

App-Control-Diario

Prototipo de Digitalización



Proceso de digitalización de los libros del
control diario del plan infoex

Contenido

- 1. Presentación
- 2. Contenido
- 3. Mi elección
- 4. Un reto necesario
- 5. Funcionamiento
- 6. Administrador
- 7. Pantallas Administrador
- 8. Jefes de Servicio
- 9. Pantallas de Jefes de Servicio
- 10. Bomberos
- 11. Pantallas Bomberos
- 12. Frontend
- 13. Frontend Tecnologías
- 14. Frontend Usos
- 15. Backend
- 16. Backend Tecnologías
- 17. Arquitectura
- 18. Despliego GitHub
- 19. Despliego Vercel
- 20. Fin

Mi elección

- Muchas empresas aún usan papel para sus jornadas
- Mi propuesta digitalizar este proceso
- Adaptable a servicios públicos y privados
- Mejora organización y reduce errores
- Acceso claro a tipo de jornada, horas extra y solicitudes
- Responde a una necesidad real del personal



‘Un reto necesario’



- **Solucionan** la falta de información en tiempo real
- **Sustituye** calendarios improvisados por una herramienta propia
- **Impulsa** la digitalización del sector público
- **Reduce uso de papel** y protege los bosques
- **Aprovecha** los avances tecnológicos actuales
- Proyecto ambicioso pero alcanzable

Funcionamiento

Tres roles que dan sentido al sistema

- La app funciona mediante tres roles conectados
- Cada rol tiene permisos y funciones propias
- La comunicación entre ellos garantiza coherencia
- El sistema controla accesos y valida procesos
- Aporta orden y seguridad



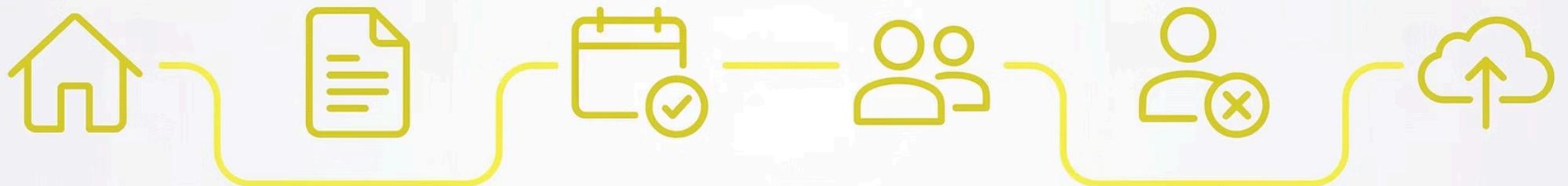
Administrador

Control y supervisión



- Supervisa y valida toda la información
- Recibe anotaciones del jefe de servicio
- Revisa solicitudes de los bomberos
- Controla las reglas del negocio
- Mantiene coherencia y orden en los datos

Funcionalidades de los Administradores



Listar
Datos

Datos
Personales

Resumen
Mensual

Agregar
Bombero

Eliminar
Bombero

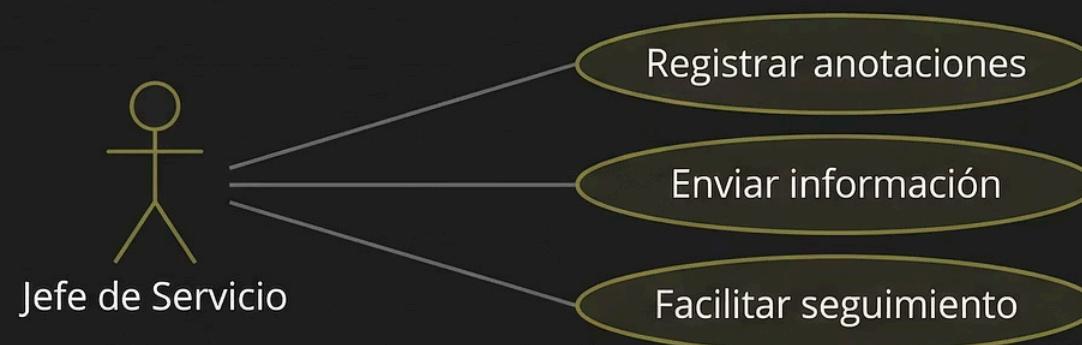
Importar
Usuarios

Jefes de Servicio

Gestión diaria del equipo



- Registra las anotaciones diarias del personal
- Añade también sus propias anotaciones
- Envía la información al administrador
- Facilita el seguimiento de turnos
- Es clave en la operativa diaria



Funcionalidades de los Jefes de Servicio



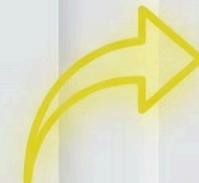
Inicio

El inicio de sesión de los Jefes de Servicio les permite consultar sus datos como un bombero o continuar con las funcionalidades propias



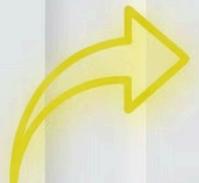
Agregar Bombero

En esta pantalla se agregan los usuarios que están el día en concreto trabajando, pueden ser del equipo A o B



Anotaciones Diarias

Una vez agregados los usuarios del día, se anotan sus códigos de trabajo



Salidas Diarias

Por último en esta ventana se agregan las intervenciones que se realizan durante el día



Bomberos

Usuario final del sistema

- Perfil más numeroso de la aplicación
- Consulta sus datos y registros
- Envía solicitudes: días libres, horas extra...
- Estas son validadas por el administrador
- Interfaz simple y pensada para uso móvil

Funcionalidades de los Bomberos



Inicio

En la pantalla de inicio del bombero se ofrece un mensaje de bienvenida y una introducción a las funciones que puede realizar



Mis Datos

En esta ventana los bomberos ven todos sus datos personales y un resumen de la actividad que han tenido los últimos días



Enviar Solicitud

Aquí se realizan las solicitudes de los días que se desean descansar y se enviarán a los administradores para ser revisadas



Resumen Mensual

Esta pantalla se muestra el resumen mensual de todos los registros, solicitudes

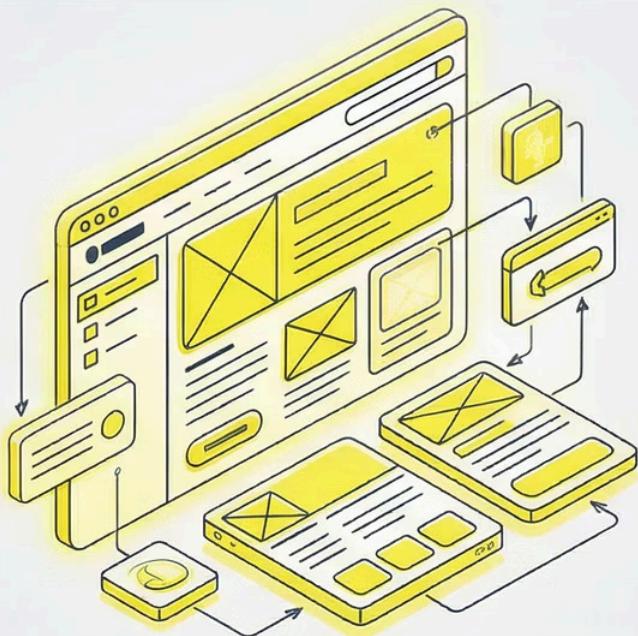


Cambiar Contraseña

El cambio de contraseña para que cada bombero tenga seguridad en sus datos

Frontend

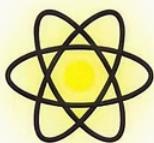
Next.js: la base del frontend



- Framework construido sobre React
- Genera rutas y estructura el proyecto automáticamente
- Ofrece gran rendimiento y carga rápida
- Permite páginas dinámicas y seguras para cada rol
- Facilita la comunicación con Supabase
- Ideal para crear prototipos escalables y ordenados

Frontend

Tecnologías que impulsan el frontend



React: componentes dinámicos y reutilizables



Tailwind CSS: estilos rápidos y consistentes



Shadcn/ui + Redux: componentes modernos y personalizables



Iconos y patrones UI: para mejorar usabilidad



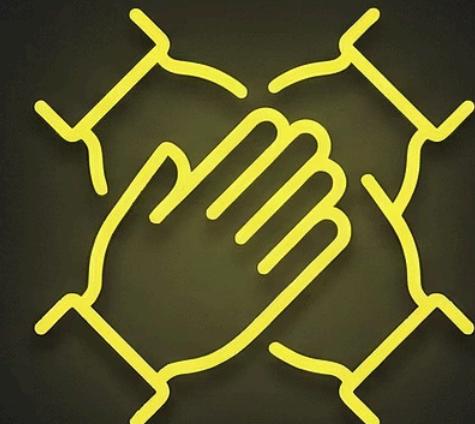
Node Modules: dependencias del proyecto



Frontend

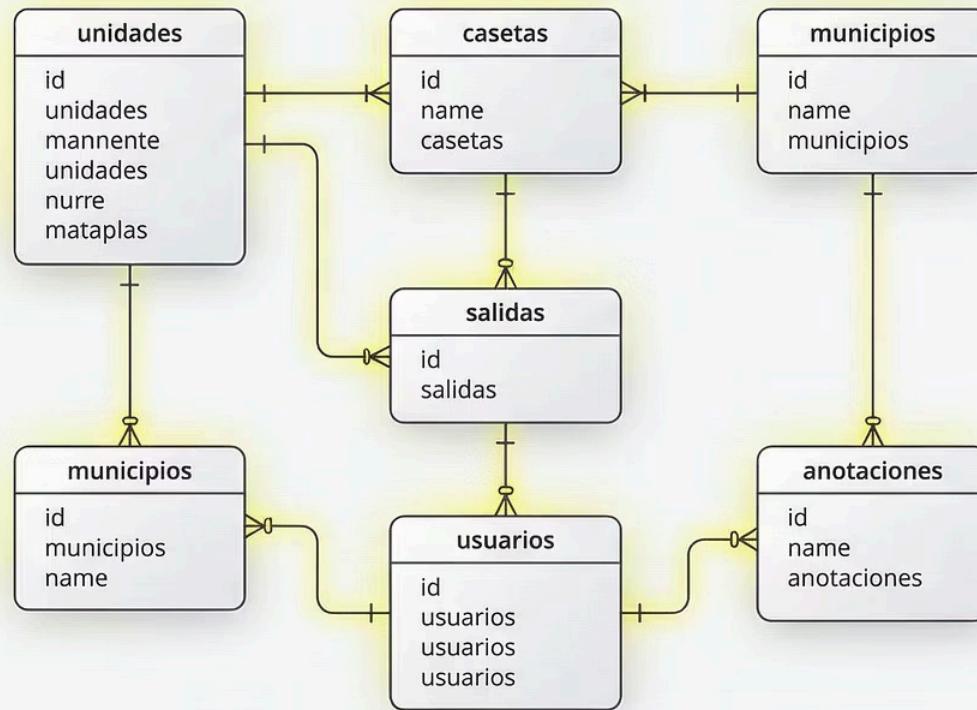
¿Por qué usar estas tecnologías?

- Next.js ofrece rapidez, rutas claras y buen rendimiento
- React permite componentes reutilizables y escalables
- Tailwind agiliza el diseño sin código repetido
- Shadcn/ui aporta estilo moderno sin esfuerzo extra
- Redux mejora la gestión del estado en pantallas clave
- Conjunto flexible y perfecto para nuevas aplicaciones



Backend

¿Por qué Supabase?



- Plataforma moderna compatible con Postgres
- Gestión fácil de usuarios y autenticación
- Consultas eficientes y actualizaciones en tiempo real
- Reduce tiempos de desarrollo del backend
- Perfecto para prototipos y proyectos escalables
- Garantiza seguridad y consistencia

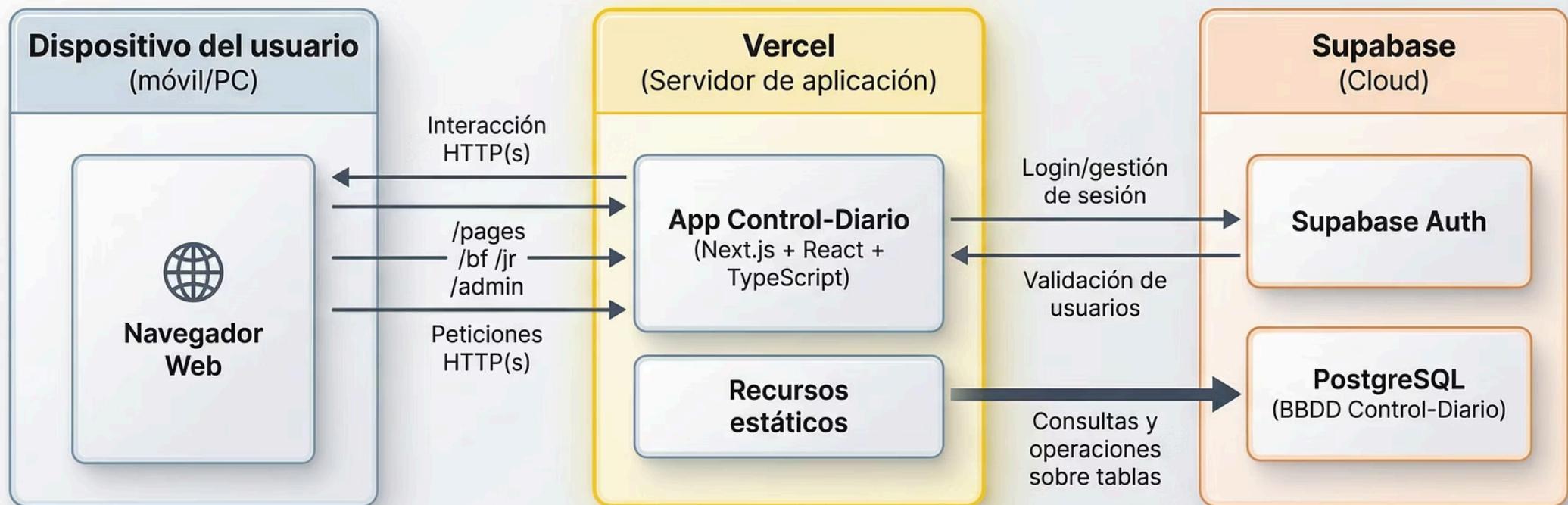
Backend

Datos organizados y dinámicos



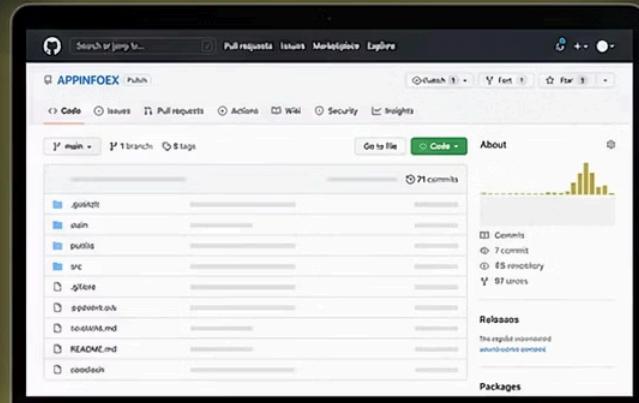
- Archivo JSON propio con destinos oficiales
- Define zonas y lugares de trabajo
- Importado a Supabase como tablas dinámicas
- Permite selectores fáciles en el frontend
- Asigna ubicaciones precisas a cada bombero
- Sistema centralizado y fácil de actualizar

Arquitectura



Despliegue

Código gestionado desde GitHub



- Proyecto versionado en GitHub
- Ramas para control de cambios y mejoras
- Commits validados con Husky (seguridad y formato)
- Subida automática tras cada actualización
- Repositorio vinculado directamente con Vercel
- Preparado para despliegue continuo

 github.com/JavyProgramador/APPINFOEX

Despliegue

Vercel: despliegue rápido y automatizado



- Despliegue automático al actualizar GitHub
- Optimizado para aplicaciones Next.js
- Servidores globales: carga más rápida
- Certificado HTTPS incluido
- Build y preview inmediatos
- Facilita mantener el proyecto siempre online

Gracias por todo

