

Logistica Global

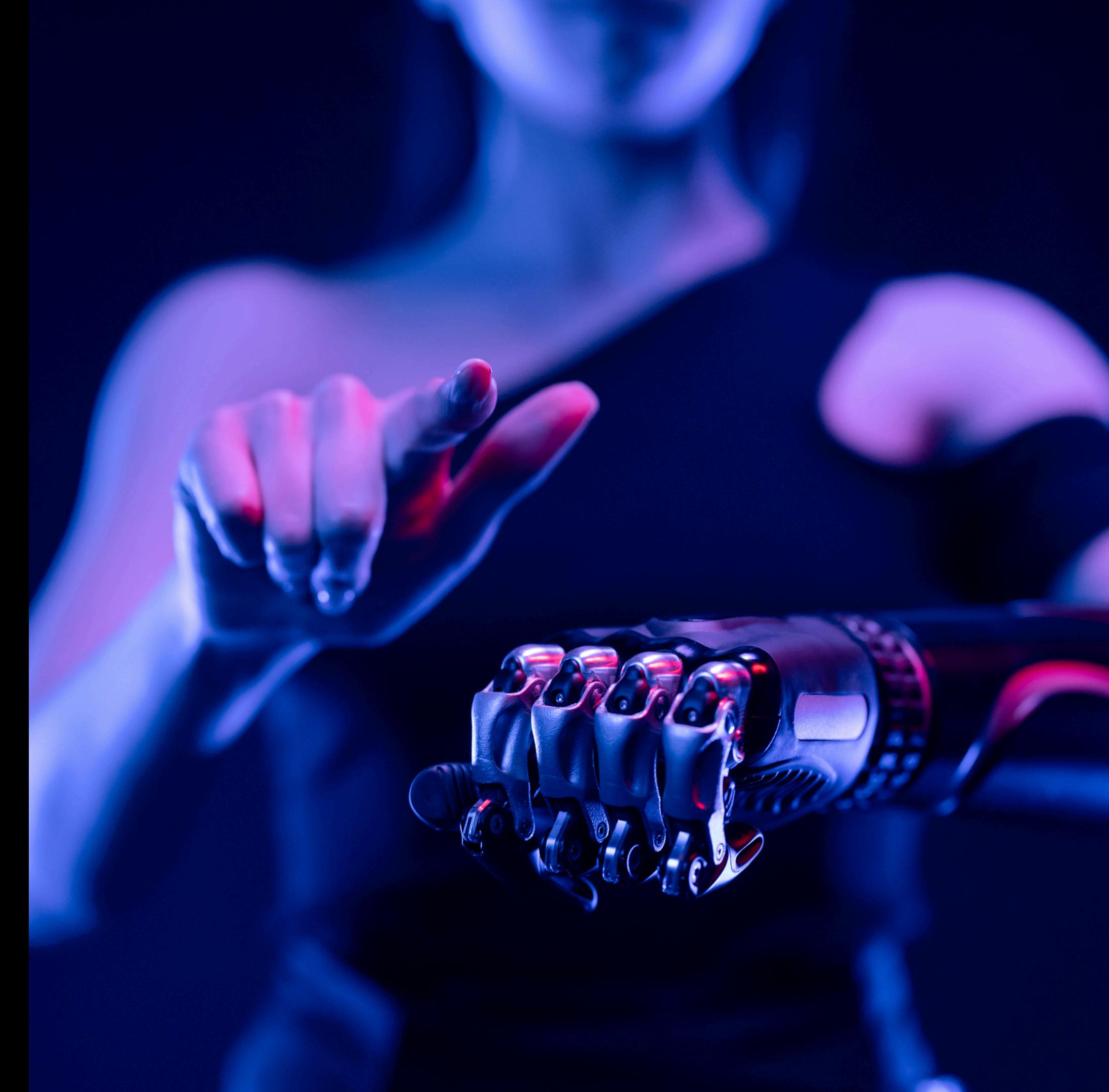
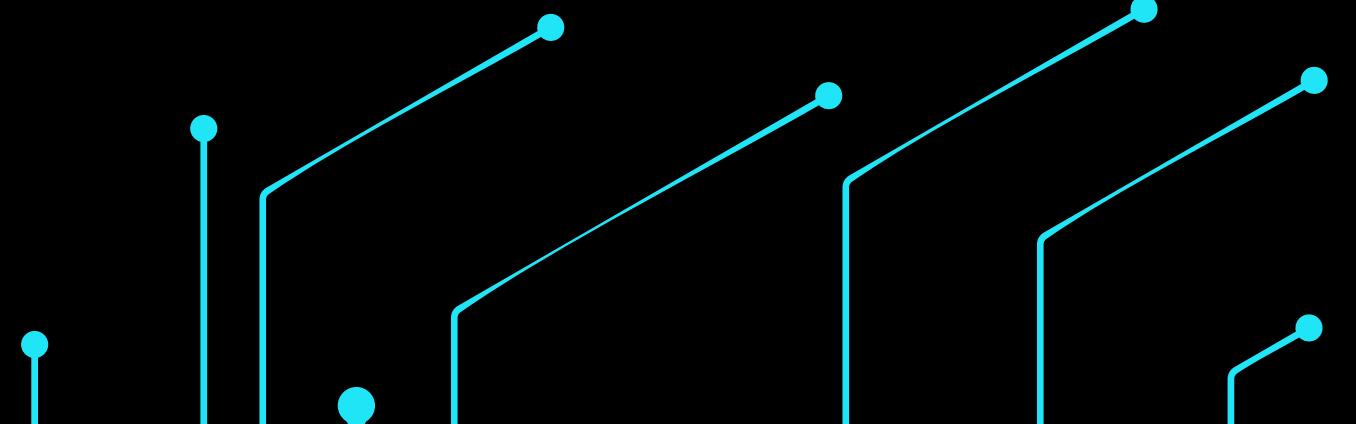
Integrantes:

Pablo Marambio - 202073108-K
Franco Reinoso - 202010021-7
Daniel Villaroel - 202073028-8

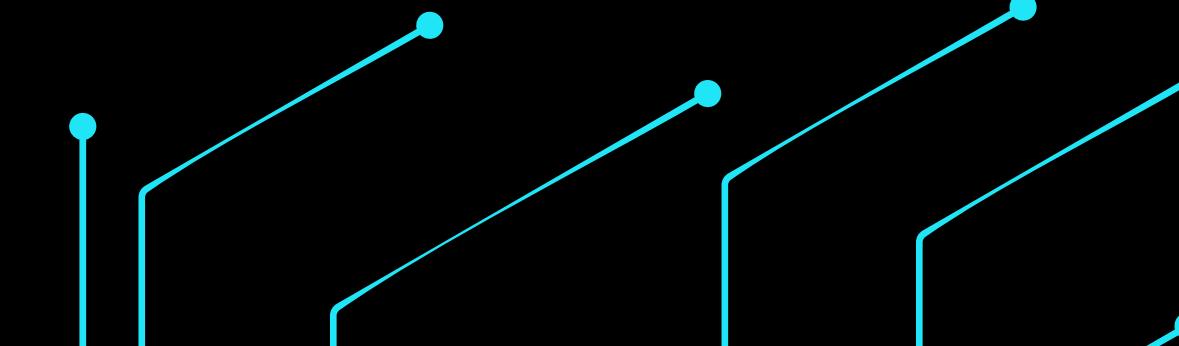
Contenido

- O1** | INTRODUCCION
- O2** | CONTEXTO Y PROBLEMA
- O3** | FUNCIONALIDADES Y AQUITECTURA
- O4** | ESTRATEGIA DE PRUEBAS
- O5** | DEMOSTRACIONES
- O6** | REFLEXION
- O7** | PROXIMOS PASOS

¿Qué es Logística global?



CONTEXTO Y PROBLEMA



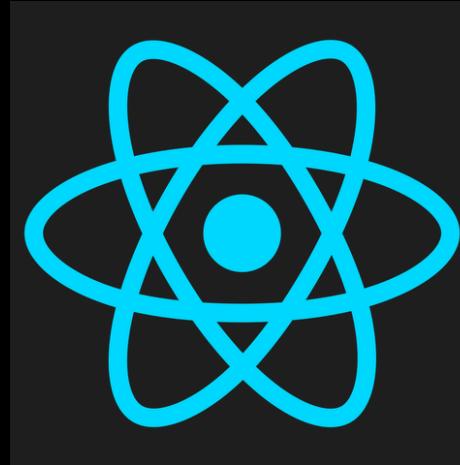
ROBOTS

MANUAL

INCIDENTES

PROBLEMA

TECNOLOGÍAS Y ROLES



- **Frontend:** React
- **Backend:** Nodejs
- **Base de datos:** PostgreSQL
- **Herramienta de testing:** Puppeteer

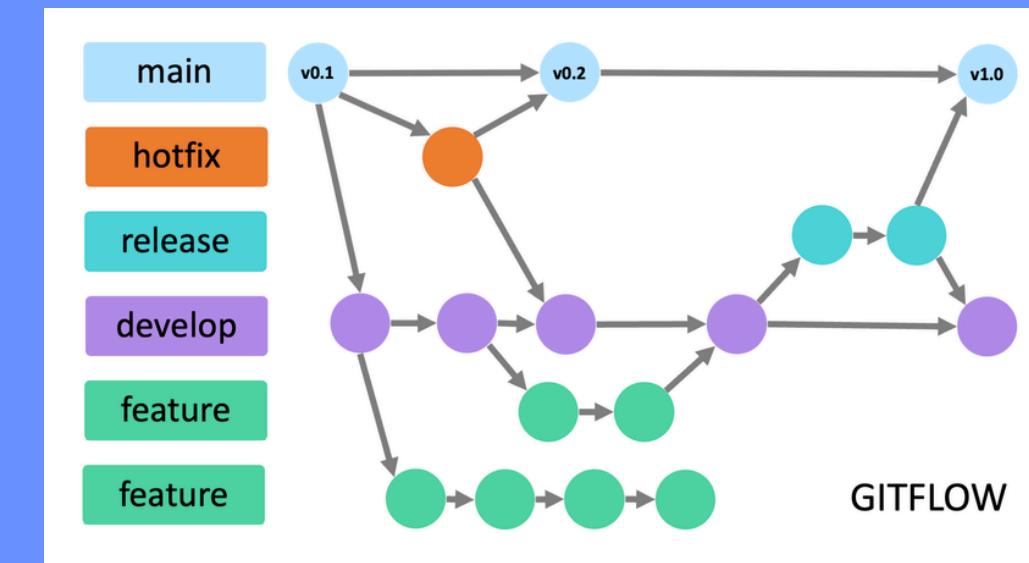
- Roles:**
- **Product Owner:** Pablo
 - **Scrum Master:** Franco
 - **Documentación:** Daniel

Metodologías

SCRUM + KANBAN



GITFLOW



HERRAMIENTAS

A screenshot of a project management tool interface. The top navigation bar includes 'Proyectos', 'LogisticaGlobal.com ...', 'Resumen', 'Cronograma', 'Tablero' (which is selected), and 'Calendario'. Below the navigation are sections for 'POR HACER' (with 5 items) and 'EN CURSO'. A search bar 'Buscar tablero' and filter options are also present. On the right, there's a sidebar for 'Gestion de usuarios' with a checked item 'LC-12' and a 'Panel de Técnicos' section.



Requisitos funcionales

RFO1: El sistema debe permitir registrar un incidente con fecha, hora, ubicación, tipo de fallo y descripción.

RFO2: Debe permitir asociar uno o más robots a un incidente.

RFO3: Cada incidente debe comenzar en estado “Creado”.

RFO4: Se debe poder consultar incidentes por filtro (código de incidente)

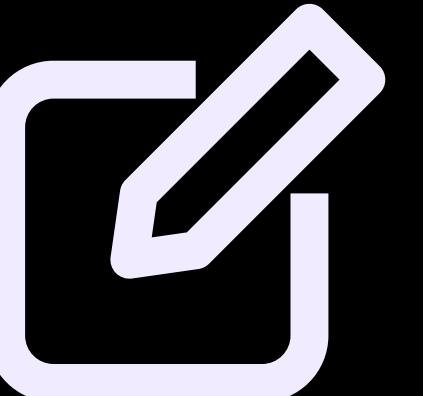
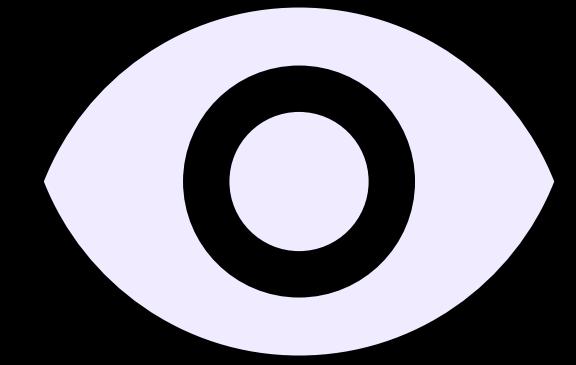
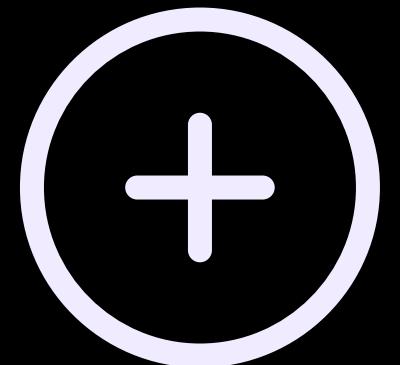
Requisitos no funcionales

RNFO1: Seguridad Los datos deben estar cifrados. Acceso solo con credenciales.

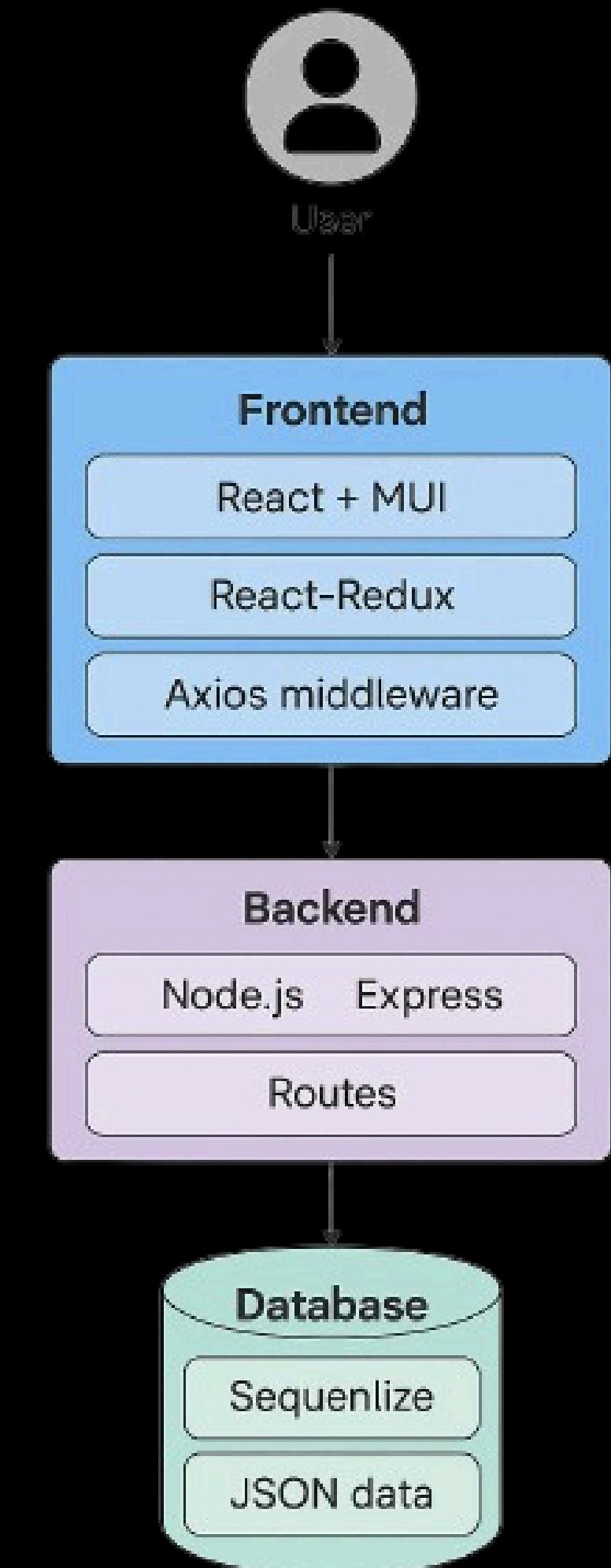
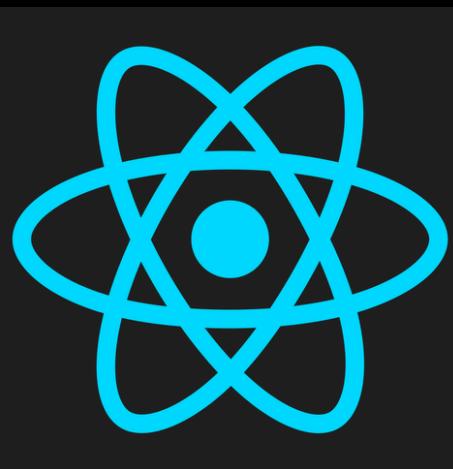
RNFO2: Rendimiento aceptable para cargar listas de incidentes (<2s).

Flujo general

C.R.U.D



Arquitectura



Estrategia de pruebas

BACKEND

- Se verificaron manualmente los endpoints RESTful desarrollados en Node.js + Express, asegurando que:
 - Las operaciones CRUD funcionaran correctamente.
 - La comunicación con la base de datos PostgreSQL fuera efectiva mediante Sequelize.
 - Las respuestas cumplieran con los formatos y estructuras esperadas por el frontend.
 - Se uso Postman para probar cada ruta antes de integrarlas a la rama de desarrollo.

FRONTEND

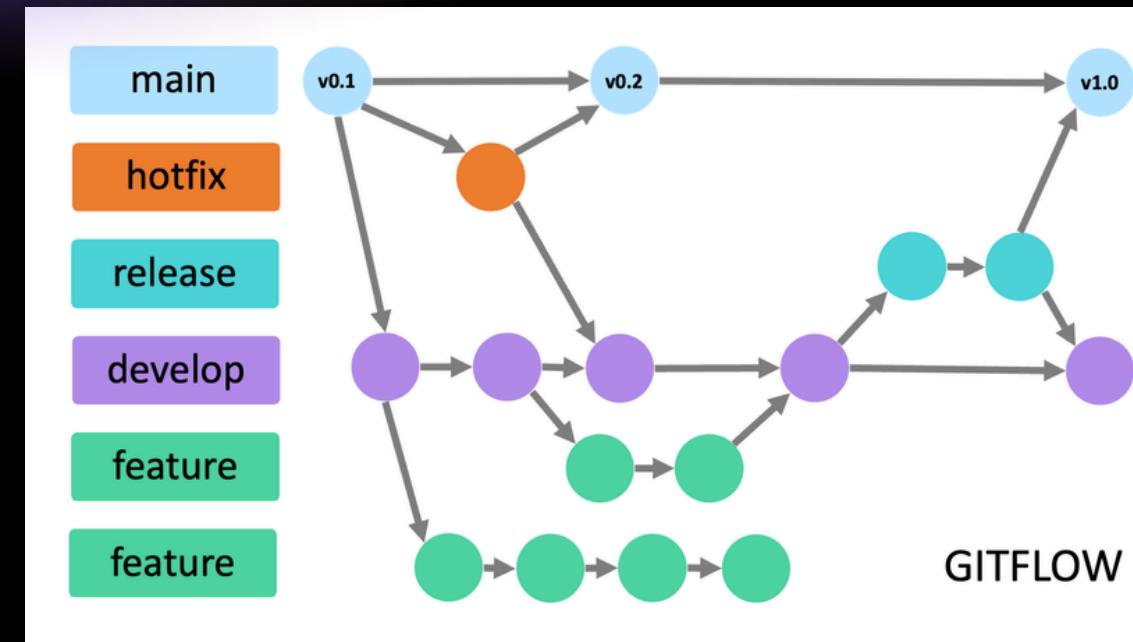
- Revisiones visuales para verificar la correcta integración con el backend
- adecuada representación de estados y datos.



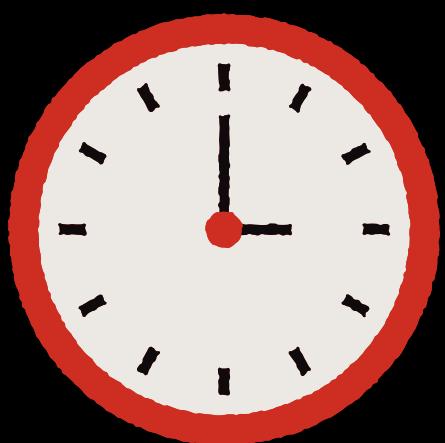
Demostración



COMUNICACION



FLUJO DE TRABAJO



REVISION DE TRABAJO
FUERA DE TIEMPO



DOCUMENTACION

- Sprint Planning Meeting Minutes
 - Registrar decisiones
 - Documento alcance sprint

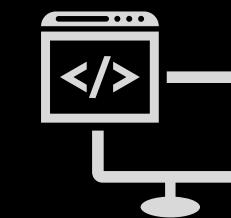
Reflexión



Próximos pasos



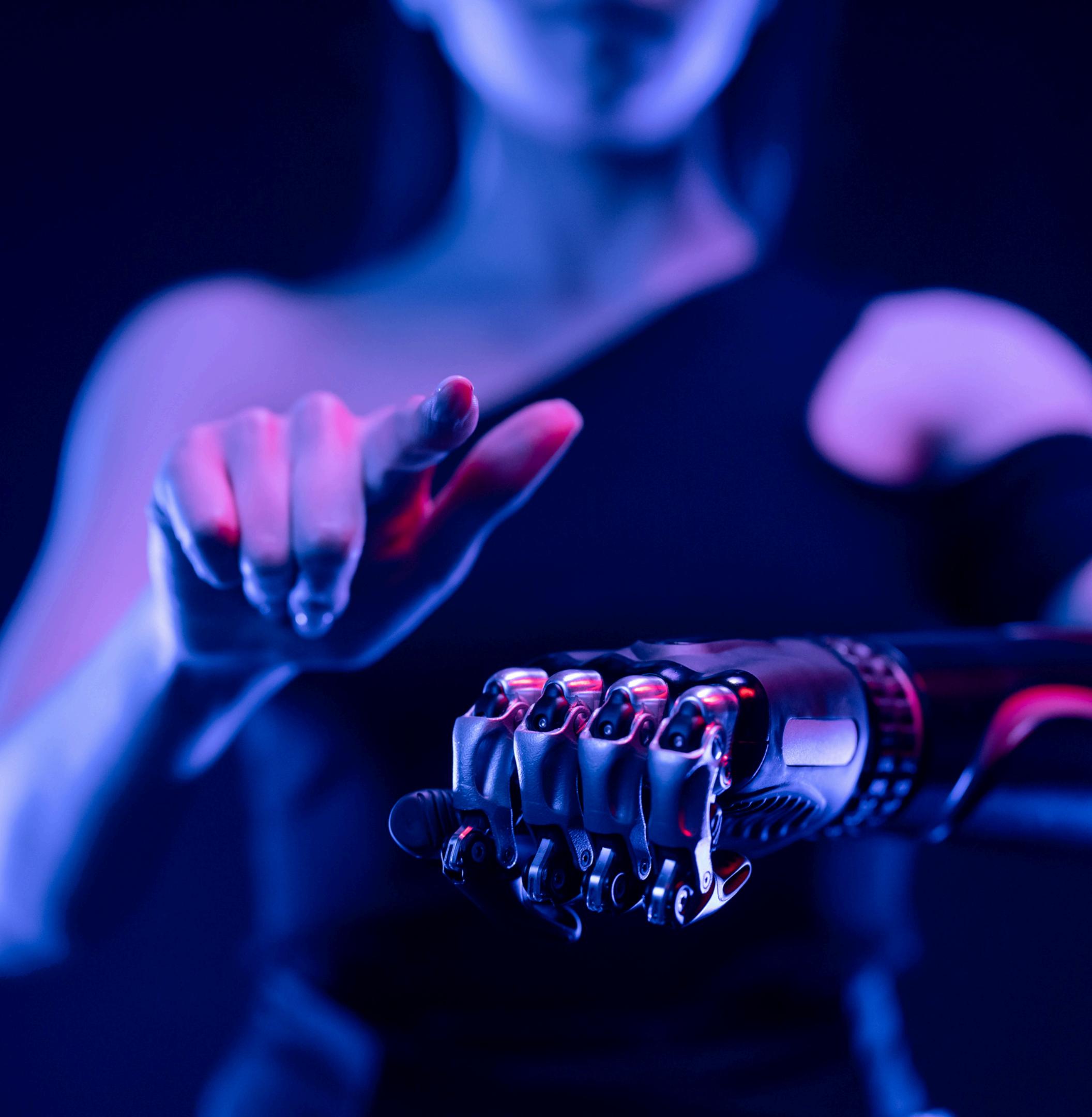
GESTION DE USUARIOS + PRUEBAS AUTOMATIZADAS



INTEGRACION CI/CD



Integración Jenkins



**Gracias por su
atención**

**Proyecto:
Logística
Global**

Integrantes:

Pablo Marambio - 202073108-K
Franco Reinoso - 202010021-7
Daniel Villaroel - 202073028-8