

Proyecto CRM: CrossoverX

ÍNDICE

1. Justificación del proyecto	2
2. Introducción	3
3. Objetivos	4
4. Desarrollo	5
4.1 Fundamentación teórica	5
4.2 Materiales y métodos	6
4.3 Resultados y análisis	8
5. Conclusiones	9
6. Líneas de investigación futuras	10
7. Bibliografía	11
8. Anexos	12
9. Otros puntos	13

1. JUSTIFICACIÓN DEL PROYECTO

El proyecto CRM CrossoverX surge como una necesidad real dentro del contexto de una tienda online especializada en productos tecnológicos. La empresa requiere una herramienta capaz de centralizar la información de clientes, gestionar incidencias, facilitar el seguimiento comercial y optimizar la toma de decisiones mediante un sistema moderno, escalable y de fácil mantenimiento.

La elección de Angular para el desarrollo responde al objetivo de obtener una aplicación robusta, modular y con alto rendimiento, alineada con las prácticas actuales de desarrollo frontend. Debido a la necesidad de datos estructurados

para pruebas y a la ausencia de una base de datos pública adecuada sobre la Liga Española de baloncesto y sus equipaciones

2. INTRODUCCIÓN

Este documento recoge el proceso completo de diseño, desarrollo e implementación del CRM CrossoverX. El frontend se desarrolló utilizando **Angular 18**, adoptando un enfoque modular basado en componentes reutilizables, comunicación asíncrona eficiente, gestión de estado mediante Signals y principios de diseño orientados al usuario.

Debido a que el repositorio actual se encuentra en desarrollo y aún no está completado, este documento se ha actualizado para reflejar fielmente el estado real del proyecto, sus avances, módulos implementados de forma parcial y las funcionalidades pendientes.

Asimismo, se describen los fundamentos teóricos, decisiones tecnológicas, métodos de trabajo y resultados obtenidos hasta el momento, incluyendo la creación de una API propia ante la falta de una fuente de datos fiable.

3. OBJETIVOS

Objetivo general

Desarrollar un CRM funcional con Angular 18 que permita gestionar clientes, incidencias y tareas comerciales, integrando herramientas modernas de asincronía y reactividad.

Objetivos específicos

- Diseñar una arquitectura modular escalable basada en componentes independientes.
- Implementar comunicación eficiente con un backend simulado mediante Observables y RxJS.
- Aplicar Signals de Angular para una gestión de estado interna simple y eficaz.
- Crear interfaces intuitivas y orientadas a la experiencia del usuario.
- Integrar un sistema de notificaciones en tiempo real.
- Generar un entorno de pruebas mediante una API.
- Documentar detalladamente el proceso de desarrollo.

4. DESARROLLO

4.1 Fundamentación teórica

(Se mantiene la base del borrador, ampliada en relación al estado actual del repositorio.)

El sistema se fundamenta en principios modernos de desarrollo frontend, donde Angular 18 ofrece un entorno estructurado basado en componentes, servicios y herramientas de reactividad integradas. La incorporación de Signals permite un flujo de datos predecible y optimizado, lo cual mejora la eficiencia del renderizado.

4.2 Materiales y métodos

Estrategias de búsqueda

- Documentación oficial de Angular, RxJS y patrones arquitectónicos.
- Consultas en repositorios GitHub.
- Revisión de casos reales de CRM empresariales.
- Investigación de bases de datos públicas deportivas.

Al no encontrarse una base actualizada de la Liga Española con equipaciones, se empezó a desarrollar una API propia.

Metodología

Se aplicó una metodología incremental con control de versiones mediante GitHub. Dado que el repositorio **no está finalizado**, algunas funcionalidades permanecen en fase de implementación.

Hitos alcanzados:

1. Diseño inicial de componentes, rutas y servicios.
2. Estructura base con Angular 20.
3. Configuración de comunicación asíncrona con Maven Project.
4. Implementación parcial de los módulos de clientes e incidencias.
5. Inicio del desarrollo del dashboard.

6. Integración inicial de Signals.
7. Pruebas manuales en navegador.

Técnicas utilizadas

- Angular 18
- TypeScript
- RxJS / Observables
- Signals
- Maven Proyect
- Bootstrap / CSS
- GitHub

4.3 Resultados y análisis

Hasta la fecha, se han implementado parcialmente los módulos principales del CRM. El listado de clientes y el sistema de incidencias presentan un funcionamiento inicial estable, aunque todavía requieren optimización.

Se identificaron limitaciones como:

- Necesidad de mejorar la actualización de datos reactivos.
- Pendiente la implementación completa de notificaciones y dashboard.

Aun así, la arquitectura se mantiene escalable y preparada para integraciones futuras.

5. CONCLUSIONES

El CRM CrossoverX, incluso en su estado actual no finalizado, demuestra el potencial de Angular 18 para desarrollar aplicaciones modernas, reactivas y escalables.

6. LÍNEAS DE INVESTIGACIÓN FUTURAS

- Mejora de la API y Base de datos
- Integración de autenticación mediante JWT u OAuth.
- Implementación completa de WebSockets para notificaciones.