

Contenido

Sistema Interno de Automatización DISA	2
RESUMEN GENERAL	2
TECNOLOGÍAS USADAS	3
Diagrama Stack Technologies	4
ARQUITECTURA DEL SISTEMA	4
Diagrama de Arquitectura	6
Servicio Complementario: Seguimiento de Órdenes de Garantía y Reportes de Equipos	6
DISEÑO DE INTERFAZ DE USUARIO	7
Principios de Diseño Aplicados:.....	8
Estructura Principal:	9
CASOS DE USO Y ACTORES	13
MODELOS DE DATOS / BASES DE DATOS	15
COSAS POR HACER / PRÓXIMAS TAREAS	22
INSTRUCCIONES BÁSICAS DE DESPLIEGUE LOCAL DEL SISTEMA	24
Requisitos Previos:	24
Clonación del Proyecto:.....	25
Estructura General del Proyecto:	25
Configuración de Base de Datos:.....	25
Despliegue con Docker:.....	26
Acceso al Sistema:	26
Notas de Seguridad Importante:	27
CONCLUSIÓN	27

Sistema Interno de Automatización DISA

RESUMEN GENERAL

Este proyecto corresponde al desarrollo de un Sistema Interno de Automatización de Procesos para DISA (Distribuidora de Equipos y Accesorios para Organizaciones e Instituciones).

El sistema tiene como objetivo principal automatizar procesos operativos críticos de la empresa, tales como:

- Gestión de inventario: Registro, actualización y control de equipos y accesorios.
- Administración de proyectos: Alta, seguimiento y actualización de proyectos relacionados con la distribución de equipos.
- Generación de reportes: Consultas y reportes automáticos sobre inventario, historial de movimientos y datos relacionados.

Adicionalmente, se busca garantizar la seguridad e integridad de los datos sensibles de la organización, eliminando la dependencia de archivos locales (como hojas de cálculo de Excel) para el manejo de información. Esto permite evitar errores manuales, pérdida de datos y problemas de control relacionados con el acceso no autorizado o la falta de actualizaciones automáticas entre los diferentes registros.

Para asegurar una estructura moderna, flexible y preparada para el crecimiento, el sistema se ha diseñado utilizando una arquitectura basada en microservicios. Esta decisión permite:

- Escalabilidad: Facilitar la adición de nuevas funcionalidades o módulos sin afectar los servicios ya existentes.
- Mantenibilidad: Permitir actualizaciones o cambios en componentes específicos sin interrumpir el resto del sistema.

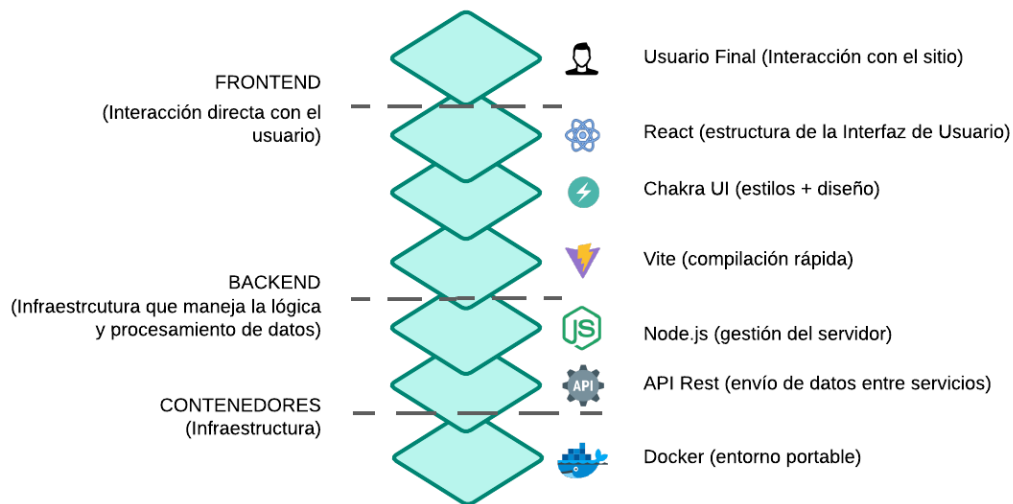
- Integración futura: Facilitar la conexión con sistemas externos o nuevas herramientas que puedan implementarse dentro de la operación de DISA.

El desarrollo está centrado en tecnologías modernas, priorizando una estructura sólida para el backend, integraciones API y bases de datos relacionales, con una documentación orientada a facilitar la continuidad del proyecto por parte de otros desarrolladores o equipos técnicos.

TECNOLOGÍAS USADAS

El sistema está desarrollado utilizando un stack tecnológico moderno, estructurado en tres capas principales: Frontend, Backend e Infraestructura mediante Contenedores. Para el frontend, se implementa React como framework principal para la construcción de interfaces dinámicas, complementado con Chakra UI para el manejo de estilos y componentes visuales prediseñados, y Vite para una compilación rápida y eficiente del proyecto. En el backend, se utiliza Node.js para la gestión del servidor y la lógica de negocio, junto con un esquema de API Rest que permite la comunicación entre servicios y módulos del sistema. Finalmente, toda la infraestructura se encapsula mediante Docker, facilitando un entorno portable y consistente para el despliegue y mantenimiento del sistema, lo cual refuerza la filosofía de microservicios y asegura la escalabilidad futura del proyecto.

Diagrama Stack Technologies



ARQUITECTURA DEL SISTEMA

El Sistema Interno DISA está diseñado siguiendo un enfoque basado en arquitectura de microservicios, buscando asegurar la escalabilidad, mantenibilidad y flexibilidad de cada uno de los módulos que lo componen. La estructura general se divide en varias capas y componentes clave:

- *Cliente (Frontend):*

Representa la interfaz con la que interactúan los usuarios finales, conectándose al sistema a través de un API Gateway.

- *API Gateway:*

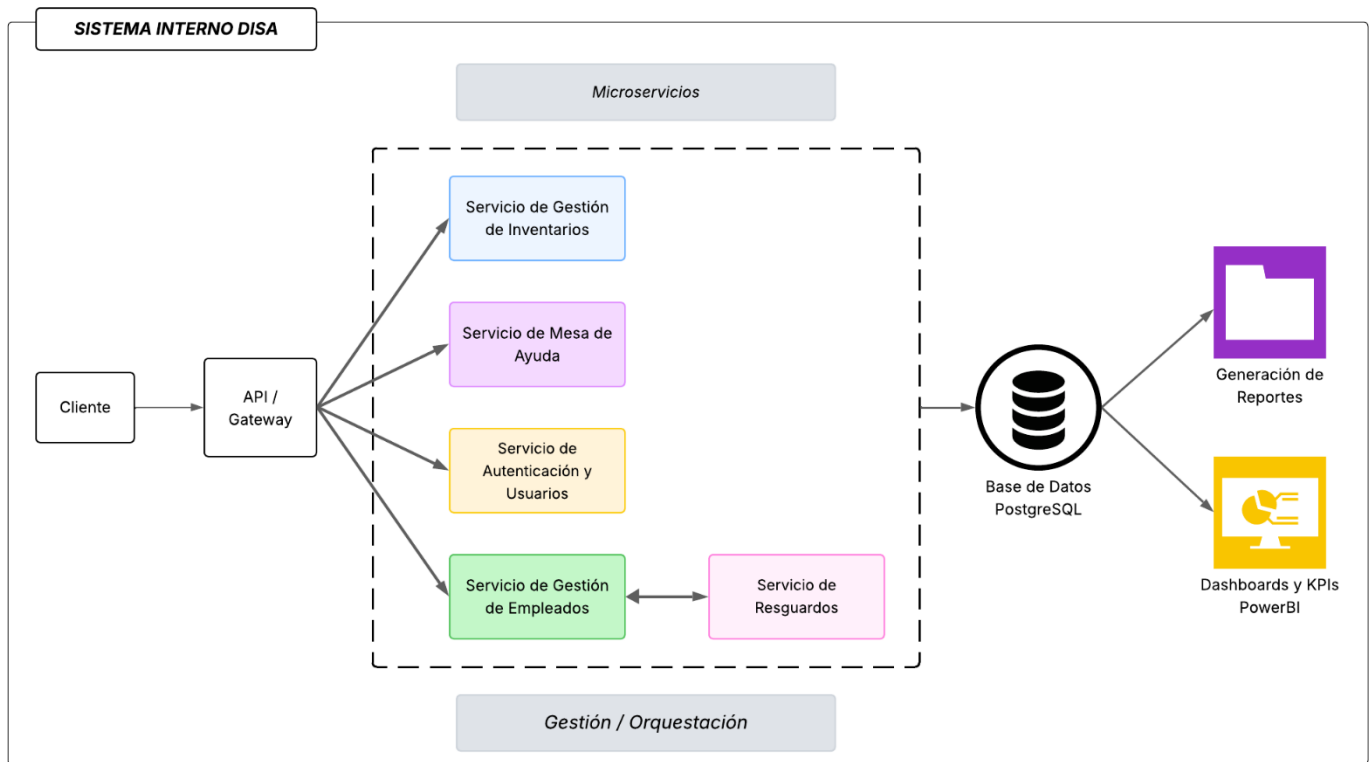
Actúa como punto de entrada unificado para todas las peticiones del cliente, gestionando el enrutamiento hacia los diferentes microservicios, centralizando la autenticación y simplificando la comunicación.

- *Microservicios Principales:*

Cada funcionalidad crítica del sistema se gestiona de forma independiente mediante microservicios específicos, los cuales se comunican entre sí y con la base de datos:

1. **Servicio de Gestión de Inventarios:** Maneja el registro, actualización y control de equipos, accesorios y existencias generales.
 2. **Servicio de Autenticación y Usuarios:** Controla el acceso al sistema, gestionando usuarios, roles y permisos.
 3. **Servicio de Gestión de Empleados:** Administra el registro y actualización de información relacionada con el personal de la empresa.
 4. **Servicio de Resguardos:** Gestiona los préstamos o asignaciones de equipos y accesorios a empleados o proyectos, asegurando el seguimiento adecuado de dichos recursos.
- Base de Datos Central (PostgreSQL): Todos los microservicios comparten una misma base de datos relacional, donde se almacenan los datos críticos de la operación. La información registrada es utilizada tanto para la operativa diaria como para la generación de reportes y análisis.
 - Procesos de Reportes y Análisis: A partir de la información contenida en la base de datos, se generan reportes automáticos y se alimentan dashboards y KPIs mediante herramientas de visualización como Power BI, apoyando la toma de decisiones estratégicas.

Diagrama de Arquitectura



Servicio Complementario: Seguimiento de Órdenes de Garantía y Reportes de Equipos

Adicionalmente, se contempla el desarrollo de un sistema externo especializado para el seguimiento de órdenes de garantía y tickets de reportes relacionados con equipos y accesorios.

Este sistema operará como un microservicio independiente, pero se mantendrá en comunicación directa con el Sistema Interno DISA mediante integraciones API, asegurando el intercambio de información relevante entre ambos entornos. Esto permite un control unificado sin afectar la estructura modular y escalable del sistema principal.

DISEÑO DE INTERFAZ DE USUARIO

El frontend del Sistema Interno DISA fue desarrollado utilizando React.js con Vite para optimizar tiempos de carga, complementado con Chakra UI para asegurar consistencia visual y componentes reutilizables con estilo responsivo.

Principios de Diseño Aplicados:

- **Minimalismo Funcional:** Predominan colores neutros con acentos en azul institucional, priorizando la lectura de datos y evitando distracciones visuales.
- **Jerarquía de Información:** Sección de navegación lateral siempre visible con iconos y texto, y área principal de trabajo enfocada en datos y gráficas.
- **Consistencia y Accesibilidad:** Tipografía clara, botones con estados activos y hover definidos, uso de modales para confirmaciones.

Estructura Principal:

Pantalla de Inicio de Sesión

- Formulario centrado con campos básicos: correo electrónico y contraseña.
- Placeholder con ejemplos y opción de recuperación de contraseña.
- Branding visible con el logo de DISA y pie de página.

DISA

Inicia Sesión

Correo electrónico

Ej. frico8@disa.com

Contraseña

Ej. MGisjb98_

Ingresa

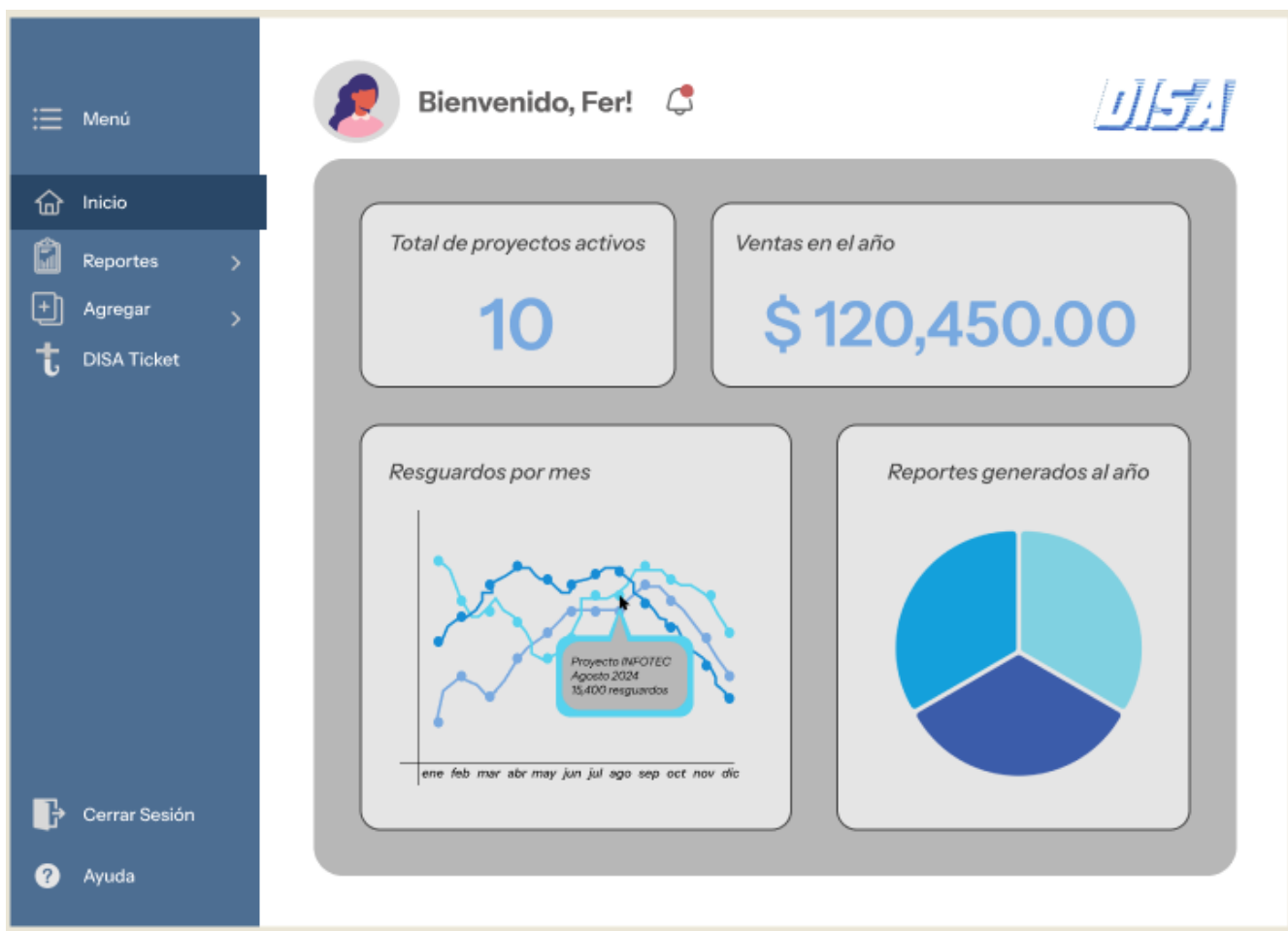
[¿Olvidaste tu contraseña?](#)

Sistema Interno DISA®

Dashboard Principal

- Bienvenida personalizada mostrando el nombre del usuario logueado.
- KPIs visuales:
 - Total de proyectos activos.
 - Ventas anuales (dato editable según necesidades futuras).

- Resguardos por mes con gráficos lineales.
- Reportes generados al año con gráfico de pastel.
- Menú lateral fijo con acceso directo a:
 - Inicio.
 - Reportes (submenú).
 - Agregar registros.
 - Sistema de tickets (DISA Ticket).
 - Cerrar sesión y ayuda.



Vista de Reportes

- Tabla de inventario con filtros avanzados:
 - Proyecto asociado.
 - Fecha de periodo (date picker).
 - Buscador dinámico y paginación.
- Columnas clave como folio, marca, número de serie, perfil, costo, sede y fecha.
- Botones de acción visibles: exportar, imprimir o aplicar filtros.

Menú

Inicio

Reportes

- Inventarios
- Resguardos
- Proyectos
- Historial Movimientos

Agregar

DISA Ticket




Cerrar Sesión

Ayuda

Reportes Inventario

Proyecto: INFOTEC

Fecha de periodo: 01/04/2025 - 30/04/2025



FOLIO	MARCA	N.O. DE SERIE	PERFIL	COSTO TOTAL	SEDE	FECHA
L2607	DELL	6S16BG3	L2 Windows (Portátil intern	\$744.00	SITIO	08/03/2024
L2608	DELL	9106BG3	L2 Windows (Portátil intern	\$642.00	SITIO	21/12/2022
L2609	DELL	G916BG3	L2 Windows (Portátil intern	\$642.00	SITIO	06/03/2023
L2610	DELL	8K26BG3	L2 Windows (Portátil intern	\$744.00	SITIO	05/04/2024
L2611	DELL	8136BG3	L2 Windows (Portátil intern	\$642.00	SITIO	13/02/2025
L2612	DELL	9Z16BG3	L2 Windows (Portátil intern	\$744.00	SITIO	21/01/2025
L2613	DELL	8H26BG3	L2 Windows (Portátil intern	\$744.00	SITIO	30/06/2023
L2614	DELL	9C36BG3	L2 Windows (Portátil intern	\$642.00	SITIO	29/12/2022
L2615	DELL	JD36BG3	L2 Windows (Portátil intern	\$642.00	SITIO	17/04/2023
L2616	DELL	6RZ5BG3	L2 Windows (Portátil intern	\$642.00	SITIO	29/12/2022
L2617	DELL	DY16BG3	L2 Windows (Portátil intern	\$642.00	SITIO	29/12/2022
L2618	DELL	HD36BG3	L2 Windows (Portátil intern	\$642.00	SITIO	29/12/2022
L2619	DELL	H006BG3	L2 Windows (Portátil intern	\$642.00	SITIO	26/06/2024
L2620	DELL	2Q26BG3	L2 Windows (Portátil intern	\$642.00	SITIO	29/12/2022
L2621	DELL	9B36BG3	L2 Windows (Portátil intern	\$642.00	SITIO	29/12/2022

Mostrar:

15

< 1 2 3 4 5 6 7 8... >

Dar de Alta Datos

- Formulario de llenado de datos del objeto.
- Botón para guardar datos.

Equipos

Formato de Registro

Accesorio PC Portátil

Datos de Registro del Equipo

No. de Serie Folio Descripción

Estatus

Datos Técnicos del Equipo

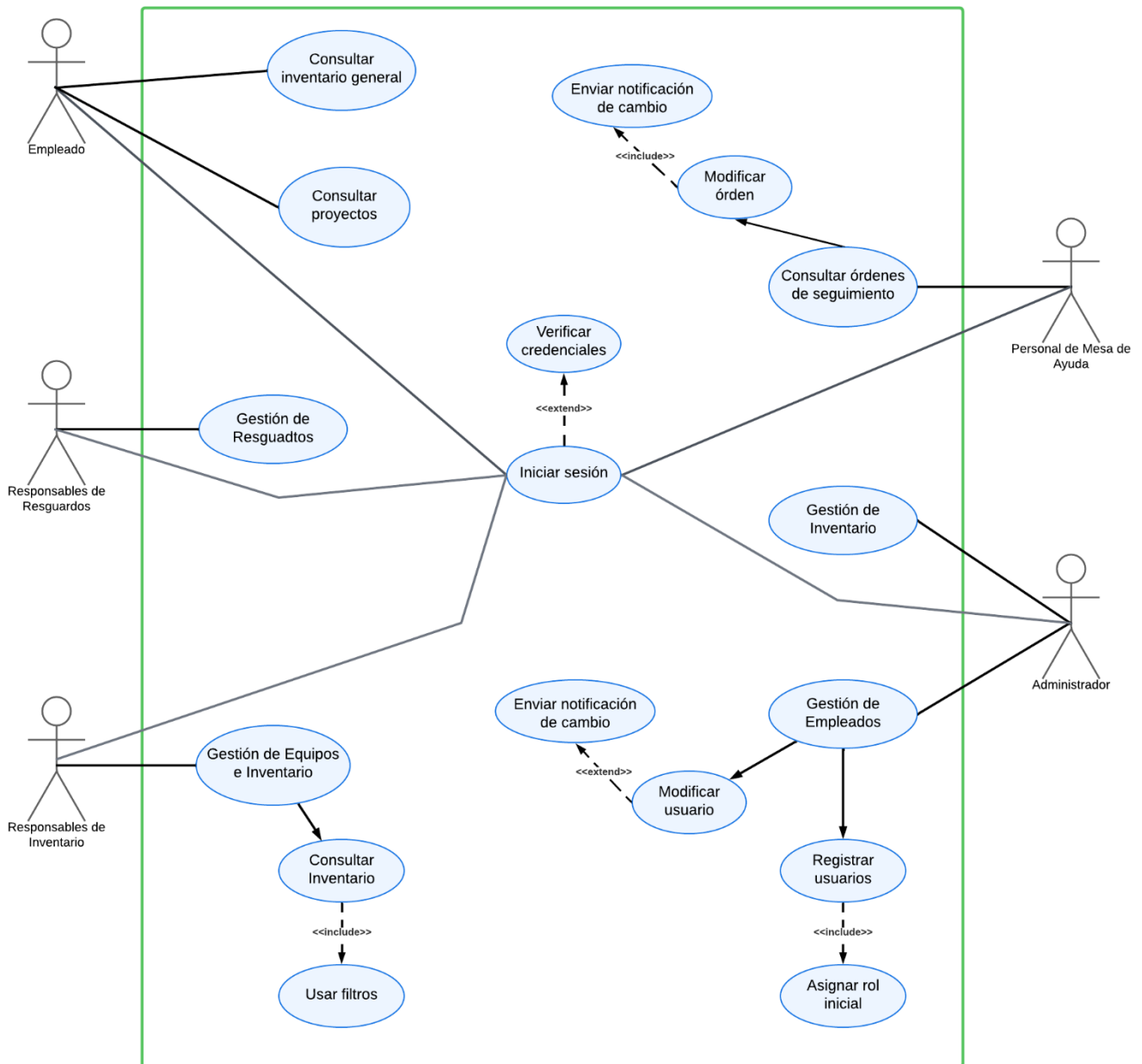
Marca MAC Procesador

DELL DELL HP ASUS MAC Ethernet RAM

Tarjeta de Red Sistema Operativo

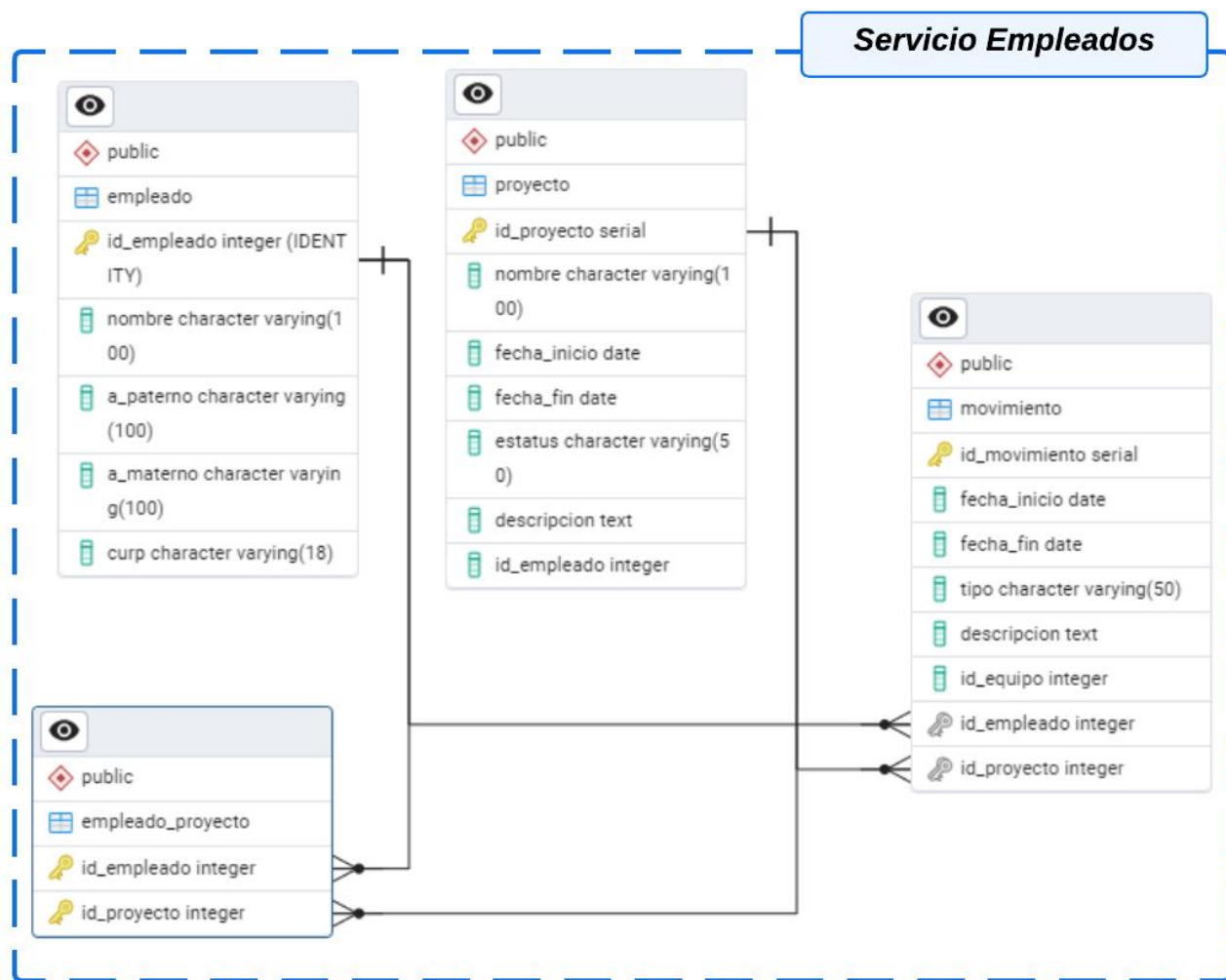
Guardar

CASOS DE USO Y ACTORES

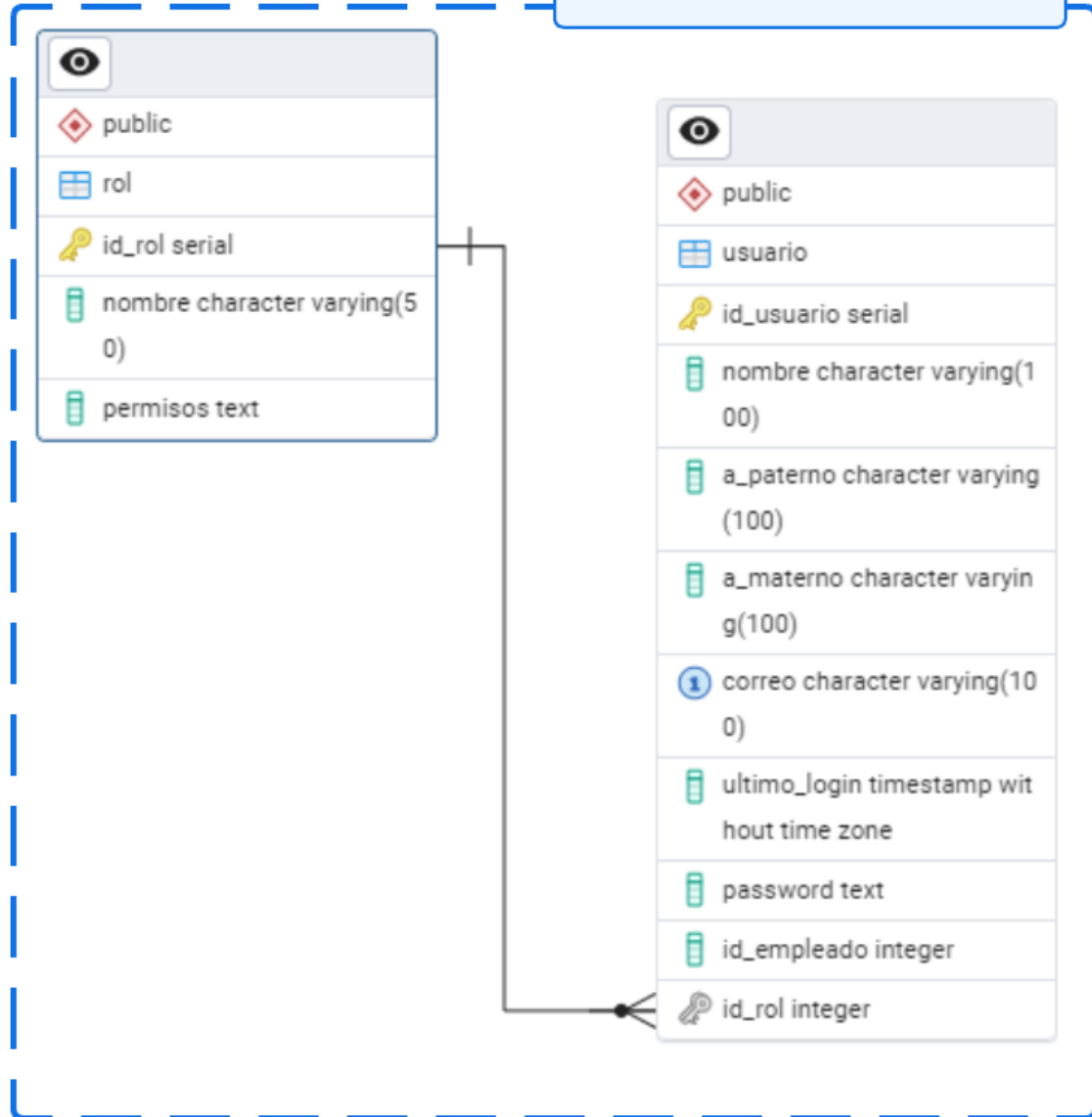


El diagrama de casos de uso del Sistema Interno DISA representa las interacciones clave entre los diferentes perfiles de usuarios y las funcionalidades principales del sistema. Se identifican actores como el Administrador, responsable de Inventario, responsable de Resguardos, Personal de Mesa de Ayuda y Empleados, cada uno asociado a distintos módulos según sus responsabilidades. Las funcionalidades incluyen desde la autenticación de usuarios hasta la gestión de inventario, empleados, resguardos, y el seguimiento de órdenes mediante sistemas externos. Además, se destacan relaciones específicas: los casos de uso marcados con <<include>> indican acciones obligatorias que forman parte de otros procesos, como el uso de filtros al consultar inventario o la asignación de un rol inicial al registrar usuarios. Por otro lado, los casos de uso con <<extend>> representan acciones opcionales o condicionales, como enviar notificaciones al modificar datos o verificar credenciales al iniciar sesión. El diagrama refleja así una estructura modular y clara, que facilita tanto el entendimiento del flujo del sistema como su futura escalabilidad.

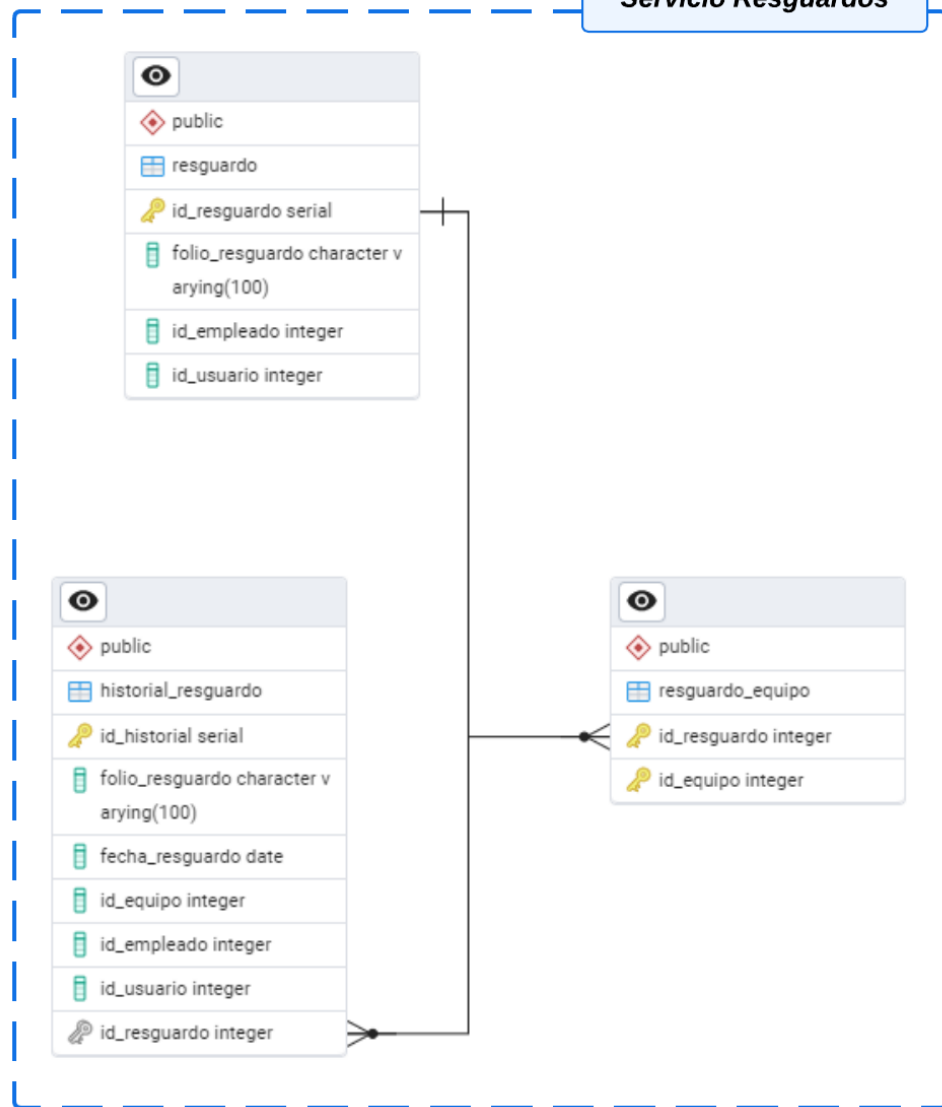
MODELOS DE DATOS / BASES DE DATOS



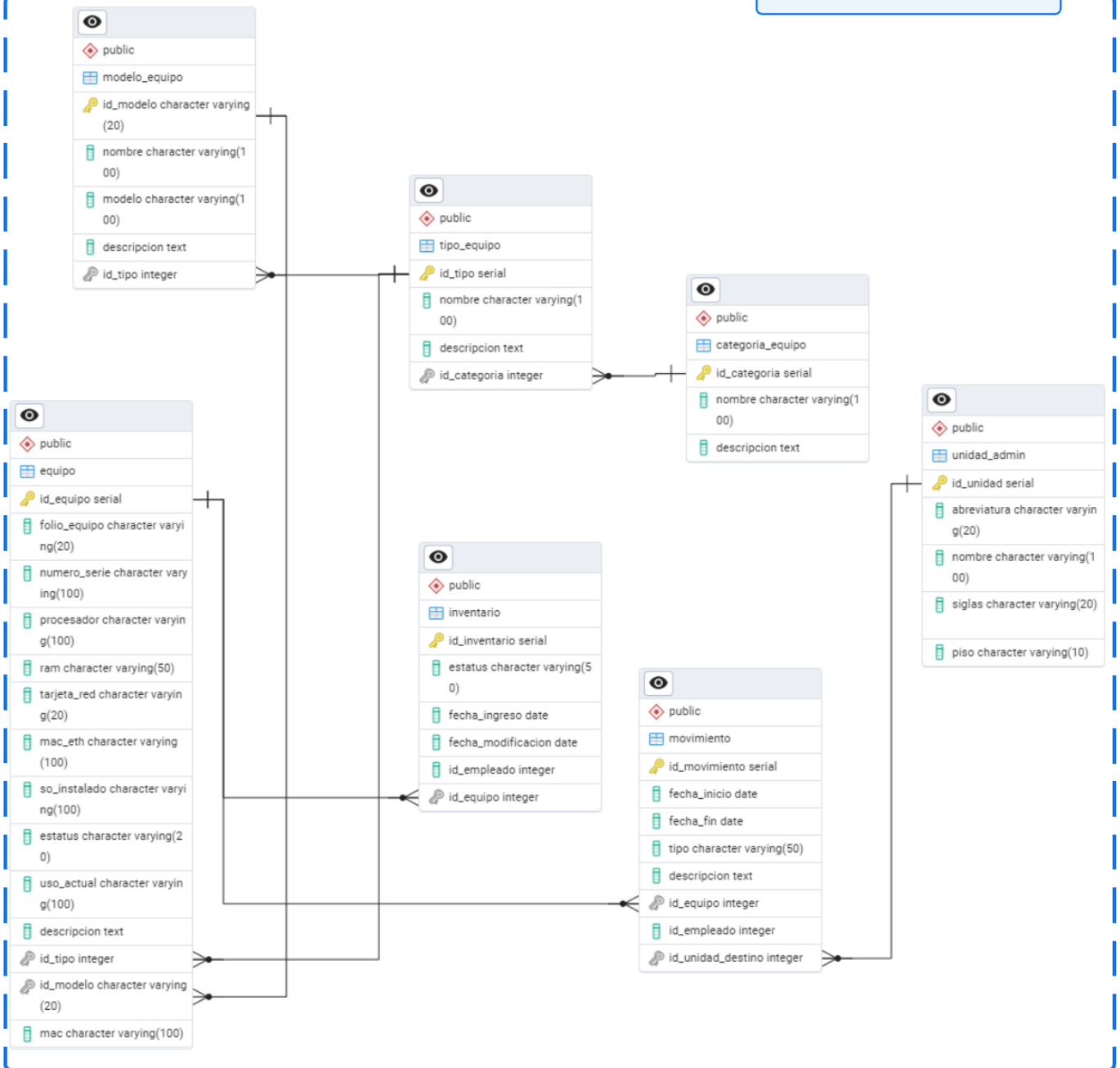
Servicio Autenticación



Servicio Resguardos



Servicio Inventario



Descripción Detallada de los Modelos de Datos:

1. **EQUIPO:**

Almacena la información individual y específica de cada equipo registrado dentro del sistema. Incluye atributos como número de serie, características técnicas, así como claves foráneas que lo vinculan con las tablas **TIPO_EQUIPO**, **CATEGORIA_EQUIPO** y **MODELO_EQUIPO**. Esta estructura permite identificar de forma precisa cada unidad dentro del inventario global.

Nota importante: Originalmente, el identificador principal de la tabla era el campo **folio**, utilizado como clave primaria. Para mejorar modelo eliminando el folio como clave primaria. la estructura y adaptarla a prácticas más robustas, se ha modificado el Ahora, se ha creado un nuevo campo de clave primaria autoincrementable específico para el sistema, asegurando un control interno único. El campo **folio** sigue existiendo como una columna normal, manteniéndose disponible junto con el número de serie de los equipos como referencias clave para búsquedas o relaciones cruzadas dentro del sistema.

2. **TIPO_EQUIPO:**

Catálogo que clasifica los equipos según su naturaleza o función (por ejemplo: laptop, monitor, impresora, accesorio). Esta categoría permite segmentar los equipos para su gestión y visualización diferenciada dentro del sistema.

3. **CATEGORIA_EQUIPO:**

Define las categorías técnicas o comerciales de los equipos, asociándolos a criterios como marca, línea o familia de productos. Su objetivo es agrupar modelos que comparten características generales.

4. **MODELO_EQUIPO:**

Detalla de forma específica los modelos técnicos de los equipos, incluyendo atributos como marca, línea, especificaciones técnicas, configuraciones predeterminadas o ediciones. Funciona en conjunto con **CATEGORIA_EQUIPO** para lograr una clasificación precisa.

5. **INVENTARIO:**

Registra el estado general de los equipos dentro del inventario, almacenando información como ubicación física, código interno, número de inventario, estatus (activo, en mantenimiento, fuera de servicio, etc.), y observaciones adicionales que puedan ser relevantes para la gestión operativa.

6. **UNIDAD_ADMIN:**

Representa las distintas unidades administrativas, áreas o departamentos de la organización a las cuales se asignan o destinan los equipos. Permite conocer de forma estructurada la distribución interna de los recursos.

7. **EMPLEADO:**

Contiene los datos personales y administrativos de los empleados de la organización que utilizan o resguardan equipos. Además, estos registros pueden estar vinculados a proyectos específicos, permitiendo el seguimiento de recursos humanos asociados a tareas o responsabilidades.

8. **RESGUARDO:**

Documento formal que valida la asignación de uno o varios equipos a un empleado determinado. Registra detalles como fecha de entrega, empleado asignado y observaciones pertinentes.

9. **RESGUARDO_EQUIPO:**

Tabla intermedia o puente que relaciona uno o varios equipos con un resguardo específico, habilitando la posibilidad de agrupar múltiples equipos dentro de un solo acto formal de asignación o préstamo a un empleado.

10. **MOVIMIENTO:**

Registra todos los cambios relacionados con los equipos, incluyendo cambios de ubicación, estado, reasignación entre empleados o entre unidades administrativas. Esta tabla funciona como una bitácora de historial, garantizando el seguimiento detallado de cada equipo a lo largo de su ciclo de vida dentro de la organización.

11. **USUARIO:**

Registra a los usuarios autorizados para acceder al sistema. Incluye datos como nombre de usuario, contraseña (protegida mediante mecanismos de cifrado), roles y permisos asociados, permitiendo controlar el acceso a diferentes funcionalidades según el perfil del usuario.

12. **PROYECTO:**

Define los proyectos activos o históricos desarrollados por la organización, en los cuales puede estar involucrado tanto personal como equipos asignados. Esta tabla permite visualizar y gestionar la asignación de recursos a actividades específicas.

13. **EMPLEADO_PROYECTO:**

Tabla puente que vincula registros de empleados con los proyectos en los que han participado o participan actualmente. Refleja relaciones de tipo muchos-a-muchos, ya que un empleado puede estar asignado a múltiples proyectos y un proyecto puede involucrar a varios empleados.

Como parte de las pruebas preliminares y para validar la estructura de las bases de datos antes del desarrollo del backend, se desarrollaron scripts en Python dedicados a la inserción automática de datos falsos o simulados. Estos scripts permiten poblar las tablas clave con registros generados de forma controlada, facilitando pruebas de consultas, relaciones y reportes sin comprometer datos reales o sensibles de la organización. Esta práctica asegura que el comportamiento de los microservicios y las consultas a nivel de base de datos puedan verificarse de manera anticipada, optimizando tiempos de desarrollo y reduciendo posibles errores de integridad durante la implementación del sistema final.

COSAS POR HACER / PRÓXIMAS TAREAS

1. Generación de Datos Falsos:

- Actualizar los scripts de inserción automática de datos falsos en Python, adaptándolos a los cambios recientes en los modelos de datos (ejemplo: nuevo ID autoincrementable en resguardos).
- Poblar todas las tablas clave para realizar pruebas de consulta antes de insertar datos reales.

2. Desarrollo de APIs:

- Implementar los endpoints REST para cada microservicio, incluyendo autenticación, inventarios, resguardos, empleados, proyectos y mesa de ayuda.
- Definir y documentar las rutas y los métodos disponibles por cada servicio.

3. Corrección de Dockerfiles y Entornos:

- Revisar y corregir los Dockerfiles de cada microservicio, asegurando el correcto levantamiento independiente y en conjunto mediante Docker Compose.
- Ajustar volúmenes, redes y puertos expuestos.

4. Pruebas de Consulta:

- Realizar pruebas de integridad y rendimiento en la base de datos utilizando los datos simulados.
- Validar relaciones y claves foráneas en los distintos módulos.

5. Visualización en Power BI:

- Configurar conexiones entre PostgreSQL y Power BI para mostrar dashboards y KPIs utilizando los datos de prueba y posteriormente los reales.

6. Inserción de Datos Reales:

- Planificar y ejecutar la migración de datos reales al sistema, respetando los modelos y evitando pérdidas o duplicación de información.

7. Importación desde Documentos Excel:

- Implementar funciones dentro de la interfaz de usuario para cargar datos desde archivos de Excel, facilitando la actualización masiva de registros.

8. Gestión de Dashboards Internos:

- Desarrollar y afinar dashboards internos dentro del sistema para visualizar de forma sencilla resguardos, inventario, usuarios y métricas clave sin depender exclusivamente de Power BI.

9. Búsqueda de Servicios Cloud para Despliegue:

- Evaluar opciones de nube (como AWS, Azure, DigitalOcean o Railway) para el despliegue del sistema completo, considerando presupuesto y facilidad de integración con Docker.

10. Pruebas Finales de Funcionamiento y Despliegue:

- Ejecutar pruebas de usuario y pruebas técnicas para asegurar el correcto funcionamiento de todos los microservicios y APIs.
- Validar el comportamiento del sistema una vez desplegado en el entorno de producción en la nube.

INSTRUCCIONES BÁSICAS DE DESPLIEGUE LOCAL DEL SISTEMA

Requisitos Previos:

- Docker y Docker Compose instalados.
- PostgreSQL configurado (opcional si se usa Docker para BD).
- Node.js y npm instalados localmente para desarrollo frontend y backend si no se usa contenedor.

Clonación del Proyecto:

- Ubicación recomendada:

```
C:\Users\DELL\Documents\SistemaDISA
```

- Pasos:

```
git clone https://github.com/FerRico18/DISA.git
cd DISA
```

Estructura General del Proyecto:

SistemaDISA/

```
├── frontend/      → Proyecto React (Vite + Chakra UI)
├── backend/       → Microservicios (Node.js APIs + Express)
│   ├── inventario/
│   ├── empleados/
│   ├── resguardos/
│   ├── usuarios/
│   └── mesa_ayuda/
├── database/      → Scripts SQL y configuración inicial
├── docker-compose.yml
└── README.md
```

Configuración de Base de Datos:

- **Motor:** PostgreSQL
- **Contraseña de superusuario:** DISAHP

- **Usuario/DB recomendado:** Crear usuario `disa_user` con permisos limitados para producción.

Importante: La contraseña DISAHP se utiliza también en el archivo `docker-compose.yml` para levantar el contenedor de PostgreSQL.

Despliegue con Docker:

- Desde la raíz del proyecto:

```
docker-compose up --build
```

Asegúrate de que los puertos configurados en el `docker-compose.yml` no estén ocupados localmente.

Acceso al Sistema:

- **Frontend:** `http://localhost:5173` (por defecto con Vite)
- **Backend API Gateway:** Revisar el puerto expuesto en `docker-compose.yml`
- **Base de Datos:** Puede accederse mediante herramientas como DBeaver o PgAdmin usando:
 - Host: `localhost`
 - Usuario: `postgres`
 - Contraseña: `DISAHP`

Notas de Seguridad Importante:

- Las contraseñas de producción y entornos sensibles deben cambiarse antes de hacer un despliegue público.
- Los archivos .env por cada microservicio deben estar configurados correctamente antes de levantar el sistema.

CONCLUSIÓN

El Sistema Interno DISA representa un avance significativo en la gestión y automatización de procesos críticos para la organización, eliminando la dependencia de métodos manuales y fortaleciendo la seguridad y trazabilidad de los datos. La implementación de una arquitectura de microservicios garantiza flexibilidad y escalabilidad a futuro, permitiendo integrar nuevas funcionalidades o módulos según las necesidades operativas de la empresa. Esta documentación proporciona una guía clara tanto para el despliegue técnico como para la comprensión funcional del sistema, asegurando la continuidad del proyecto por parte de cualquier equipo de desarrollo que lo retome en el futuro.