ChatGPT

IES José Mor de Fuentes Proyecto Final DAW Curso 2024-2025

Datos del proyecto

- Código: IFC_303_999
- Título: Aplicación Gestor de alquileres SECUO
- Alumno: Oumar Traore
- Profesor Tutor: Modesto Sierra Callau
- Convocatoria: Junio

---

1. RESUMEN

Español SECUO es una aplicación MERN (MongoDB, Express, React, Node.js) orientada a facilitar la comunicación entre propietarios e inquilinos, permitiendo la gestión de reportes de mantenimiento, contratos de alquiler, perfiles y documentos validados, todo en tiempo real mediante sockets.

English SECUO is a MERN-based web app designed to streamline communication between landlords and tenants, offering maintenance reporting, rental contract management, profile/document validation, and real-time interaction using sockets.

2. PALABRAS CLAVE

Español Mantenimiento | Reportes | Propietarios | Inquilinos | Servicios | Validación | Contratos | Chat

English Maintenance | Reports | Landlords | Tenants | Services | Validation | Contracts | Chat

3. INTRODUCCIÓN

SECUO nace de la necesidad de una plataforma integral que centralice la comunicación, el mantenimiento y la validación documental entre inquilinos y propietarios. Además, el sistema cuenta con administradores que aprueban documentos e identidades para asegurar la transparencia y legalidad del sistema.

4. ANÁLISIS

Requisitos funcionales

- Gestión de usuarios: registro/login, validación, roles.
- CRUD de propiedades, contratos, reportes.
- Sistema de chat en tiempo real.
- Notificaciones push en tiempo real.
- Validación de documentos por administradores.

Requisitos no funcionales

- Seguridad JWT.
- Escalabilidad modular (frontend/backend desacoplado).

Requisitos de software

- Node.js, Express, MongoDB, React, Vite, Socket.IO, Mongoose.

Requisitos de hardware

- Servidor cloud (4 vCPUs, 8GB RAM, 100GB SSD).

Marco legal

- Cumplimiento GDPR.
- Términos de uso, políticas de privacidad, validez documental.

Tecnologías

- MERN Stack, Axios, WebSockets, Multer (uploads), JWT, React Router, Context API.

5. PLANIFICACIÓN

Fase	Inicio	Fin	Horas
Análisis	10/03/2025	15/03/2025	12
Diseño	20/03/2025	31/03/2025	12
Codificación	31/03/2025	20/04/2025	50
Backend	12/04/2025	20/04/2025	40
Frontend	25/04/2025	30/04/2025	23
Pruebas	01/05/2025	15/05/2025	17
Despliegue	16/05/2025	05/06/2025	8

6. ESTIMACIÓN DE RECURSOS

Humanos

Rol	Horas	Precio/Hora	Total
Backend Developer	40	35€	1400€
Frontend Developer	23	35€	805€
Tester	17	30€	510€
DevOps	8	45€	360€
Total			**3075€**

Materiales

Servidor, herramientas de desarrollo (Vite, Figma, MongoDB Atlas), coworking, SSL, dominio y dispositivos de prueba. Estimación: 5000€.

7. DISEÑO

- Arquitectura MVC en backend.
- Componentes reutilizables en frontend.
- Manejo de estado con Context API.
- Separación de roles por rutas protegidas y validaciones JWT.
- Socket.IO en servidor y cliente para notificaciones/chat.

8. DESARROLLO

Decisiones

- Implementar sockets para eventos críticos.
- Middleware de validación para protección de rutas.
- Modularización por carpetas: users, contracts, reports, etc.
- Implementación de pruebas manuales y automáticas con Postman.

Incidencias

- Problemas de rutas y autenticación al principio.
- Validaciones en el frontend no sincronizadas con backend.
- Problemas de permisos entre propietarios/inquilinos.

9. PLAN DE PRUEBAS

- Rutas API testadas con Postman (users, spaces, contracts, reports).
- Pruebas de UI documentadas.
- JWT funcional y validado para roles.

10. MANUAL DE DESPLIEGUE

- Clonar el repositorio.
- Ejecutar `npm install` en `/frontend` y `/backend`.
- Lanzar `npm run dev` en ambos.
- MongoDB debe estar corriendo o configurado en `.env`.
- Se dispone de un script `.bat` para facilitar el despliegue local.

11. MANUAL DE USUARIO

Inquilino

- Reportar problemas.
- Ver historial.
- Chatear con propietarios.

Propietario

- Ver propiedades.
- Crear contratos.
- Gestionar reportes.

Administrador

- Validar usuarios y documentos.
- Aprobar/rechazar espacios y contratos.

12. RECURSOS EMPLEADOS

- Backend: Node.js, Express, Mongoose.
- Frontend: React + Vite, Context API, Axios.
- Base de datos: MongoDB.
- Deploy: local y automatizado.

13. ASPECTOS LABORALES Y EMPRESARIALES

- Propuesta de valor clara: gestión documental y de mantenimiento.
- Plan de monetización: suscripciones premium.
- DAFO, PEST y modelo de negocio tipo SaaS por validar.

14. CONCLUSIONES

El desarrollo de SECUO me ha permitido integrar tecnologías modernas, resolver problemas reales de gestión documental, y afianzar conceptos de arquitectura full-stack, pruebas y despliegue.

15. BIBLIOGRAFÍA

- https://reactjs.org/
- https://expressjs.com/
- https://mongoosejs.com/
- https://socket.io/docs/
- https://developer.mozilla.org
- Documentación de Postman, Vite, JWT, y MongoDB Atlas