con la lista completa de funcionarios registrados en el sistema.

Tabla 3.19: Funciones principales módulo reghardppalhandler.

<b>Funciones</b>	<b>Descripción</b>
<b>RegHardwarePpal</b>	Registra un nuevo hardware principal en el sistema. Si el hardware ya existe (identificado por su serial), se notifica el conflicto. Devuelve el ID del nuevo hardware y su serial.
<b>LeerTiposDeHardware</b>	Recupera y devuelve en formato JSON los diferentes tipos de hardware (hardware principal) registrados en el sistema.
<b>LeerHardwarePpal</b>	Recupera y devuelve la lista de todo el hardware principal registrado en la base de datos en formato JSON.
<b>LeerSerialHardwarePpal</b>	Recupera la información de un hardware principal específico a partir de su número de serial. Si no se encuentra, indica que el hardware no existe.
<b>ActualizarUnHardwarePpal</b>	Actualiza la información de un hardware principal existente utilizando los datos recibidos en la solicitud y el número de serial proporcionado. Devuelve un mensaje confirmando la actualización.
<b>EliminarUnHardwarePpal</b>	Elimina un hardware principal de la base de datos según su número de serial. Si hay algún problema al eliminar, se muestra un mensaje de error.
<b>ExportarHardPpalAExcel</b>	Genera un archivo Excel con la lista completa del hardware principal registrado en el sistema.

### 3.2.3.10. regpfricohandler

El módulo regpfricohandler gestiona el ciclo de vida de los registros de periféricos en el sistema. Permite registrar nuevos periféricos, consultar la lista completa o información específica por número de serial, actualizar información existente, eliminar registros y exportar los datos a un archivo Excel descargable. La lógica de control está diseñada para manejar errores comunes, como fallos en el registro, actualizaciones o eliminación de periféricos, proporcionando mensajes adecuados al usuario o administrador del sistema. La Tabla 3.20 detalla las funciones principales de este módulo.

Tabla 3.20: Funciones principales módulo regpfricohandler.

Funciones	Descripción
<b>RegPeriferico</b>	Registra un nuevo periférico en el sistema. Si el proceso de registro falla, devuelve un error de servidor. Devuelve el ID del nuevo periférico registrado.
<b>LeerTiposDeHardware</b>	Recupera y devuelve en formato JSON los diferentes tipos de hardware (periféricos) registrados en el sistema.
<b>LeerPerifericos</b>	Recupera y devuelve la lista de todos los periféricos registrados en la base de datos en formato JSON.
<b>LeerSerialHardwarePfico</b>	Recupera la información de un periférico específico a partir de su número de serial. Si no se encuentra, se devuelve un mensaje de error.
<b>ActualizarUnHardwarePfico</b>	Actualiza la información de un periférico existente usando los datos enviados en la solicitud y el número de serial proporcionado. Devuelve un mensaje confirmando la actualización.
<b>EliminarUnHardwarePfico</b>	Elimina un periférico de la base de datos según su número de serial. Si hay un problema al eliminar el registro, se muestra un mensaje de error.
<b>ExportarPerifericosAExcel</b>	Genera un archivo Excel con la lista completa de los periféricos registrados en el sistema.

### 3.2.3.11. secretariashandler

El módulo secretariashandler gestiona la información de las secretarías y subsecretarías en el sistema. Permite recuperar listas de secretarías y subsecretarías en formato JSON y exportar la lista de secretarías a un archivo Excel descargable. Las funciones están diseñadas para facilitar el acceso y la gestión de la información sobre las secretarías y sus correspondientes subsecretarías, proporcionando una forma organizada de visualizar y exportar estos datos. La Tabla 3.21 detalla las funciones principales de este módulo.

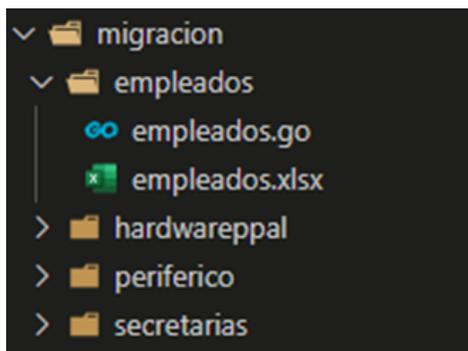
Tabla 3.21: Funciones principales módulo secretariashandler.

Funciones	Descripción
<b>LeerSecretarias</b>	Recupera y devuelve en formato JSON la lista de secretarías disponibles en el sistema. Si no se encuentran, devuelve un error.
<b>LeerSubSecretarias</b>	Recupera y devuelve en formato JSON la lista de subsecretarías de una secretaría específica basada en el parámetro recibido. Si la secretaría no existe, devuelve un mensaje de error.
<b>ExportaSecretariasAExcel</b>	Genera un archivo Excel con la lista completa de secretarías.

### 3.2.4. migracion

Este directorio incluye el código necesario para migrar las bases de datos de los archivos Excel necesarios para empezar a funcionar la aplicación.

Figura 3.6: Estructura del módulo migracion



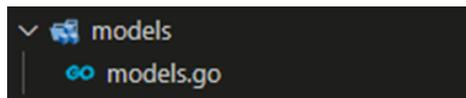
Como se puede ver en la Figura 3.6, cada directorio dispone de su archivo Excel con los datos, y su código respectivo de migración. Datos de los empleados, hardware principales, periféricos y todas las secretarías y subsecretarías de la Gobernación de Sucre. Las demás tablas de la base de datos se empiezan a llenar a medida que se realicen operaciones en la aplicación, como es el registro de permisos, registro de usuarios, las asignaciones, el historial de los diferentes hardware, etc.

### 3.2.5. models

Este directorio contiene la información relacionada a todas las estructuras utilizadas en el software, como también los tipos de datos, y los nombres o etiquetas que reciben para ser llamados o ser accesibles a su contenido tanto en el backend como en el frontend, se usó el mismo etiquetado para evitar confusiones.

Esta la Figura 3.2.5 muestra la estructura general del módulo models, que define las diferentes estructuras de datos utilizadas en el sistema. El módulo incluye definiciones para usuarios, permisos, hardware, periféricos, funcionarios y más, facilitando la integración y manejo de datos en el backend y frontend del software.

Figura 3.7: Estructura del módulo models



La Figura 3.8 detalla la estructura del modelo PermisosUsuario, que define los datos relacionados con los permisos asignados a los usuarios del sistema. Incluye el ID del permiso, tipo de permiso y el correo electrónico del usuario.

Figura 3.8: Estructura y tipos de datos de permisos de usuario

```
type PermisosUsuario struct {
  ID      int    `db:"id" json:"id"`
  Tipo   string `db:"tipo" json:"tipo"`
  Email  string `db:"email" json:"email"`
}
```

En la Figura 3.9 puede ver el modelo Usuario, que representa la información básica de los usuarios en el sistema. Incluye campos para ID, tipo de usuario, nombre, apellido, nombre de usuario, contraseña y URL asociada a la ruta de la imagen de perfil.

La Figura 3.10 Presenta el modelo secretarías, que define las secretarías y subsecretarías en el sistema. Incluye el ID, nombre de la secretaría y nombre de la subsecretaría asociada.

La Figura 3.11 intenta Ilustrar el modelo funcionario, que contiene información detallada sobre los funcionarios del sistema, como identificación, nombres, apellidos, email,

Figura 3.9: Estructura y tipos de datos de un usuario

```
type Usuario struct {
  ID      int    `db:"id" json:"id"`
  Tipo    string `db:"tipo" json:"tipo"`
  Nombre  string `db:"nombre" json:"nombre"`
  Apellido string `db:"apellido" json:"apellido"`
  Username string `db:"username" json:"username"`
  Password string `db:"password" json:"password"`
  Url     string `db:"url" json:"url"`
}
```

Figura 3.10: . Estructura y tipos de datos de secretarias

```
type Secretarias struct {
  ID          int    `db:"id" json:"id"`
  Secretaria  string `db:"secretaria" json:"secretaria"`
  Subsecretaria string `db:"subsecretaria" json:"subsecretaria"`
}
```

teléfono, departamento, ciudad, dirección, tipo y fecha de vinculación, cargo, grado, secretaria, subsecretaria, estado y URL correspondiente a la ruta donde se encuentra su imagen de perfil.

Figura 3.11: Estructura y tipos de datos de un funcionario

```
type Funcionario struct {
  ID          int    `db:"id" json:"id"`
  Identificacion int  `db:"identificacion" json:"identificacion"`
  Apellidos   string `db:"apellidos" json:"apellidos"`
  Nombres     string `db:"nombres" json:"nombres"`
  Email       string `db:"email" json:"email"`
  Telefono    int    `db:"telefono" json:"telefono"`
  Departamento string `db:"departamento" json:"departamento"`
  Ciudad      string `db:"ciudad" json:"ciudad"`
  Direccion   string `db:"direccion" json:"direccion"`
  Tipo_vinculacion string `db:"tipo_vinculacion" json:"tipo_vinculacion"`
  Fecha_vinculacion string `db:"fecha_vinculacion" json:"fecha_vinculacion"`
  Cargo       string `db:"cargo" json:"cargo"`
  Nombre_cargo string `db:"nombre_cargo" json:"nombre_cargo"`
  Grado       string `db:"grado" json:"grado"`
  Secretaria  string `db:"secretaria" json:"secretaria"`
  Subsecretaria string `db:"subsecretaria" json:"subsecretaria"`
  Estado      string `db:"estado" json:"estado"`
  Url         string `db:"url" json:"url"`
}
```

En la figura 3.12 se evidencia el modelo HardwarePpal, que describe los detalles de los equipos de hardware principal. Incluye el número de serie, tipo de hardware, nombre del dispositivo, código gubernamental, marca, procesador, RAM, almacenamiento, y otros componentes relevantes.

La Figura 3.13 detalla el modelo Periferico, que define los periféricos registrados en el sistema. Incluye el número de serie, tipo de hardware, nombre del dispositivo, código gubernamental, marca, modelo, referencia del tóner, número de puertos, IP y extensiones.

La Figura 3.14 presenta el modelo HardwareAsignados, que registra los detalles de

Figura 3.12: Estructura y tipos de datos de un hardware principal

```
type HardwarePpal struct {
    ID          int    `db:"id" json:"id"`
    Serial      string `db:"serial" json:"serial"`
    Tipohardware string `db:"tipohardware" json:"tipohardware"`
    Nombre_dispositivo string `db:"nombre_dispositivo" json:"nombre_dispositivo"`
    Cod_gob     string `db:"cod_gob" json:"cod_gob"`
    Marca       string `db:"marca" json:"marca"`
    Procesador  string `db:"procesador" json:"procesador"`
    RAM         string `db:"RAM" json:"RAM"`
    Almacenamiento string `db:"almacenamiento" json:"almacenamiento"`
    Camara      string `db:"camara" json:"camara"`
    Unidad_CD_DVD string `db:"unidad_CD_DVD" json:"unidad_CD_DVD"`
    Adaptador  string `db:"adaptador" json:"adaptador"`
    Modelo      string `db:"modelo" json:"modelo"`
    SO          string `db:"SO" json:"SO"`
    Key_S0      string `db:"key_S0" json:"key_S0"`
    Version_office string `db:"version_office" json:"version_office"`
    Key_office  string `db:"key_office" json:"key_office"`
    Antivirus   string `db:"antivirus" json:"antivirus"`
    IP          string `db:"IP" json:"IP"`
    MAC         string `db:"MAC" json:"MAC"`
    Estado      string `db:"estado" json:"estado"`
}
```

Figura 3.13: Estructura y tipos de datos de un periférico

```
type Periferico struct {
    ID          int    `db:"id" json:"id"`
    Serial      string `db:"serial" json:"serial"`
    Tipohardware string `db:"tipohardware" json:"tipohardware"`
    Nombre_dispositivo string `db:"nombre_dispositivo" json:"nombre_dispositivo"`
    Cod_gob     string `db:"cod_gob" json:"cod_gob"`
    Marca       string `db:"marca" json:"marca"`
    Modelo      string `db:"modelo" json:"modelo"`
    Referencia_toner string `db:"referencia_toner" json:"referencia_toner"`
    N_puertos   int    `db:"n_puertos" json:"n_puertos"`
    IP          string `db:"IP" json:"IP"`
    N_ext       int    `db:"n_ext" json:"n_ext"`
    Estado      string `db:"estado" json:"estado"`
}
```

los hardware principales asignados a los funcionarios. Incluye el ID, identificación del funcionario, marca, modelo, tipo de hardware, fecha de asignación y número de serie.

Figura 3.14: Estructura y tipos de datos de un hardware principal asignado

```
type HardwareAsignados struct {
  ID          int    `db:"id" json:"id"`
  Identificacion int  `db:"identificacion" json:"identificacion"`
  Marca       string `db:"marca" json:"marca"`
  Modelo      string `db:"modelo" json:"modelo"`
  TipoHardware string `db:"tipohardware" json:"tipohardware"`
  Fecha_asignacion string `db:"fecha_asignacion" json:"fecha_asignacion"`
  Serial      string `db:"serial" json:"serial"`
}
```

La figura 3.15 muestra el modelo HardwareHistorial, que detalla el historial de eventos relacionados con el hardware principal. Incluye un identificador, número de serie, identificación del funcionario, nombres, estado del hardware, fecha del evento y observaciones.

Figura 3.15: Estructura y tipos de datos de un historial de hardware principal

```
type HardwareHistorial struct {
  ID          int    `db:"id" json:"id"`
  Serial      string `db:"serial" json:"serial"`
  Identificacion int  `db:"identificacion" json:"identificacion"`
  Apellidos   string `db:"apellidos" json:"apellidos"`
  Nombres     string `db:"nombres" json:"nombres"`
  Estado      string `db:"estado" json:"estado"`
  Fecha       string `db:"fecha" json:"fecha"`
  Observacion string `db:"observacion" json:"observacion"`
}
```

### 3.2.6. plantillaDeDatos

Este directorio incluye el documento Excel, como se ilustra en la Figura 3.16, que se entrega a cada funcionario luego de una asignación. Aquí se incluye información del funcionario y el hardware que se le esta asignando, para posteriormente ser firmado y llevar un registro físico en la entidad. En la Figura 3.17 se muestra el formato del documento.

Figura 3.16: Plantilla de datos en formato Excel para generar documento



### 3.2.7. repository

El módulo repository es esencial para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre la base de datos utilizando SQL. Este módulo define funciones

Figura 3.17: Documento generado después de una asignación



REPÚBLICA DE COLOMBIA  
GOBERNACIÓN DE SUCRE

FORMATO DE ENTREGA DE PARTES O EQUIPOS

F-0085-02  
VERSION 1.0  
FECHA:

N° Contrato: TELEFONIA MOVISTAR  
Fecha del Contrato: AGOSTO 02 DE 2023

Tipo: N/A	Marca: N/A
Nombre del Funcionario: N/A	Módulo: N/A
Dependencia u Oficina: N/A	

Hardware				
Dispositivo	Marca	Modelo	Cantidad	Serie
Chasis	N/A	N/A	N/A	N/A
Main Board	N/A	N/A	N/A	N/A
Disco Duro 1	N/A	N/A	N/A	N/A
Disco Duro 2	N/A	N/A	N/A	N/A
Fuente de Poder	N/A	N/A	N/A	N/A
Procesador	N/A	N/A	N/A	N/A
RAM: MCO 1	N/A	N/A	N/A	N/A
RAM: MCO 2	N/A	N/A	N/A	N/A
Fanera de Video	N/A	N/A	N/A	N/A
Fanera de Sonido	N/A	N/A	N/A	N/A
Fanera de Red	N/A	N/A	N/A	N/A
Unidad DVD	N/A	N/A	N/A	N/A
Monitor	N/A	N/A	N/A	N/A
Teclado	N/A	N/A	N/A	N/A
Mouse	N/A	N/A	N/A	N/A
Parlante	N/A	N/A	N/A	N/A
Camara	N/A	N/A	N/A	N/A
Otro	N/A	N/A	N/A	N/A

Software			
Nombre de Software	Version	Tipos de Licencia	URL Licencia / Activación
Control Operativo	N/A	PROPRIETARIA	N/A
Control Operativo	N/A	RENTADA	N/A
Activación	N/A	N/A	N/A

Conectividad			
Tipos de Servicio	Nombre	# Puertos	Tipos de Interfaz
Red inalámbrica	WiFi	1	WiFi
Red cableada	Ethernet	1	Ethernet
Modem	Modem	1	Modem
Otro			

**Observaciones**

---

---

---

---

---

---

---

---

Fecha: \_\_\_\_\_

ENTREGA: CARLOS MAURICIO DIAZ ROBERTO \_\_\_\_\_

RECIBE: N/A / C.C. \_\_\_\_\_

Elaboró: Superio Gestión Órgano  
Aprobó: Administrativa

para insertar nuevos registros, recuperar datos específicos o completos, actualizar información existente y eliminar registros de las tablas en la base de datos. La Tabla 3.22 detalla las funciones principales de este módulo.

### 3.2.8. src

En este directorio se almacenan todos los recursos usados en el software de la aplicación, los cuales son accesibles desde el frontend y compartidos de manera estática por el servidor mediante el endpoint `/src`, con la finalidad de servir todas las imágenes que forman parte de la interfaz de usuario. En el subdirectorio `css` se encuentran los estilos de las diferentes páginas HTML, así como los archivos JavaScript. Estos archivos están organizados por secciones, como se ilustra en la Figura 3.18. El contenido de este directorio se expone a detalle en la Sección 3.3.

### 3.2.9. utils

Este directorio posee las funciones que se usan recursivamente en los diferentes módulos, en la Figura 3.19 se muestra que únicamente posee el archivo `go` con las funciones y se describen a continuación. En la Tabla 3.23 se muestran las funciones principales de este modulo.

Tabla 3.22: Funciones principales módulo repository.

Funciones	Descripción
<b>Create</b>	Inserta un nuevo elemento en una tabla usando una consulta SQL.
<b>Read</b>	Recupera un elemento de una tabla usando una consulta SQL.
<b>ReadAll</b>	Recupera todos los elementos que coinciden con una consulta SQL.
<b>List</b>	Recupera elementos de una tabla usando una consulta SQL.
<b>ListAll</b>	Recupera todos los elementos de una tabla usando una consulta SQL.
<b>Delete</b>	Elimina un elemento de una tabla usando una consulta SQL.
<b>Update</b>	Actualiza la información en una fila de una tabla usando una consulta SQL.

Figura 3.18: Directorio src con los archivos CSS, HTML y JavaScript

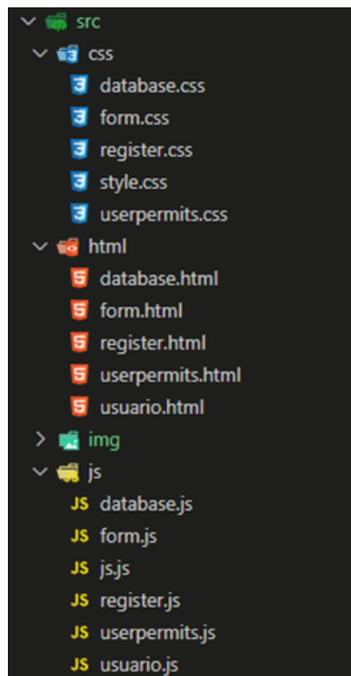


Figura 3.19: Modulo utils



Tabla 3.23: Funciones principales módulo utils.

Funciones	Descripción
<b>GetLocalIP</b>	Obtiene la dirección IP local de la interfaz de red principal. Recorre todas las interfaces de red activas y devuelve la primera dirección IP válida que no sea loopback ni APIPA.
<b>ConectarDB</b>	Conecta a una base de datos usando un URL y un controlador especificado ( <code>sqlite3</code> ). Devuelve un objeto de conexión a la base de datos ( <code>*sqlx.DB</code> ) o un error.
<b>FormatearFecha</b>	Formatea una fecha ( <code>time.Time</code> ) en un string con el formato "MES DÍA DE AÑO", donde el mes está en español. Por ejemplo, "JULIO 03 DE 2024".

### 3.2.10. main.go

El archivo `main.go` es el núcleo de la aplicación web, encargándose de gestionar todos los endpoints necesarios para el funcionamiento del sistema. A continuación, se proporciona una breve descripción de los diferentes grupos de endpoints que se encuentran en el archivo:

1. **Archivos Estáticos:** Maneja la entrega de archivos estáticos desde el directorio "src" (ver Tabla 3.24).
2. **Dirección IP del Servidor:** Permite mostrar la dirección IP del servidor (ver Tabla 3.25).
3. **Inicio de Sesión:** Administra las operaciones relacionadas con el inicio y cierre de sesión de los usuarios, así como la gestión de datos de usuario (ver Tabla 3.26).
4. **Usuarios:** Incluye endpoints para la gestión de usuarios, incluyendo registro, actualización y eliminación (ver Tabla 3.27).
5. **Permisos de Usuario:** Ofrece funcionalidad para gestionar los permisos de los usuarios, así como la carga de imágenes de perfil (ver Tabla 3.28).
6. **Secretarías:** Facilita la consulta de secretarías y subsecretarías (ver Tabla 3.29).

7. **Funcionarios:** Administra la información de los empleados, incluyendo la carga de imágenes de perfil y la actualización de datos (ver Tabla 3.30).
8. **Hardware Principales:** Permite gestionar la información sobre el hardware principal, incluyendo registros y actualizaciones (ver Tabla 3.31).
9. **Asignaciones Hardware Principal:** Maneja la asignación y desasignación de hardware principal, así como el historial de asignaciones (ver Tabla 3.32).
10. **Asignaciones Hardware Periférico:** Similar a los endpoints de hardware principal, pero para periféricos, incluyendo la exportación de datos (ver Tabla 3.33).
11. **Exportar Base de Datos:** Ofrece endpoints para exportar diversas tablas de la base de datos a formato Excel (ver Tabla 3.34).

Cada uno de estos grupos de endpoints se detalla en las tablas, proporcionando una visión completa de las funcionalidades ofrecidas por la aplicación.

Tabla 3.24: Endpoint estáticos.

Archivos Estáticos		
Endpoint	Método	Descripción
/src/	GET	Sirve archivos estáticos desde el directorio "src".

Tabla 3.25: Endpoint para servir IP del servidor.

Dirección IP del Servidor		
Endpoint	Método	Descripción
/servirIP	GET	Muestra la dirección IP del servidor.

### 3.3. FrontEnd

El frontend de la aplicación desarrollada para el Sistema de Gestión de Equipos Informáticos es la interfaz visible y accesible para los usuarios finales, a través de la cual pueden interactuar con el sistema. Esta capa está diseñada para ser intuitiva y fácil de usar, asegurando una experiencia de usuario fluida y eficiente. Se construyó

Tabla 3.26: Endpoints para el inicio de sesión.

Inicio de Sesión		
Endpoint	Método	Descripción
/	GET	Página de inicio de sesión.
/account/login	POST	Inicio de sesión del usuario.
/account/userdata	GET	Obtiene los datos de ingreso del usuario, requiere autenticación.
/account/welcome	GET	Página de bienvenida con formularios, requiere autenticación.
/account/logout	GET	Cierra la sesión del usuario, requiere autenticación.

utilizando tecnologías web estándar como HTML, CSS y JavaScript. La arquitectura del frontend se basa en la separación de responsabilidades, dividiendo los diferentes componentes de la interfaz en archivos específicos que cumplen funciones bien definidas dentro de la aplicación.

El objetivo principal de este apartado es detallar cómo se organizan estos componentes y cuál es su propósito en el contexto general del sistema. Se muestran tanto los archivos HTML, responsables de la estructura básica y del contenido estático de las páginas, como los archivos CSS que proporcionan estilos visuales coherentes con los estándares modernos. Por último, el archivo JavaScript es la pieza clave que otorga dinamismo al frontend, gestionando la interacción del usuario con el backend, a través de llamadas asincrónicas a la API, y manipulando el DOM para reflejar actualizaciones en tiempo real.

- **Archivos HTML:** Cada página principal del sistema cuenta con su propio archivo HTML. Estos archivos son ligeros, conteniendo solo el esqueleto de las páginas, y delegan la mayor parte de la lógica de interacción al JavaScript.
- **Archivos CSS:** El diseño visual de la aplicación está centralizado en varios archivos CSS que controlan la disposición de los elementos, los colores, las fuentes y otros aspectos visuales. Se ha utilizado un enfoque de diseño responsivo, garantizando que la interfaz se adapte a diferentes tamaños de pantalla, desde computadoras de escritorio hasta dispositivos móviles.
- **Archivos JavaScript:** La capa de comportamiento de la aplicación reside en los

Tabla 3.27: Endpoints de manejo de usuarios.

Usuarios		
Endpoint	Método	Descripción
<code>/cuentausuario</code>	<b>GET</b>	Página de perfil de usuario, requiere autenticación.
<code>/register/account</code>	<b>GET</b>	Página de registro de usuarios.
<code>/register/account/user/registered/info</code>	<b>GET</b>	Consulta la tabla de usuarios, requiere autenticación.
<code>/register/account/user</code>	<b>POST</b>	Registra un nuevo usuario.
<code>/register/account/user/email</code>	<b>GET</b>	Consulta la información de un usuario específico por email.
<code>/register/account/user/email</code>	<b>PATCH</b>	Actualiza la información de un usuario específico por email.
<code>/register/account/user/email</code>	<b>DELETE</b>	Elimina la cuenta de un usuario específico por email.

Tabla 3.28: Endpoints para los permisos de usuario.

Permisos de Usuario		
Endpoint	Método	Descripción
/register	GET	Página de registro de permisos de usuario, requiere autenticación.
/register/user	GET	Consulta la tabla de permisos de usuario, requiere autenticación.
/register/user	POST	Registra permisos de usuario, requiere autenticación.
/register/user/imgperfil	POST	Guarda la imagen de perfil de usuarios.
/register/user/email	GET	Consulta permisos de usuario de un email.
/register/user/email	PATCH	Actualiza permisos de usuario de un email.
/register/user/email	DELETE	Elimina permisos de usuario de un email.

Tabla 3.29: Endpoints para secretarías.

Secretarías		
Endpoint	Método	Descripción
/secretarias	GET	Consulta la tabla de secretarías.
/secretarias/secretaria	GET	Obtiene las subsecretarías por secretaria.

Tabla 3.30: Endpoints relacionados a funcionarios.

<b>Funcionarios</b>		
<b>Endpoint</b>	<b>Método</b>	<b>Descripción</b>
<code>/regfuncionario</code>	<b>GET</b>	Consulta la tabla de empleados.
<code>/regfuncionario</code>	<b>POST</b>	Registra un nuevo funcionario.
<code>/regfuncionario/imgperfil</code>	<b>POST</b>	Guarda imagen de perfil del funcionario.
<code>/regfuncionario/identificacion</code>	<b>GET</b>	Busca un funcionario por identificación.
<code>/regfuncionario/identificacion</code>	<b>PATCH</b>	Actualiza la información del funcionario.
<code>/regfuncionario/identificacion</code>	<b>DELETE</b>	Elimina un funcionario por identificación.

Tabla 3.31: Endpoints de hardware principales.

<b>Hardware Principales</b>		
<b>Endpoint</b>	<b>Método</b>	<b>Descripción</b>
<code>/reghardwareppal</code>	<b>GET</b>	Consulta la tabla de hardware principal.
<code>/reghardwareppal</code>	<b>POST</b>	Registra un nuevo hardware principal.
<code>/reghardwareppal/tipohardware</code>	<b>GET</b>	Lee la columna de tipos de hardware.
<code>/reghardwareppal/serial</code>	<b>GET</b>	Busca un hardware principal por serial.
<code>/reghardwareppal/serial</code>	<b>PATCH</b>	Actualiza la información de un hardware principal.
<code>/reghardwareppal/serial</code>	<b>DELETE</b>	Elimina un hardware principal por serial.

Tabla 3.32: Endpoints de asignación de hardware principal.

Asignaciones Hardware Principal		
Endpoint	Método	Descripción
/hardwareppal/asignados	GET	Consulta la tabla del hardware asignado.
/hardwareppal/asignados	POST	Asigna un hardware principal.
/hardwareppal/asignados/serial	GET	Lee el serial del hardware asignado.
/hardwareppal/asignados/serial	DELETE	Elimina un hardware principal asignado.
/hardwareppal/asignados/identificacion/identificacion	GET	Lee los hardware asignados a una identificación.
/hardwareppal/asignados/historial/info	GET	Consulta la tabla de historial de asignaciones del hardware.
/hardwareppal/asignados/historial	POST	Registra un historial de asignaciones de hardware principal.
/hardwareppal/asignados/historial/serial	GET	Lee el historial del hardware por serial.
/hardwareppal-excel/serial	GET	Genera un documento de asignación al hardware principal.

Tabla 3.33: Endpoints asignaciones de periféricos.

Asignaciones Hardware Periférico		
Endpoint	Método	Descripción
/periferico/asignados	GET	Consulta la tabla de periféricos asignados.
/periferico/asignados	POST	Asigna un periférico.
/periferico/asignados/serial	GET	Lee el serial de un periférico asignado.
/periferico/asignados/serial	DELETE	Elimina un periférico asignado.
/periferico/asignados/identificacion/identificacion	GET	Lee los periféricos asignados a una identificación.
/periferico/asignados/historial/info	GET	Consulta la tabla de historial de asignaciones de periféricos.
/periferico/asignados/historial	POST	Registra un historial de asignaciones de periféricos.
/periferico/asignados/historial/serial	GET	Lee el historial de un periférico por serial.
/periferico-excel/serial	GET	Genera un documento de asignación de periféricos.

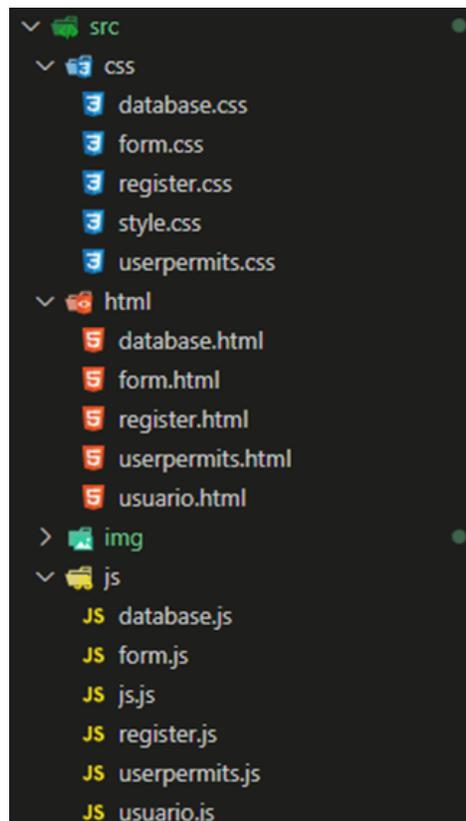
Tabla 3.34: Endpoints de exportación de base de datos.

Exportar Base de Datos		
Endpoint	Método	Descripción
/basededatos/empleados	GET	Exporta la tabla de empleados a Excel.
/basededatos/hardwareppal	GET	Exporta la tabla de hardware principal a Excel.
/basededatos/hardwareppal_asignados	GET	Exporta la tabla de hardware principal asignados a Excel.
/basededatos/hardwareppal_historial	GET	Exporta la tabla de historial de hardware principal a Excel.
/basededatos/periferico	GET	Exporta la tabla de periféricos a Excel.
/basededatos/periferico_asignados	GET	Exporta la tabla de periféricos asignados a Excel.
/basededatos/periferico_historial	GET	Exporta la tabla de historial de periféricos a Excel.
/basededatos/permisos_usuario	GET	Exporta la tabla de permisos de usuario a Excel.
/basededatos/secretarias	GET	Exporta la tabla de secretarías a Excel.
/basededatos/usuarios	GET	Exporta la tabla de usuarios a Excel.

scripts JavaScript. Estos scripts se encargan de manejar eventos del usuario, como clics en botones, envío de formularios, y otras interacciones, así como de realizar peticiones asíncronas al servidor utilizando Axios. La integración con el backend es fundamental para proporcionar datos actualizados, lo que se logra mediante el uso de APIs RESTful que permiten recuperar información de la base de datos y reflejar cambios en la interfaz de forma instantánea.

El frontend sigue una arquitectura modular ilustrado en la Figura 3.20, donde cada página o sección de la aplicación tiene archivos dedicados que cumplen funciones específicas. Esta organización facilita el mantenimiento y permite futuras expansiones de la aplicación. A continuación, se detalla la estructura básica de los principales archivos y carpetas:

Figura 3.20: Estructura general del Frontend



### 3.3.1. database

Los archivos mostrados en la Figura 3.35 hacen parte al funcionamiento que tiene la función de la base de datos del menú en el apartado de Consulta, aquí se encuentran los estilos realizados `database.css` y el HTML respectivo `database.html`, como también el JavaScript `database.js`.

Tabla 3.35: Funciones principales de `database.js`

Funciones	Descripción
<b>ObtenerIPservidor</b>	Obtiene la dirección IP del servidor desde un endpoint (/servirIP) y la asigna a la variable <code>apiBaseUrl</code> para usarla en futuras consultas a la base de datos.
<b>mostrarTabla</b>	Crea y muestra una tabla en la interfaz con el título, encabezados y datos proporcionados. Incluye opciones de búsqueda y selección de columna para filtrar la tabla.
<b>filterTable</b>	Filtra los datos mostrados en la tabla de acuerdo con la entrada de búsqueda y la columna seleccionada.
<b>renderTable</b>	Renderiza o actualiza la tabla con las filas de datos proporcionadas, eliminando las filas existentes antes de agregar las nuevas.
<b>AnimacionAlerta</b>	Aplica un efecto de animación a un mensaje de alerta al mostrarlo.
<b>ValidarFormulario</b>	Valida el formulario. Si faltan campos por llenar, muestra un mensaje de alerta; si está completo, llama a <code>postFunc</code> para enviar el formulario.
<b>ConsultarBaseDeDatos</b>	Controla las operaciones de búsqueda, cancelación y descarga de datos desde diferentes tablas en la base de datos según la selección del usuario.
<b>generarExcel</b>	Genera un archivo Excel con los datos de una tabla específica de la base de datos y lo descarga en el navegador.
<b>procesarPeticon</b>	Agrega eventos de clic a los botones de Buscar, Cancelar y Descargar para ejecutar las operaciones correspondientes según la tabla seleccionada.
<b>peticion</b>	Realiza las operaciones de búsqueda, cancelación o descarga dependiendo de la operación seleccionada por el usuario (BUSCAR, CANCELAR, DESCARGAR).

### 3.3.2. form

Este es el archivo de más peso, donde se encuentran todos los formularios de todo el menú, en este archivo se encuentran a manera de seguir una organización, las funciones con el mismo nombre al cual hacen alusión y que dan el funcionamiento respectivo al botón del menú. La Figura 3.21 ilustra como fue diseñada su arquitectura principal, en la Tabla 3.36 describe las principales funciones.

Figura 3.21: Estructura general del archivo form.js

```
228 /*-----PESTAÑAS-----  
229 > async function RegistrarPermisos() { ...  
240 }  
241 > async function Registrar() { ...  
1158 }  
1159 > async function Asignaciones() { ...  
2026 }  
2027 > async function Consultar() { ...  
2808 }  
2809
```

Tabla 3.36: Funciones principales de form.js.

Funciones	Descripción
<b>RegistrarPermisos</b>	Maneja todo lo referente al registro de permisos de usuario.
<b>Registrar</b>	Se encuentran todas las funciones para registrar funcionarios, registrar hardware principal y periféricos.
<b>Asignaciones</b>	Aquí se maneja todo lo referente a las asignaciones de hardware principal y periféricos.
<b>Consultar</b>	Se comprenden las funciones que permiten hacer las consultas de funcionario, hardware principales y periféricos.

### 3.3.3. register

El proceso de registro de usuarios se realiza en dos fases principales, en las que se recopila y valida la información necesaria antes de enviar los datos al servidor para su almacenamiento. Durante este proceso, el sistema realiza varias verificaciones, tales como la validación del formulario, el formato de los datos y la carga de una foto de perfil. La Tabla 3.37 describe las principales funciones.

Tabla 3.37: Funciones principales de `register.js`.

Funciones	Descripción
<b>ObtenerIPservidor</b>	Obtiene la dirección IP del servidor mediante una solicitud HTTP y la utiliza como base para las solicitudes posteriores.
<b>FormatearEmail</b>	Formatea el correo electrónico ingresado, convirtiendo todo a minúsculas y eliminando los espacios en blanco innecesarios.
<b>FormatearNombre</b>	Formatea los nombres y apellidos, asegurando que cada palabra comience con mayúscula y el resto en minúsculas.
<b>fotoVistaPrevia</b>	Muestra una vista previa de la foto de perfil seleccionada por el usuario. Permite también restablecer la imagen a una predeterminada si es necesario.
<b>siguientePagina</b>	Cambia entre dos páginas del formulario, mostrando una y ocultando la otra.
<b>Peticiones</b>	Maneja las solicitudes GET para verificar permisos de usuario, y redirige la página o muestra mensajes de error según sea necesario.
<b>Pag2Peticiones</b>	Maneja la segunda parte del registro del usuario, incluyendo la carga de datos y la imagen de perfil.
<b>cargarDatos</b>	Recopila los datos ingresados en el formulario, como el nombre, apellido, y contraseña, y los formatea correctamente.
<b>cargarImagen</b>	Carga la imagen de perfil seleccionada por el usuario al servidor, o confirma el uso de la imagen por defecto si no se seleccionó ninguna.
<b>enviarForm</b>	Envía el formulario completo con los datos personales y la ruta de la imagen de perfil al servidor para crear la cuenta de usuario.

### 3.3.4. userpermits

Los permisos de usuario son una parte fundamental de la gestión de roles dentro del sistema. Permite restringir o conceder acceso a diferentes funcionalidades según el tipo de rol de usuario. La Tabla 3.38 describe las principales funciones que se incluyeron.

### 3.3.5. usuario

Se maneja lo respectivo a la página donde el usuario podría hacer sus modificaciones del perfil, como cambiar la imagen de perfil, renovar la contraseña, o modificar sus nombres y apellidos. La Tabla 3.39 muestra con detalle una descripción de sus principales funciones.

## 3.4. Diagramas de Flujo

### 3.4.1. Permisos de Usuario

En el sistema de gestión de permisos de usuario, se permite a los administradores controlar los accesos y privilegios de los usuarios a diversas funcionalidades dentro de la aplicación. Los permisos son esenciales para garantizar que cada usuario tenga acceso únicamente a las funciones y datos necesarios para sus tareas, manteniendo así la seguridad y el orden en el sistema.

El registro de permisos de usuario implica varios pasos clave que se reflejan en el diagrama de flujo para un permiso de usuario Figura 3.22. El proceso general es el siguiente:

1. **Acceso a la Página de Registro de Permisos:** El administrador accede a la página designada para el registro de permisos de usuario.
2. **Consulta de Permisos Existentes:** Se consulta la tabla de permisos existentes para verificar si el permiso que se desea registrar ya está presente en el sistema.
3. **Registro de Permiso:** Si el permiso no existe, el administrador puede registrar un nuevo permiso. Esto incluye definir el tipo de permiso y asignarlo como administrador o usuario.
4. **Actualización y Eliminación de Permisos:** Además del registro, el sistema permite la actualización de permisos existentes y la eliminación de ellos, eliminando también la cuenta.
5. **Confirmación y Guardado:** Una vez registrado, el nuevo permiso se guarda en la base de datos y se confirma su asignación al usuario correspondiente.

El diagrama de flujo de la Figura 3.22 proporciona una representación visual de estos pasos, detallando los procesos involucrados y las decisiones clave en cada etapa.

Tabla 3.38: Funciones principales de `userpermits.js`.

<b>Funciones</b>	<b>Descripción</b>
<b>PermisosDeUsuario</b>	Controla los eventos relacionados con la gestión de permisos (registrar, actualizar, eliminar, regresar) mediante botones asociados a las acciones del formulario.
<b>peticion</b>	Maneja la lógica interna para cada operación (POST, PATCH, DELETE, BACK), coordinando la validación de datos y la interacción con la API del servidor.
<b>cargarDatos</b>	Extrae los valores del formulario y los formatea (como el correo electrónico) para enviarlos en las solicitudes a la API.
<b>registrarPermisos</b>	Envía una petición POST a la API para registrar un nuevo usuario con su correo y tipo de permiso, y muestra los resultados en pantalla.
<b>actualizarPermisos</b>	Envía una petición PATCH para actualizar los permisos de un usuario existente, recuperando sus datos primero y luego actualizándolos.
<b>obtenerUsuario</b>	Obtiene los datos del usuario actual en la base de datos para verificar si es necesario actualizar su tipo de permiso.
<b>actualizarUsuario</b>	Envía una petición PATCH para actualizar el tipo de usuario en la base de datos.
<b>eliminarPermisos</b>	Envía una petición DELETE para eliminar un usuario, y posteriormente llama a otra función para eliminar también su cuenta asociada.
<b>eliminarUsuario</b>	Elimina completamente un usuario de la base de datos tras eliminar sus permisos.
<b>FormatearEmail</b>	Formatea el correo electrónico ingresado para asegurarse de que no contenga espacios en blanco y esté en minúsculas.

Tabla 3.39: Funciones principales de `usuario.js`.

Funciones	Descripción
<b>fotoVistaPrevia</b>	Gestiona la vista previa de la imagen de perfil en el formulario de registro. Permite cargar una nueva imagen y restaurar la imagen por defecto.
<b>FormatearNombre</b>	Formatea una cadena de texto para capitalizar la primera letra de cada palabra.
<b>Pag2Peticiones</b>	Gestiona los eventos asociados a las acciones de la página, como registrar, eliminar o cancelar, además de manejar la carga de datos del usuario.
<b>infoUsuario</b>	Recupera los datos del usuario actual desde el servidor, incluidos nombre, apellido y foto de perfil.
<b>llenarDatos</b>	Rellena los campos del formulario con la información del usuario recuperado mediante su username.
<b>peticion</b>	Controla las operaciones de actualización (PATCH), eliminación (DELETE) o cancelación en el formulario, enviando los datos correspondientes al servidor.
<b>cargarDatos</b>	Recolecta y formatea los datos ingresados en el formulario, incluidos nombre, apellido y contraseña, para prepararlos para ser enviados al servidor.
<b>cargarImagen</b>	Gestiona la carga de la imagen de perfil, enviándola al servidor si se selecciona una imagen nueva, o confirmando si se mantiene la actual.
<b>actualizarInfo</b>	Llama a las funciones de validación y carga de imagen para actualizar la información del usuario en el servidor.
<b>enviarForm</b>	Envía los datos del formulario, incluyendo la ruta de la imagen de perfil, al servidor para actualizar la cuenta del usuario.
<b>enviarFormDelUser</b>	Envía una solicitud al servidor para eliminar la cuenta del usuario actual, usando los datos recolectados.
<b>redireccionar</b>	Redirige a la página de inicio o bienvenida después de una acción de eliminación de cuenta.

Figura 3.22: Diagrama de flujo para un permiso de usuario



Este proceso asegura que los permisos sean gestionados de manera eficiente y que los usuarios dispongan de los accesos adecuados según sus roles y necesidades en el sistema.

### 3.4.2. Registro de Usuarios

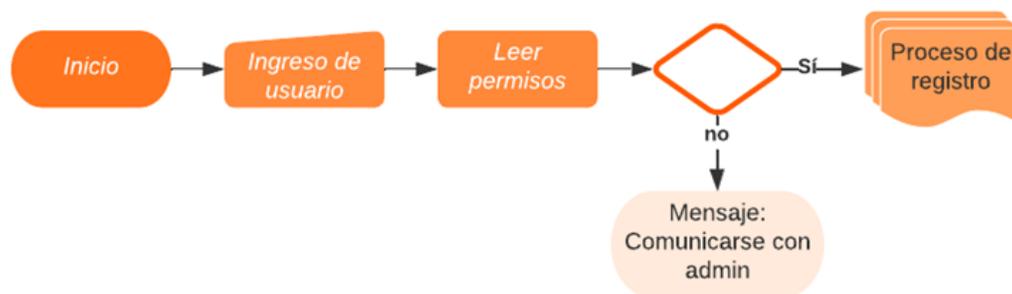
El registro de usuarios en el sistema es un proceso crucial que permite la incorporación de nuevos usuarios y la asignación de credenciales necesarias para acceder a las funcionalidades del sistema. Este proceso está diseñado para ser sencillo y eficiente, garantizando que los nuevos usuarios sean creados con la información adecuada y con los permisos necesarios.

El registro de un nuevo usuario sigue una serie de pasos que se ilustran en el Diagrama de flujo para un registro de usuario. Los pasos del proceso son los siguientes:

1. **Acceso a la Página de Registro:** El administrador o el usuario autorizado accede a la página específica para el registro de nuevos usuarios.
2. **Introducción de Datos del Usuario:** Se ingresan los datos requeridos del nuevo usuario.
3. **Validación de Datos:** El sistema valida que los datos ingresados sean correctos y completos. Esto incluye verificar la unicidad del correo electrónico.
4. **Creación del Usuario:** Si los datos son válidos, el sistema crea una nueva cuenta de usuario en la base de datos.
5. **Confirmación de Registro:** El usuario recibe una confirmación de que su cuenta ha sido creada exitosamente.

El diagrama de flujo de la Figura 3.23 proporciona una representación visual de estos pasos, mostrando claramente el flujo del proceso desde la introducción de datos hasta la confirmación final.

Figura 3.23: Diagrama de flujo para un registro de usuario



Este proceso garantiza que los nuevos usuarios sean registrados de manera eficaz y que reciban las credenciales necesarias para acceder al sistema de acuerdo con los roles y permisos definidos.

### 3.4.3. Registro de Funcionarios, Hardware y Asignaciones

El registro de funcionarios, hardware principales, periféricos y asignaciones en el sistema es un proceso fundamental para la gestión eficiente de recursos y personal. Este proceso asegura que todos los datos relevantes se registren adecuadamente y estén disponibles para su consulta y administración.

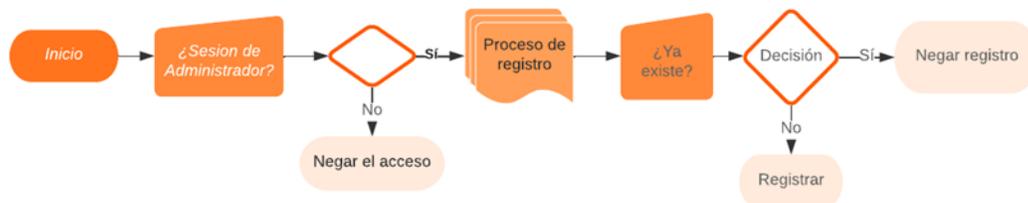
El proceso para registrar funcionarios, hardware principales, periféricos y asignaciones sigue una serie de pasos detallados en el Diagrama de flujo para un registro de funcionario, hardware principal, periférico o asignación. Los pasos del proceso son los siguientes:

1. **Acceso a la Página de Registro:** El administrador o personal autorizado accede a la interfaz del sistema para el registro de nuevos datos.
2. **Introducción de Datos:**
  - **Funcionarios:** Se ingresan los datos personales y profesionales del funcionario, incluyendo información como nombre, identificación y cargo, etc.
  - **Hardware Principales:** Se registran los detalles del hardware principal, como el tipo, marca, modelo y número de serie, etc.
  - **Periféricos:** Se introducen los datos de los periféricos, incluyendo su tipo, modelo y número de serie, etc.
  - **Asignaciones:** Se registran las asignaciones de hardware a funcionarios, especificando qué hardware se ha asignado a cada funcionario y la fecha de asignación o recepción.

3. **Validación de Datos:** El sistema verifica que toda la información ingresada sea correcta y esté completa. Esto incluye validar que no existan duplicados en el registro y que los datos sean coherentes.
4. **Creación de Registros:** Una vez validados los datos, el sistema crea los registros correspondientes en la base de datos para funcionarios, hardware principales, periféricos y asignaciones.
5. **Confirmación de Registro:** El usuario recibe una confirmación de que el registro se ha realizado exitosamente. Esto asegura que la información ha sido almacenada correctamente.
6. **Actualización y Eliminación de Registros:** Además del registro inicial, el sistema permite actualizar la información de funcionarios, hardware y periféricos, así como eliminar registros obsoletos.

El diagrama de flujo de la Figura 3.24 ilustra visualmente estos pasos, mostrando el flujo del proceso desde la introducción de datos hasta la confirmación final de los registros.

Figura 3.24: Diagrama de flujo para un registro de funcionario, hardware principal, periférico o asignación

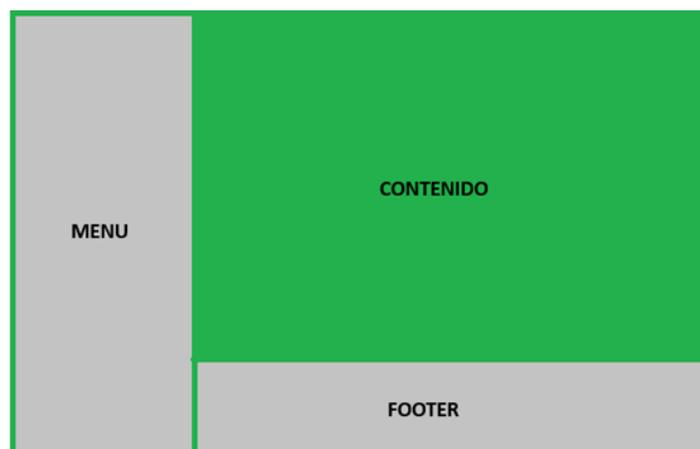


Este proceso garantiza una gestión integral y eficiente de los datos relacionados con funcionarios, hardware, periféricos y asignaciones, facilitando la administración y el control de estos recursos en el sistema.

### 3.5. Mockup de la Interfaz de Usuario

La figura 3.25 presenta el mockup de la interfaz de la página principal de la aplicación, mostrando una vista clara de los componentes principales que estructuran la interfaz de usuario. Este diseño busca ofrecer una experiencia de navegación sencilla y funcional, permitiendo a los usuarios acceder rápidamente a las funcionalidades clave del sistema.

Figura 3.25: Mockup de la interfaz de la página principal



1. **MENU:** Se ubica en la parte lateral derecha de la página, facilitando la navegación entre las distintas secciones de la aplicación. Este menú es intuitivo y está diseñado para que los usuarios puedan moverse entre las diferentes funcionalidades del sistema sin complicaciones, para el diseño responsivo puede ocultarse para dar al usuario una mejor visualización del contenido que desea.
2. **CONTENIDO:** Se encuentra en la zona central, el cual muestra la información que el usuario selecciona desde el menú. Esta área se actualiza dinámicamente según la opción escogida, brindando al usuario acceso directo a la información o las acciones que necesita realizar.
3. **FOOTER:** Se incluye información breve pero importante, como enlaces a la sección ¿Quiénes somos?, datos de contacto, y logos representativos de la institución, reforzando la identidad y profesionalismo en la plataforma.

Este diseño del mockup representa una estructura clara y organizada, orientada a facilitar la navegación y el acceso a la información dentro de la aplicación, mejorando la eficiencia y la experiencia de uso.

# Capítulo 4

## Desarrollo e Implementación

El desarrollo del sistema para la Gobernación de Sucre se realizó de manera estructurada, dividida en fases para facilitar la implementación de las funcionalidades y asegurar la calidad del software. El proceso se enfocó principalmente en tres áreas clave: el registro de usuarios y equipos, la asignación de permisos, y la gestión integral de hardware. La aplicación se diseñó con una arquitectura modular para asegurar la escalabilidad, flexibilidad, y mantenimiento del sistema en el futuro.

Cada módulo del sistema fue desarrollado teniendo en cuenta los requerimientos funcionales y no funcionales establecidos previamente, y se llevó a cabo una integración cuidadosa entre el frontend y el backend, lo que permitió crear un flujo de datos coherente y eficiente. Se puso especial atención en la interacción del sistema con la base de datos para garantizar la seguridad y consistencia de los datos.

### 4.1. Tecnologías Utilizadas

En el desarrollo del sistema para la Gobernación de Sucre, se utilizaron varias tecnologías y librerías para garantizar la funcionalidad y eficiencia del sistema. A continuación, se detallan las principales librerías y herramientas empleadas:

- **Gorilla Mux:** Utilizada para el enrutamiento en el backend del sistema desarrollado en Go. Esta librería permite manejar rutas complejas y parametrizadas, facilitando la gestión de las solicitudes HTTP y el enrutamiento a los controladores correspondientes.
- **Gorilla Sessions:** Librería utilizada para manejar el almacenamiento de sesiones en el backend. Permite gestionar la autenticación de usuarios mediante cookies seguras, asegurando que las sesiones se mantengan activas mientras los usuarios navegan por el sistema. La librería gorilla/sessions proporciona funcionalidades para crear, recuperar y destruir sesiones de usuario.
- **Excelize:** Librería utilizada para la generación y manipulación de archivos Excel

en el sistema. Permite la creación de archivos Excel que contienen datos exportados desde la base de datos, como informes de periféricos y hardware principal.

- **bcrypt:** Utilizada para el hashing de contraseñas en el backend. Esta librería garantiza la seguridad de las contraseñas almacenadas en la base de datos, protegiéndolas mediante un algoritmo de hash robusto.
- **flag:** Esta librería permite la definición de banderas de línea de comandos, que fueron utilizadas para configurar aspectos como el puerto del servidor y la ruta de la base de datos. Mediante `flag.StringVar`, se establecieron valores predeterminados para el puerto del servidor y la ubicación de la base de datos, los cuales pueden ser modificados al ejecutar la aplicación.
- **fmt:** Esta librería se utilizó para formatear cadenas de texto y mensajes de salida en la consola. Resulta útil en la depuración y en la generación de mensajes que se muestran al usuario o en los logs de ejecución del sistema.
- **log:** La librería `log` fue utilizada para registrar información relevante durante la ejecución del sistema, como errores y eventos importantes, como la conexión del servidor o fallos en la base de datos. Esta librería facilita la captura y el seguimiento de errores y permite que el sistema gestione situaciones excepcionales sin interrupciones críticas.
- **net/http:** Esta es la librería fundamental para manejar el servidor HTTP. Fue utilizada para iniciar el servidor, definir rutas y manejar las solicitudes de los usuarios. `net/http` permite gestionar peticiones y respuestas, siendo clave en la implementación de la API que conecta el frontend con el backend, y en la creación de endpoints para interactuar con el sistema.
- **Axios:** Librería JavaScript empleada para realizar solicitudes HTTP desde el frontend. Axios facilita la comunicación con el servidor, permitiendo la obtención y envío de datos en formato JSON.
- **encoding/json:** La librería `encoding/json` es la herramienta nativa de Go para codificar y decodificar datos en formato JSON (JavaScript Object Notation). Dado que el sistema incluye comunicación entre el frontend y el backend usando Axios (que suele enviar datos en formato JSON), esta librería es crucial para convertir estructuras de datos de Go a JSON y viceversa. Es fundamental para el envío y recepción de información entre el servidor y los clientes, permitiendo un intercambio de datos eficiente y estructurado.
- **bytes:** La librería `bytes` en Go proporciona funciones para la manipulación eficiente de secuencias de bytes. Es útil cuando se trabaja con datos binarios o cuando se necesita realizar operaciones complejas sobre secuencias de bytes, como concatenación, búsqueda, o particionado. En el sistema desarrollado, puede ser utilizada

para gestionar datos antes de enviarlos o almacenarlos, especialmente al procesar entradas o salidas de archivos o cadenas de texto.

- **errors:** La librería errors proporciona un mecanismo sencillo y flexible para la creación y manipulación de errores en Go. Es utilizada para generar errores personalizados que describen problemas específicos en la ejecución del programa
- **strings:** La librería strings permite la manipulación y procesamiento de cadenas de texto en Go. Ofrece una amplia gama de funciones para realizar búsquedas, comparaciones, reemplazos y segmentaciones de texto. En el sistema, esta librería es utilizada para manejar datos textuales, como los que ingresan los usuarios al sistema, realizando operaciones como la validación de formularios, limpieza de entradas y formateo de resultados antes de mostrarlos en el frontend.

Estas tecnologías fueron seleccionadas para cubrir las necesidades específicas del proyecto, garantizando una implementación efectiva y segura de las funcionalidades del sistema.

## 4.2. Configuración Inicial

Para facilitar el acceso y la administración inicial del sistema, se ha configurado una cuenta predeterminada con privilegios de administrador. Esta cuenta permite al personal técnico o encargado de la implementación realizar las configuraciones básicas y ajustar los parámetros iniciales del sistema sin complicaciones.

### 4.2.1. Cuenta Administradora

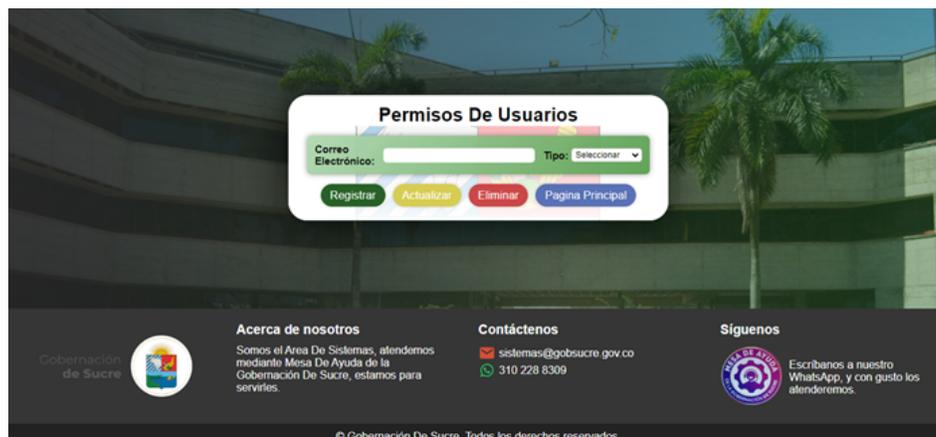
- **Usuario:** admin
- **Contraseña:** admin

Esta configuración predeterminada es temporal y se recomienda cambiar las credenciales inmediatamente después de la primera sesión para garantizar la seguridad del sistema. La cuenta administradora tiene acceso completo a todas las funcionalidades, incluyendo la gestión de usuarios, permisos, hardware, periféricos y asignaciones.

### 4.2.2. Recomendación de Seguridad

Por razones de seguridad, una vez finalizada la configuración inicial, es obligatorio que el administrador cambie las credenciales a un usuario y contraseña más seguros. Además, se recomienda habilitar un sistema de autenticación robusto, como la verificación en dos pasos o el uso de contraseñas complejas, para proteger el acceso no autorizado al sistema.

Figura 4.1: Interfaz del registro de permisos de usuario



Cuando se inicia sesión con una cuenta administradora, en el menú de usuario (Figura 4.2 parte izquierda) disponible en la página principal, se pueden registrar los nombres de usuarios que podrán acceder a la plataforma junto con su rol (usuario o administrador), como se muestra en la Figura 4.1. Solo los usuarios registrados con estos nombres podrán crear una contraseña y completar sus datos para el inicio de sesión.

- **Rol o cuenta de Administrador:** Estas cuentas tienen acceso completo para modificar las bases de datos.
- **Rol o cuenta de Usuario:** Estas cuentas solo pueden visualizar información y no tienen acceso a las opciones de permisos de usuario, registro ni asignaciones.

### 4.3. Página de Inicio de Sesión

En la página de inicio de sesión mostrada en la Figura 4.3, los usuarios podrán ingresar sus credenciales, como el usuario y la contraseña, las cuales serán verificadas por el sistema. También incluye el enlace Registrarse, el cual le permite ir a la Página de Registro de Usuarios y previamente teniendo los permisos dados por un administrador, registrarse una cuenta. Las contraseñas se almacenan cifradas en la base de datos al momento del registro del usuario.

### 4.4. Página de Registro de Usuarios

La página de registro de usuarios mostrada en la Figura 4.4 es accesible desde la página de inicio de sesión, al hacer clic en el enlace “Registrarse”. Aquí, se les pedirá el correo electrónico, si este correo se encuentra registrado con permisos de usuario, los usuarios pueden ingresar sus datos de perfil y, una vez registrados, acceder a la plataforma.

Figura 4.2: Menú de una cuenta administradora y cuenta de usuario

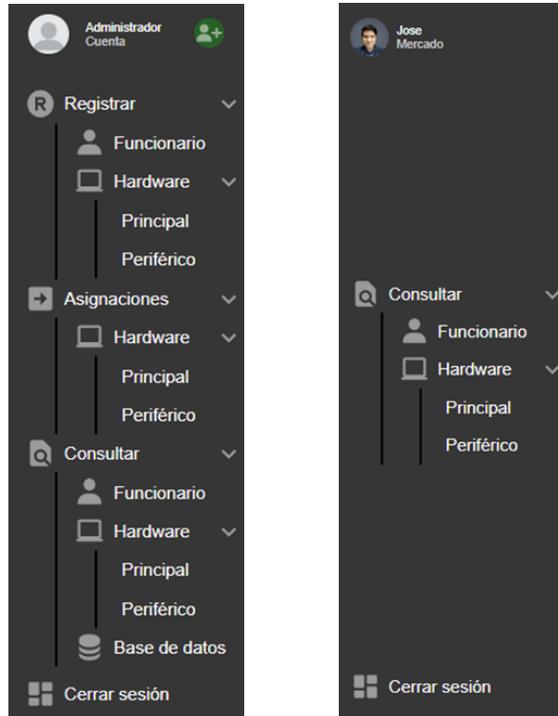


Figura 4.3: Página de inicio de sesión



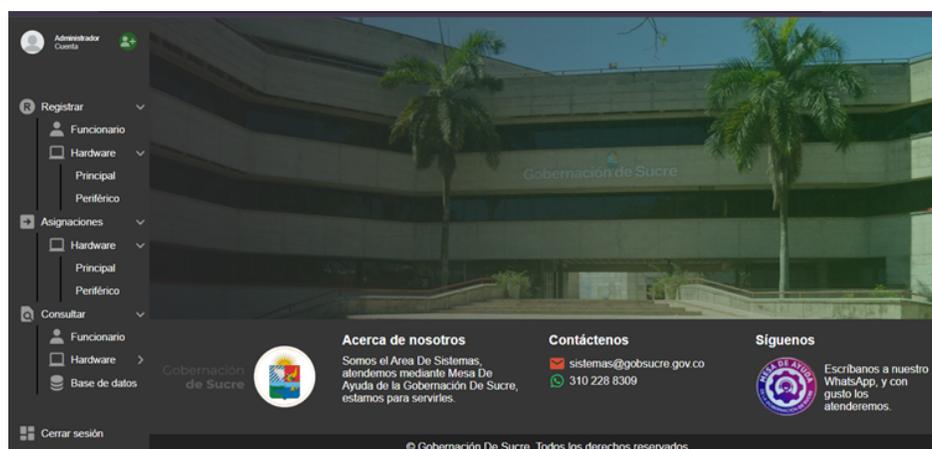
Figura 4.4: Interfaz para el registro de usuarios

The image displays two screenshots of a web application's user registration interface. The left screenshot shows a form titled 'Registro De Usuarios' with the instruction 'Ingresar su correo electrónico con el que desea iniciar sesión.' It features a text input field for 'Correo Electrónico' and two buttons: 'Siguiete' (green) and 'Cancelar' (red). The right screenshot shows a more comprehensive registration form with the title 'Registro De Usuarios' and the instruction 'Ingresar sus datos personales.' It includes fields for 'Apellidos', 'Nombres', 'Contraseña', and 'Confirmar contraseña'. There is also a file upload section with 'Seleccionar archivo' and 'Sin archivos' options. At the bottom, there are three buttons: 'Registrar' (green), 'Volver' (blue), and 'Pagina Principal' (red).

## 4.5. Página Principal

Es esta sección donde se encuentran las funcionalidades de la aplicación web, es decir, la barra de navegación que posee la aplicación, donde se pueden encontrar todos los enlaces, como es el perfil de usuarios, permisos de usuario, los formularios de Registrar, Asignaciones y Consultas. La Figura 4.5 ilustra como es su interfaz.

Figura 4.5: Interfaz de la página principal



### 4.5.1. Perfil

Al dar clic sobre la imagen de perfil, permite modificar los datos del perfil de usuario, cambiar su imagen de perfil, pero no podría cambiar su correo electrónico, ya que esta es la llave de su cuenta de usuario dentro de la plataforma. Obligatoriamente se exige cambiar la contraseña, aunque el usuario puede digitar la misma que ya tiene y confirmarla. La Figura 4.6 se muestra la interfaz que tiene un usuario para realizar cambios en su perfil.

Figura 4.6: Página de actualización de información de cuentas de usuario

## 4.5.2. Registrar

En esta opción se pueden registrar tanto los funcionarios de la institución como los dispositivos de hardware principales y periféricos. Todos los campos están habilitados, por lo que se pueden modificar.

Para utilizar la función de lupa o buscar, basta con ingresar la identificación del funcionario que se desea actualizar, y hacer clic en la imagen de lupa al lado del campo nombrado. Esto traerá la información de la base de datos relacionada a esta identificación, para poder ser modificada si así se requiere.

### 4.5.2.1. Registrar Funcionario

En esta sección del sistema, los usuarios tienen la capacidad de registrar nuevos funcionarios en la base de datos de la institución, lo que permite gestionar y mantener actualizada la información del personal. El formulario de registro de funcionarios está diseñado para ser fácil de usar y contiene campos esenciales como la identificación, nombre completo, cargo, dependencia, y otros datos relevantes que describen el perfil del funcionario.

Cada campo en el formulario es obligatorio y puede ser modificado en caso de que se detecten errores o se necesite actualizar la información. Además, se ha integrado una función de búsqueda que facilita la localización rápida de funcionarios ya registrados. Al ingresar la identificación del funcionario en el campo correspondiente y hacer clic en la lupa, el sistema realiza una consulta en la base de datos, devolviendo la información asociada a ese funcionario. Esto permite que los usuarios puedan modificar o actualizar datos de manera eficiente.

El proceso de registro incluye validaciones en tiempo real para asegurar que no haya duplicados de identificación y que los campos requeridos estén correctamente llenados antes de enviar los datos. Al final del proceso, el sistema confirma el registro exitoso o muestra mensajes de error si ocurre algún problema. La Figura 4.7 representa el

formulario de registro de funcionarios.

Figura 4.7: Formulario para el registro de funcionarios

#### 4.5.2.2. Registrar Hardware Principal

En esta sección del sistema, se facilita el registro de nuevos dispositivos de hardware principal en la base de datos institucional. Los usuarios pueden agregar información detallada sobre los equipos esenciales, como computadoras, servidores, impresoras, y otros dispositivos fundamentales para el funcionamiento de la institución.

El formulario para registrar hardware principal incluye campos clave como el número de serie, tipo de hardware, marca, modelo, y características técnicas relevantes. Cada campo ha sido diseñado para proporcionar toda la información necesaria, permitiendo un control preciso y completo sobre los recursos tecnológicos. Además, se incluyen validaciones automáticas para evitar duplicaciones de números de serie y garantizar que todos los campos requeridos sean completados correctamente antes de finalizar el registro.

Una funcionalidad destacada de esta sección es la posibilidad de gestionar actualizaciones o modificaciones. Si es necesario actualizar la información de un hardware ya registrado, el usuario puede utilizar la herramienta de búsqueda ingresando el número de serie del dispositivo y haciendo clic en la lupa. Esto traerá la información del dispositivo desde la base de datos, permitiendo su edición si es necesario.

El sistema también muestra mensajes de confirmación cuando el registro es exitoso, o notificaciones de error si se produce algún inconveniente durante el proceso. La Figura 4.8 ilustra el formulario para el registro de Hardware principal.

#### 4.5.2.3. Registrar Hardware Periférico

En esta sección, los usuarios pueden registrar nuevos dispositivos periféricos en el sistema, como impresoras, teclados, ratones, entre otros. Esta funcionalidad es funda-

Figura 4.8: Formulario para el registro de hardware principal

mental para llevar un control detallado de los periféricos disponibles en la institución, facilitando su gestión y el acceso rápido a la información de estos dispositivos.

El formulario para el registro de hardware periférico incluye campos esenciales como el número de serie del dispositivo, el tipo de periférico, la marca, el modelo, y observaciones adicionales que el usuario considere relevantes. Todos los campos están habilitados para ser editados, lo que permite una completa personalización de la información almacenada. El sistema también valida automáticamente si el número de serie ya existe, evitando registros duplicados y manteniendo la integridad de los datos.

Además, el sistema proporciona una herramienta de búsqueda que, al ingresar el número de serie del periférico y hacer clic en el icono de lupa, recupera la información existente del dispositivo en la base de datos, lo que permite su actualización o revisión en caso de ser necesario. Una vez realizado el registro el sistema muestra un mensaje que se ha realizado con éxito, o un mensaje de error que detalla lo ocurrido al momento del registro. La Figura 4.9 representa la interfaz de usuario para el registro de un Hardware periférico.

### 4.5.3. Asignaciones

Permite realizar asignaciones de dispositivos de hardware a los funcionarios registrados en la base de datos, los campos están bloqueados, por lo que solo se puede ingresar una identificación, y posteriormente dar clic en la imagen de lupa al lado del campo, esto inmediatamente traerá la información de los campos referente a esta.

Igualmente, para la información del hardware, los campos se han bloqueado para que el administrador ingrese el número de serial del hardware, y posteriormente al dar clic en la imagen de lupa al lado derecho del campo, extraiga la información de la base de datos pertinente a este hardware. Luego se podrá hacer una observación que sería almacenada en el historial del hardware.

Figura 4.9: Formulario para el registro de hardware periférico

The image shows a web application interface for registering peripheral hardware. The main form, titled "Registrar Hardware-Periférico", is centered and contains the following fields: "Serial:" with a search icon, "Código gobernación:", "Tipo de Hardware:" (a dropdown menu), "Marca:", "Nombre dispositivo:", "Modelo:", "Referencia tóner:", "Nro. puertos:", "IP:", "Nro. Extensión:", and "Estado:" (a dropdown menu). Below these fields are four action buttons: "Registrar" (green), "Actualizar" (yellow), "Eliminar" (red), and "Cancelar" (blue). The interface also features a sidebar on the left with navigation options: "Registrar", "Funcionario", "Hardware", "Principal", "Periférico", "Asignaciones", and "Consultar". The footer contains contact information for the "Gobernación de Sucre" and "Mesa de Ayuda", including an email address (sistemas@gobsucre.gov.co) and a phone number (310 228 8309).

#### 4.5.3.1. Asignación de Hardware Principal

El sistema permite la asignación de dispositivos de hardware principal a los funcionarios de la institución. A través de esta funcionalidad, los usuarios pueden registrar qué equipos han sido entregados a cada funcionario, asegurando un control eficiente de los recursos tecnológicos y permitiendo un seguimiento adecuado de los equipos asignados.

El formulario para la asignación de hardware principal incluye campos como el número de identificación del funcionario, el número de serie del hardware, la fecha de asignación, y observaciones relevantes. Estos campos permiten registrar de manera detallada la relación entre los dispositivos y los usuarios a quienes han sido entregados. Además, el sistema verifica automáticamente que el hardware no esté previamente asignado y que el número de identificación del funcionario sea válido, evitando errores y duplicaciones.

La herramienta de búsqueda asociada permite al usuario ingresar el número de identificación del funcionario o el número de serie del hardware y, al hacer clic en la lupa, obtener la información correspondiente para verificar o modificar la asignación. Esto facilita la gestión y actualización de las asignaciones cuando sea necesario.

Una vez realizada la asignación, el sistema muestra un mensaje de confirmación y permite exportar la información de las asignaciones a un archivo Excel para tener un registro adicional fuera del sistema. La Figura 4.10 ilustra el formulario para asignar un Hardware Principal.

#### 4.5.3.2. Asignación de Hardware Periférico

Los usuarios pueden gestionar la asignación de dispositivos periféricos a funcionarios específicos dentro de la institución. Esta funcionalidad es crucial para asegurar un seguimiento adecuado de los recursos, permitiendo a los administradores asignar y controlar qué periféricos están en uso y por quién.

El formulario para la asignación de hardware periférico permite ingresar datos esen-

Figura 4.10: Formulario para la asignación de hardware principal

The screenshot shows a web application interface for hardware assignment. On the left is a dark sidebar with navigation options: 'Registrar', 'Asignaciones' (with sub-options 'Hardware', 'Principal', 'Periférico'), and 'Consultar'. At the bottom of the sidebar is 'Cerrar sesión'. The main content area is titled 'Asignaciones De Hardware-Principal'. It features two green-bordered sections. The first, 'Información Básica Del Funcionario', contains input fields for 'Identificación' (with a search icon), 'Nombres', 'Apellidos', 'Teléfono', and dropdown menus for 'Nombre del cargo', 'Secretaria', 'Sub-Secretaria', and 'Estado'. The second, 'Información Del Hardware', contains input fields for 'Serial' (with a search icon), 'Nombre dispositivo', 'Modelo', 'Tipo de Hardware', 'Marca', and 'Estado'. Below these is an 'Observación...' text area. At the bottom are three buttons: 'Asignar' (green), 'Recibir' (red), and 'Cancelar' (blue).

ciales, como el número de serie del dispositivo que se desea asignar, la identificación del funcionario que lo recibirá y la fecha de asignación. Cada uno de estos campos es obligatorio para garantizar que toda la información necesaria se capture correctamente. La interfaz también incluye un sistema de validación que verifica que el periférico esté disponible para la asignación y que el funcionario existe en la base de datos.

Una característica clave de esta sección es la función de búsqueda. Al ingresar el número de serie del periférico o la identificación del funcionario y hacer clic en el icono de lupa, el sistema recupera automáticamente la información correspondiente. Esto permite al usuario verificar que los datos son correctos antes de proceder con la asignación. Al finalizar el proceso, el sistema genera un mensaje de confirmación que asegura al usuario que la asignación se ha realizado con éxito. Además, se ofrece la opción de imprimir un recibo o generar un documento Excel que contenga un resumen de la asignación realizada, lo que facilita el seguimiento y la gestión de los recursos periféricos. La Figura 4.11 ilustra la interfaz que tiene el usuario para asignar un periférico.

Figura 4.11: Formulario para la asignación de hardware periférico

The screenshot shows a web application interface for hardware assignment. On the left is a dark sidebar with navigation options: 'Registrar', 'Asignaciones' (with sub-options 'Hardware', 'Principal', 'Periférico'), and 'Consultar'. At the bottom of the sidebar is 'Cerrar sesión'. The main content area is titled 'Asignaciones De Hardware-Periférico'. It features two green-bordered sections. The first, 'Información Básica Del Funcionario', contains input fields for 'Identificación' (with a search icon), 'Nombres', 'Apellidos', 'Teléfono', and dropdown menus for 'Nombre del cargo', 'Secretaria', 'Sub-Secretaria', and 'Estado'. The second, 'Información Del Hardware', contains input fields for 'Serial' (with a search icon), 'Nombre dispositivo', 'Modelo', 'Nro. puertos', 'IP', 'Nro. Extensión', 'Tipo de Hardware', 'Marca', and 'Estado'. Below these is an 'Observación...' text area. At the bottom are three buttons: 'Asignar' (green), 'Recibir' (red), and 'Cancelar' (blue).

#### **4.5.4. Consultar**

En esta opción se pueden consultar los datos de los funcionarios registrados, los dispositivos de hardware y el historial de asignaciones, así como realizar el seguimiento del estado y la vida útil de los equipos.

Es importante mencionar que en esta sección de consulta todos los campos están bloqueados y no pueden ser modificados, por lo que no se pueden realizar actualizaciones desde aquí. Al momento de dar clic en Consultar asignaciones se muestra una tabla con la información de las asignaciones que tenga el funcionario con la identificación ingresada, divididas en Hardware Principal o Periféricos.

Al consultar hardware principales o periféricos, al dar clic en Consultar historial se mostrará una tabla con la información respectiva al historial que lleva ese hardware en cuestión durante su trayecto de vida. Observaciones realizadas en cada asignación, como también a las personas que ha sido asignado y en qué fecha se hizo.

##### **4.5.4.1. Consultar funcionario**

En esta sección, los usuarios tienen la capacidad de consultar la información detallada de los funcionarios registrados en el sistema. Esta funcionalidad es esencial para la administración eficiente de los recursos humanos dentro de la institución, permitiendo un acceso rápido y organizado a los datos relevantes de cada funcionario.

Al ingresar la identificación del funcionario y hacer clic en el botón de consulta, el sistema presenta una interfaz que muestra todos los datos asociados a dicho funcionario. Es importante destacar que todos los campos en esta sección están bloqueados, lo que significa que no se permite realizar modificaciones. Esto asegura la integridad de la información y previene cambios accidentales que puedan comprometer los registros.

Una vez consultados los datos del funcionario, el usuario puede acceder a información adicional relacionada con asignaciones de hardware. Al hacer clic en “Consultar asignaciones”, se despliega una tabla que presenta todas las asignaciones de hardware principal y periférico asociadas al funcionario, organizadas de manera clara para facilitar la visualización. La Figura 4.12 ilustra el formulario de consulta de funcionarios.

##### **4.5.4.2. Consultar Hardware principal**

Los usuarios pueden acceder a la información detallada sobre el hardware principal registrado en el sistema. Esta funcionalidad es crucial para la gestión efectiva de los recursos tecnológicos, permitiendo un seguimiento preciso de cada dispositivo y su estado operativo.

Al ingresar el número de serie del hardware principal y hacer clic en el botón de consulta, el sistema despliega un formulario que presenta todos los detalles asociados a dicho hardware. Esto incluye información sobre sus características, estado actual, y cualquier asignación previa, lo que facilita una visión integral del dispositivo.

Es importante señalar que, al igual que en la sección de consulta de funcionarios, todos los campos en este formulario están bloqueados, lo que impide cualquier modifica-

Figura 4.12: Formulario para consultar un funcionario

The screenshot shows a web application interface. On the left is a dark sidebar menu with the following items: 'Administrador Cuenta', 'Registrar', 'Asignaciones', 'Consultar', 'Funcionario', 'Hardware', 'Base de datos', and 'Cerrar sesión'. The main content area features a modal form titled 'Consultar Funcionario'. The form includes the following fields: 'Identificación:' (text input with search icon), 'Nombres:' (text input), 'Apellidos:' (text input), 'Correo Electrónico:' (text input), 'Teléfono:' (text input), 'Departamento:' (dropdown), 'Ciudad:' (dropdown), 'Dirección:' (text input), 'Tipo de vinculación:' (dropdown), 'Fecha de vinculación:' (date input 'dd/mm/aaaa'), 'Cargo:' (dropdown), 'Nombre del cargo:' (dropdown), 'Grado:' (dropdown), 'Secretaria:' (dropdown), 'Sub-Secretaria:' (dropdown), and 'Estado:' (dropdown). At the bottom of the form are two buttons: 'Consultar asignaciones' (green) and 'Cancelar' (blue). At the bottom of the page are three links: 'Acerca de nosotros', 'Contáctenos', and 'Síguenos'.

ción. Esta restricción asegura la precisión de los datos y evita cambios no autorizados, contribuyendo a la integridad del registro.

Además, al consultar el hardware principal, los usuarios tienen la opción de revisar el historial de asignaciones de ese dispositivo. Al hacer clic en “Consultar historial”, se genera una tabla que proporciona información sobre las diferentes asignaciones que ha tenido el hardware a lo largo de su vida útil, incluyendo detalles sobre las observaciones realizadas durante cada asignación y las fechas en las que se llevaron a cabo. La Figura 4.13 ilustra el formulario para consulta de Hardware principal.

Figura 4.13: Formulario para consultar un hardware principal

The screenshot shows a web application interface. On the left is a dark sidebar menu with the following items: 'Administrador Cuenta', 'Registrar', 'Asignaciones', 'Consultar', 'Funcionario', 'Hardware', 'Principal', 'Periférico', 'Base de datos', and 'Cerrar sesión'. The main content area features a modal form titled 'Consultar Hardware-Principal'. The form includes the following fields: 'Serial:' (text input with search icon), 'Código gobernación:' (text input), 'Tipo de Hardware:' (dropdown), 'Marca:' (text input), 'Nombre dispositivo:' (text input), 'Modelo:' (text input), 'Procesador:' (text input), 'RAM:' (text input), 'Almacenamiento:' (text input), 'Cámara integrada:' (dropdown), 'Unidad CD/DVD:' (dropdown), 'Cargador/Adaptador:' (dropdown), 'Sistema operativo:' (text input), 'Licencia SO:' (text input), 'Versión de Office:' (text input), 'Licencia Office:' (text input), 'Antivirus:' (text input), 'IP:' (text input), 'MAC:' (text input), and 'Estado:' (dropdown). At the bottom of the form are two buttons: 'Consultar historial' (green) and 'Cancelar' (blue).

#### 4.5.4.3. Consultar Hardware Periférico

En esta sección, los usuarios tienen la capacidad de consultar la información detallada sobre los dispositivos de hardware periférico registrados en el sistema. Esta funcionalidad es esencial para mantener un control adecuado sobre los recursos periféricos

y su uso dentro de la institución.

Para realizar una consulta, el usuario debe ingresar el número de serie del hardware periférico deseado y hacer clic en el botón correspondiente. Una vez hecho esto, se presenta un formulario que muestra todos los datos relevantes sobre el periférico, incluyendo sus características técnicas, estado actual, y cualquier asignación previa que haya tenido.

Al igual que en las secciones de consulta de funcionarios y hardware principal, todos los campos en este formulario están bloqueados, lo que impide que se realicen modificaciones. Esta medida garantiza la integridad de los datos y evita alteraciones no autorizadas, protegiendo así la información crítica sobre los periféricos.

Asimismo, los usuarios pueden acceder al historial de asignaciones del hardware periférico consultado. Al hacer clic en “Consultar historial”, se despliega una tabla que resume las diferentes asignaciones que ha tenido el dispositivo a lo largo de su vida útil, incluyendo observaciones realizadas en cada asignación y las fechas correspondientes. La Figura 4.14 representa la interfaz de usuario al momento de consultar un hardware periférico.

Figura 4.14: Formulario para consultar un hardware periférico

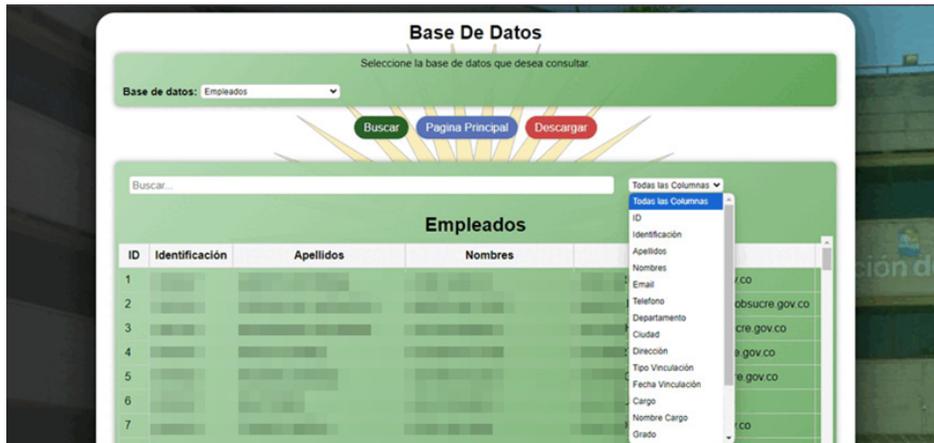


#### 4.5.4.4. Base de Datos

Permite al administrador mirar en forma de tablas toda la información que hay en las bases de datos. Le permite el filtro de información por columnas, como buscar palabras claves en todas las tablas SQL. La Figura 4.15 muestra la interfaz que se tiene al momento de querer consultar la base de datos.

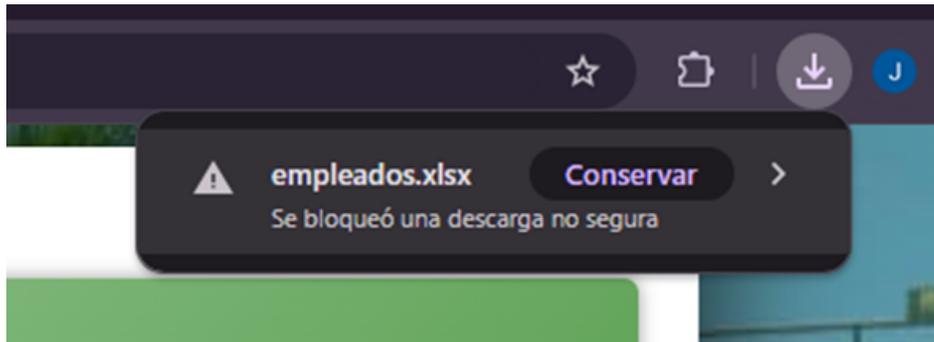
- **Exportaciones de la Base de Datos:** Se da la posibilidad de descargar cualquier tabla de la base de datos seleccionada, sencillamente sería seleccionar la tabla por su nombre, y seguidamente dar clic en el botón Descargar, aquí se generará un archivo Excel que el usuario deberá aceptar la descarga desde el navegador

Figura 4.15: Consulta de base de datos y filtrado de la información



que se esté usando, dando en la opción permitir o conservar, como se ilustra en la Figura 4.16.

Figura 4.16: Descarga de tabla empleados de la base de datos



# Capítulo 5

## Pruebas y Ajustes

Se realizaron pruebas con usuarios del área de sistemas, tanto la persona a cargo, como a los tres funcionarios que serían los usuarios finales en dirigir y utilizar el software, para validar la funcionalidad del sistema. Los errores identificados fueron corregidos y se realizaron ajustes para mejorar la usabilidad, así como sugerencias de posibles funcionalidades adicionales que podría tener la aplicación como futura escalabilidad.

### 5.1. Proceso de Pruebas

#### 5.1.1. Preparación de Casos de Prueba

- Desarrollo de casos de prueba basados en los requisitos funcionales y no funcionales del sistema.
- Creación de un entorno de prueba que simule las condiciones operativas reales.

#### 5.1.2. Ejecución de Pruebas

- Realización de pruebas unitarias para verificar la funcionalidad individual de cada módulo.
- Ejecución de pruebas de integración para asegurar que los módulos interactúan correctamente.
- Pruebas de sistema para validar el comportamiento completo del sistema.
- Pruebas de aceptación del Área de Sistemas para garantizar que el sistema cumple con las expectativas de usuario final.

##### 5.1.2.1. Creación de Permisos de Cuenta de Usuario

Se llevó a cabo una serie de pruebas para verificar que los permisos se registren correctamente y se visualicen de manera adecuada en el sistema. Durante las pruebas,

se registró un nuevo permiso para una cuenta de usuario, lo que permitió evaluar no solo la correcta ejecución del proceso, sino también la usabilidad de la interfaz. La Figura 5.1 muestra el formulario utilizado para el registro de permisos, destacando los campos esenciales que deben completarse para asignar adecuadamente los derechos de acceso.

La creación de permisos es crucial para garantizar que cada usuario tenga acceso a las funcionalidades necesarias según su rol, mejorando así la seguridad y la gestión del sistema. Los resultados obtenidos confirmaron que el proceso de registro de permisos se ejecutó con éxito, sin errores, y que la información se reflejó correctamente en la base de datos.

Este paso en el proceso de pruebas no solo valida la funcionalidad, sino que también asegura que los administradores puedan gestionar efectivamente los permisos de usuario, lo que es fundamental para el funcionamiento adecuado de la aplicación.

Figura 5.1: Registro de permisos de una cuenta de usuario



The image shows a web interface for user permissions. At the top, the title "Permisos De Usuarios" is displayed in bold black text. Below the title, there is a green header bar containing a text input field labeled "Correo Electrónico:" and a dropdown menu labeled "Tipo:" with the option "Seleccionar" and a downward arrow. Underneath the header, a confirmation message reads: "Ha sido registrado permisos al usuario alfonso.mercado@udea.edu.co con ID: 1". At the bottom of the form, there are four buttons: "Registrar" (green), "Actualizar" (yellow), "Eliminar" (red), and "Pagina Principal" (blue).

#### 5.1.2.2. Registro de una Cuenta de Usuario

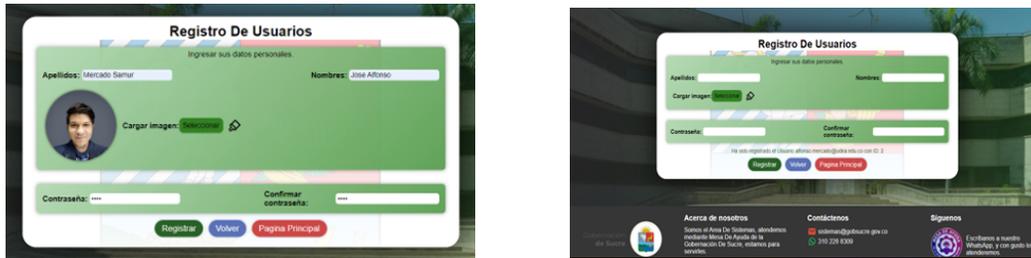
En esta sección se aborda el proceso de registro de una cuenta de usuario dentro del sistema. Durante las pruebas, se evaluó la funcionalidad del formulario de registro para asegurar que todos los datos necesarios fueran capturados correctamente y que se procesaran de acuerdo con los requisitos establecidos.

Se realizaron múltiples intentos de registro para comprobar que el sistema validara correctamente los datos ingresados. Los errores, como la introducción de correos electrónicos no válidos o identificaciones duplicadas, fueron manejados adecuadamente, proporcionando mensajes de retroalimentación claros al usuario.

Los resultados de esta fase de pruebas confirmaron que el proceso de registro de cuentas de usuario se ejecutó sin inconvenientes, permitiendo a los administradores añadir nuevos usuarios de manera eficiente. Esto asegura que el sistema no solo cumpla con los requisitos funcionales, sino que también ofrezca una experiencia de usuario intuitiva y fluida. La Figura 5.2 ilustra el formulario para registrarse como usuario,

donde se pudo ingresar correctamente la imagen de perfil, y la creación del usuario como lo indica el mensaje de confirmación de color verde.

Figura 5.2: Registro de una cuenta de usuario



### 5.1.2.3. Inicio de Sesión con Cuenta de Usuario

En esta sección se analiza el proceso de inicio de sesión para los usuarios del sistema. Se realizó una serie de pruebas para verificar que los usuarios pudieran acceder correctamente a sus cuentas mediante el uso de las credenciales asignadas.

Durante las pruebas, se simuló una variedad de escenarios, incluyendo intentos de inicio de sesión exitosos y fallidos. Se verificó que el sistema respondiera adecuadamente, mostrando mensajes claros en caso de error, como contraseñas incorrectas o cuentas no registradas. Esto contribuyó a mejorar la usabilidad y la experiencia del usuario al proporcionar retroalimentación inmediata.

Los resultados indicaron que el sistema de inicio de sesión es funcional y confiable, permitiendo a los usuarios acceder a sus cuentas de manera segura y eficiente, lo que es fundamental para el manejo adecuado de las funciones dentro de la aplicación. La Figura 46 muestra el inicio de sesión desde una cuenta de usuario creada.

Figura 5.3: Sesión de una cuenta en su rol usuario



#### 5.1.2.4. Consulta de Funcionario

Como se ilustra en la Figura 5.4, así se visualiza la información al realizar pruebas de asignaciones a un funcionario. Al momento de consultar sus asignaciones respectivas a esa identificación relacionada al funcionario, se puede visualizar todos los hardware que tendría asignados, divididos en Hardware principal y Periféricos.

Figura 5.4: Consulta de funcionario con asignaciones de diferentes tipos de hardware



## 5.2. Errores Encontrados

Durante el proceso de pruebas, se identificaron varios errores que fueron documentados y clasificados según su severidad:

### 5.2.1. Errores Críticos

- **Descripción:** Fallos que impedían el funcionamiento de funciones clave del sistema, ya que el servidor no respondía a las peticiones realizadas desde direcciones IP diferente a la del Host.
- **Ejemplo:** Error en los permisos de consultas a la dirección IP del servidor.
- **Corrección:** Se añadieron permisos de consultas GET, POST, PATCH, DELETE desde todas las fuentes de la red.

### 5.2.2. Errores Menores

- **Descripción:** Problemas que no afectaban el funcionamiento del sistema pero que podían mejorar la experiencia del usuario.
- **Ejemplo:** Errores tipográficos en los mensajes de notificación.

- **Corrección:** Revisión y corrección de todos los textos y mensajes del sistema.

## 5.3. Correcciones Realizadas

Todas las correcciones se documentaron cuidadosamente para mantener un registro de los cambios realizados:

### 5.3.1. Corrección de Errores Críticos

- Actualización del código de back-end con la implementación de CORS para asegurar que el servidor acepte peticiones desde cualquier IP de la red.
- Implementación de pruebas unitarias adicionales para prevenir la recurrencia de estos errores.
- Revisión de la arquitectura de la base de datos.

### 5.3.2. Corrección de Errores Menores

- Revisión completa de la interfaz de usuario para asegurar coherencia y corrección de todos los textos.
- Implementación de pruebas de usabilidad para identificar y corregir problemas menores adicionales.

## 5.4. Preparación para el Despliegue

Después de realizar las correcciones necesarias, se ejecutaron pruebas de verificación para confirmar que todos los errores se habían solucionado y que el sistema funcionaba según los requisitos establecidos. Los pasos finales incluyen:

### 5.4.1. Validación Final

- Ejecución de pruebas de regresión para asegurar que las correcciones no introdujeron nuevos errores.
- Confirmación de la aceptación del usuario final.

### 5.4.2. Documentación

- Actualización de la documentación técnica para reflejar los cambios realizados.
- Planificación de guías de usuario y procesos de capacitación.

### 5.4.3. Despliegue

- Configuración del entorno de producción.
- Planificación y ejecución del despliegue del sistema.

Con estas acciones completadas, el sistema de gestión de equipos informáticos estuvo listo para ser desplegado, cumpliendo con todos los requisitos establecidos y proporcionando una solución robusta y eficiente para la Gobernación de Sucre.

# Capítulo 6

## Despliegue

El despliegue del sistema de gestión de equipos informáticos en la Gobernación de Sucre se realizó exitosamente siguiendo un proceso estructurado, asegurando que todas las etapas de configuración y validación se llevaran a cabo de manera eficiente. A continuación, se describen las etapas principales del despliegue:

### 6.1. Configuración del Entorno de Producción

Para garantizar que el sistema operara de manera óptima en el entorno de producción, se llevaron a cabo varias configuraciones clave:

- **Servidor:** Se configuró un servidor dedicado para alojar la aplicación, asegurando que el hardware y software del servidor cumplieran con los requisitos del sistema.
- **Base de Datos:** La base de datos PostgreSQL, que se utilizó durante el desarrollo, fue migrada al entorno de producción. Este proceso incluyó la validación de la integridad de los datos y la implementación de copias de seguridad automáticas para garantizar la persistencia y seguridad de la información.
- **Seguridad:** Se implementaron medidas de seguridad robustas en la fase de despliegue, la persistencia de sesiones de usuario mediante la implementación de cookies seguras. Adicionalmente, se configuraron políticas de acceso controlado para garantizar que solo usuarios de la red local puedan acceder a la base de datos y a los recursos del servidor.

## 6.2. Fase de Pruebas Realizadas por Usuarios Finales

### 6.2.1. Selección de Usuarios Finales

- Se seleccionaron los usuarios representativos de diferentes roles para participar en la fase de pruebas.
- Los usuarios seleccionados incluían administrador y usuarios regular.

### 6.2.2. Ejecución de Pruebas

- **Duración:** La fase de pruebas se realizó durante 2 semanas.
- **Áreas de Prueba:** Los usuarios finales evaluaron todas las funcionalidades clave del sistema, incluyendo.
  - Registro y gestión de usuarios.
  - Asignación y seguimiento de equipos informáticos.
  - Generación de informes y exportación de datos.

### 6.2.3. Metodología de Pruebas

- **Pruebas de Usabilidad:** Evaluación de la facilidad de uso de la interfaz de usuario.
- **Pruebas de Rendimiento:** Verificación del tiempo de respuesta y la estabilidad del sistema bajo cargas de trabajo típicas.
- **Pruebas Funcionales:** Confirmación de que todas las funcionalidades operan según lo esperado.

## 6.3. Informe de Retroalimentación

### 6.3.1. Recopilación de Retroalimentación

- **Reuniones:** Se llevaron a cabo reuniones de seguimiento con los grupos de usuarios para discutir sus experiencias en detalle.
- **Registro de Incidencias:** Se acuerda llevar el registro respecto a cualquier problema o incidencia reportada por los usuarios.

### 6.3.2. Análisis de Retroalimentación

La retroalimentación de los usuarios finales fue analizada para identificar patrones comunes y áreas de mejora. Los resultados se agruparon en las siguientes categorías:

#### ■ Efectividad del Sistema

- Los usuarios reportaron que el sistema cumple con los requisitos funcionales establecidos.
- La mayoría de los usuarios encontraron que el sistema mejoró significativamente la eficiencia en la gestión de equipos informáticos.

#### ■ Eficiencia del Sistema

- Los tiempos de respuesta fueron considerados satisfactorios por la mayoría de los usuarios.
- El sistema demostró ser estable y confiable durante el período de prueba.

#### ■ Sugerencias de Mejora

- Algunos usuarios sugirieron mejoras en la interfaz de usuario para hacerla más intuitiva.
- Se identificaron algunas áreas donde la funcionalidad podría ser ampliada, como reportes más detallados y opciones adicionales de búsqueda y despliegue de información.

### 6.3.3. Correcciones y Mejoras Realizadas

Basándose en la retroalimentación recibida, se realizaron las siguientes acciones:

- **Optimización de la Interfaz de Usuario:** Ajustes en el diseño de la interfaz para mejorar la experiencia del usuario.
- **Corrección de Errores:** Resolución de todos los problemas técnicos reportados durante la fase de pruebas.

# Resultados y Análisis

Con todas las pruebas realizadas y las correcciones implementadas, el sistema de gestión de equipos informáticos ha sido validado como efectivo y eficiente. La retroalimentación positiva de los usuarios finales confirma que el sistema cumple con sus expectativas y mejora la administración de recursos tecnológicos en la Gobernación de Sucre.

Tras la implementación del sistema de gestión de equipos informáticos, se obtuvieron diversos resultados que evidencian mejoras significativas en las operaciones y la integridad en la información de la Gobernación de Sucre.

1. **Eficiencia en la gestión de equipos:** El sistema permitió la gestión centralizada de los activos de TI, facilitando la asignación y control de equipos en tiempo real, en comparación con el sistema manual anterior, ahora se podrían evitar los seriales duplicados, por lo que se tiene mejor integridad de la información de los equipos. Cada equipo está ahora vinculado a un registro único que detalla su historial, desde la adquisición hasta su estado actual, lo que disminuye las incidencias por falta de información actualizada.
2. **Reducción de los tiempos de respuesta ante incidentes:** Gracias a la implementación de flujos de trabajo automatizados, el tiempo promedio de resolución de incidencias disminuye y mejora al actualizar la información, optimizando el flujo de trabajo y mejorando la trazabilidad de las acciones realizadas.
3. **Mejora en el acceso a la información crítica:** El sistema permitió que los funcionarios del área de sistemas accedan a la información actualizada sobre el estado de los equipos, e inventarios. La plataforma centralizada ha reducido la duplicación de datos y ha mejorado la precisión en la toma de decisiones. Los usuarios pueden acceder a reportes personalizados en tiempo real, lo que ha facilitado las auditorías internas.
4. **Seguridad y control de acceso:** La implementación de controles de acceso y permisos basados en roles ha fortalecido la seguridad de la información. Solo los usuarios autorizados pueden realizar modificaciones en los registros de equipos y acceder a datos confidenciales. Además, la integración de sistemas de autenticación de dos factores ha añadido una capa extra de seguridad, minimizando los riesgos de brechas de seguridad.

5. **Estabilidad y rendimiento del sistema:** El sistema mostró un alto nivel de estabilidad, manejando sin problemas las cargas de trabajo esperadas y manteniendo un tiempo de actividad bueno durante el periodo de pruebas. El uso de tecnologías modernas para el desarrollo, como Go para la gestión de sesiones y consultas, aseguró una experiencia de usuario rápida y sin interrupciones, incluso bajo usuarios simultáneos.
6. **Satisfacción de los usuarios:** Se realizó una encuesta posterior a la implementación, en la que se obtuvo satisfacción entre los usuarios del área de sistemas. Los comentarios destacaron la facilidad de uso, la mejora en la disponibilidad de información y la rapidez con la que pueden realizarse tareas que antes se realizaban con más dificultad, como preguntar al encargado para que brinde la información solicitada.
7. **Escalabilidad y proyección futura:** El sistema fue diseñado con una arquitectura modular descrita además en este trabajo que permite su expansión. Pudiendo así incluir funcionalidades adicionales para la aplicación, con lo cual se ampliaría su funcionalidad a la gestión integral de recursos en toda la Gobernación de Sucre y no solo para el Área de Sistemas.

# Conclusiones

El desarrollo e implementación del Sistema de Gestión de Equipos Informáticos para la Gobernación de Sucre alcanzó plenamente los objetivos trazados. La centralización de la información permitió no solo una gestión más eficiente de los recursos tecnológicos, sino también una mayor transparencia en la asignación y mantenimiento de los equipos. Esta mejora en la accesibilidad de los datos fue clave para optimizar los tiempos de respuesta y reducir significativamente los errores humanos asociados con la administración manual de la información.

El sistema ofreció una plataforma segura y confiable para el personal encargado, gracias a la integración de bases de datos relacionales, controles de acceso basados en roles y la adopción de tecnologías web modernas. Estas herramientas no solo aseguraron un entorno robusto y ágil, sino que también permitieron una escalabilidad futura para adaptarse a las crecientes necesidades de la Gobernación. De esta manera, el sistema ha quedado preparado para incorporar nuevas funcionalidades y gestionar un mayor volumen de datos sin comprometer su rendimiento.

Durante las pruebas finales, se evidenció un impacto positivo en los procesos internos de la Gobernación. El sistema no solo mejoró la eficiencia operativa del Área de Sistemas, sino que también facilitó la toma de decisiones basadas en información precisa y en tiempo real. Esta herramienta ha fortalecido la seguridad de la información, minimizando riesgos asociados a la pérdida o acceso no autorizado de datos.

Finalmente, la implementación de este sistema ha demostrado ser una solución sostenible a largo plazo, ofreciendo un marco de trabajo flexible que puede ajustarse a los cambios tecnológicos y organizativos que puedan surgir. Con este proyecto, se ha dado un paso importante hacia la modernización de los procesos de gestión de activos informáticos en la Gobernación de Sucre, beneficiando tanto al personal interno como a las operaciones diarias de la entidad.

# Implementación de Sistema de Gestión de Equipos Informáticos



**UNIVERSIDAD DE ANTIOQUIA**

Facultad de Ingeniería

**PRACTICANTE:** Jose Alfonso Mercado Samur

**ASESORES:** Maria Katherine Plazas Olaya  
Carlos Mauricio Diaz Royerth

**PROGRAMA:** Ingeniería Electrónica

Semestre de la práctica: 2024-1

El proyecto se centró en la implementación de un Sistema de Gestión de Equipos Informáticos. El problema identificado era la falta de un sistema centralizado para la administración de los recursos tecnológicos, lo que causaba demoras, falta de accesibilidad a la información. El sistema desarrollado permitió gestionar en tiempo real los inventarios y la gestión de información crítica como IPs estáticas de dispositivos clave. Se emplearon lenguajes como HTML, CSS, JavaScript y Go, junto con bases de datos relacionales.

Con el crecimiento constante de la infraestructura tecnológica y el número de funcionarios, se volvió indispensable un sistema que permitiera la centralización y automatización de la gestión de los recursos informáticos. El proyecto se enfocó en desarrollar una plataforma web que permitiera a la Gobernación de Sucre contar con un sistema integrado y centralizado.



## Introducción

La tecnología desempeña un papel esencial en el funcionamiento de las organizaciones públicas, siendo los equipos informáticos uno de los activos clave para asegurar la eficiencia operativa. En la Gobernación de Sucre, la administración de los equipos tecnológicos ha sido un desafío creciente debido al aumento del número de dispositivos y usuarios a lo largo del tiempo.



## Metodología

El desarrollo del proyecto se basó en diferentes secciones para alcanzar al resultado final. Esto incluyó un análisis de requerimientos para, posteriormente, realizar el diseño del sistema; seguidamente, el desarrollo e implementación, la ejecución de pruebas y ajustes, y, finalmente, el despliegue de la aplicación.



## Resultados

Tras la implementación del sistema de gestión de equipos informáticos, se obtuvieron diversos resultados que evidencian mejoras significativas en las operaciones y la integridad en la información de la Gobernación de Sucre. Destacan la eficiencia en la gestión de equipos, reducción de los tiempos de respuesta ante incidentes, mejora en el acceso a la información crítica, estabilidad y rendimiento del sistema, seguridad y control de acceso, satisfacción de los usuarios, escalabilidad y proyección futura.



## Objetivos

- ✓ Identificar y definir los requisitos necesarios para la gestión de equipos informáticos en la Gobernación de Sucre.
- ✓ Establecer un diseño detallado y funcional del sistema, que incluya la arquitectura del software, la interfaz de usuario y funcionalidades.
- ✓ Desarrollar las funcionalidades definidas en el diseño, utilizando bases de datos relacionales y una interfaz web utilizando HTML, CSS, JavaScript y Golang.
- ✓ Realizar pruebas del sistema para identificar posibles errores y ejecutar los correctivos que sean necesarios para el cumplimiento de los requerimientos.
- ✓ Desplegar el sistema en el entorno de producción de la Gobernación de Sucre con usuarios finales capacitados.

## Conclusiones

- ✓ La centralización de la información permitió no solo una gestión más eficiente de los recursos tecnológicos, sino también una mayor transparencia en la asignación de los equipos.
- ✓ La accesibilidad de los datos fue clave para optimizar los tiempos de respuesta y reducir significativamente los errores humanos asociados con la administración manual de la información.
- ✓ El sistema ofreció una plataforma segura y confiable para el personal encargado, gracias al control de acceso basados en roles.
- ✓ Se permitió una escalabilidad futura para adaptarse a las crecientes necesidades de la Gobernación de Sucre.

# Referencias

- [1] Open Web Application Security Project (OWASP). *Cryptographic Storage Cheat Sheet*. (2023). URL: <https://owasp.org/cryptographic-storage-cheat-sheet>.
- [2] Atlassian. *¿Qué es la gestión de activos de TI (ITAM)?* (s.f). URL: <https://www.atlassian.com/es/itsm/it-asset-management#before-the-incident>.
- [3] IBM. *Best Practices for Secure Communication in Web Applications*. (s.f). URL: <https://www.ibm.com/security/web-apps>.
- [4] IBM. *Best Practices for Secure Communication in Web Applications*. (s.f). URL: <https://www.ibm.com/security/web-apps>.
- [5] International Association of IT Asset Managers (IAITAM). *Best Practices for IT Asset Management*. (s.f). URL: <https://www.iaitam.org>.
- [6] Mockup. *Mockup: ¿qué es, para qué sirve y ejemplos inspiradores?* (s.f). URL: <https://blog.hubspot.es/website/mockup-que-es>.
- [7] S Publius. *Role-Based Access Control (RBAC) Overview*. (2018). URL: <https://publiusprojects.com/rbac-overview>.
- [8] Quodem. *Plataformas Web y Accesibilidad*. (s.f). URL: <https://quodem.com/blog/que-es-la-accesibilidad-web-como-beneficiar-tu-proyecto/#:~:text=Un%20sitio%2C%20plataforma%20o%20aplicaci%C3%B3n,para%20todas%20las%20personas%20usuarias>.