



Plan de Proyecto

Prototipo de Página de Comparación de Productos

Julio, 2025

Federico Carbonell, PL Meli

Objetivo de la solución

Desarrollo de un prototipo simplificado de una página de comparación de ítems, acompañada de una API backend que proporciona los detalles de los productos. El objetivo principal es demostrar la capacidad de construir una solución funcional que permita una comparación clara y amigable para el usuario, utilizando un stack tecnológico moderno.

Características clave:

Frontend: Página web responsiva que muestra detalles de productos lado a lado (imágenes, nombres, descripciones, precios, ratings, especificaciones clave).

Backend: API RESTful básica que sirve los datos de los productos desde un archivo JSON local.

Documentación: Plan de proyecto de alto nivel que abarca la estrategia técnica, el uso de herramientas y la planificación de recursos y tiempos.

Estrategia técnica

Backend: .NET 8 (C#) Web API

Se eligió **.NET 8** por su robustez, rendimiento, madurez del ecosistema y familiaridad del equipo con el framework, lo que permite un desarrollo ágil y rápido para este prototipo. Es una excelente opción para construir APIs escalables.

Frontend: React

Seleccionado por su popularidad, eficiencia en la construcción de interfaces de usuario dinámicas y responsivas.

Integración de GenAI y Herramientas Modernas

Gemini: Utilizado para generar ideas iniciales sobre la estructura del JSON de datos, posibles diseños de UI (como los estilos CSS básicos para la grilla de productos), y la redacción de la documentación. Esto acelera la fase de conceptualización y reduce el tiempo de "lienzo en blanco"

GitHub Copilot: Para acelerar la escritura de código repetitivo, sugerir patrones comunes en C# y React, y detectar posibles errores o refactorizaciones.

Visual Studio Code: Para navegación rápida, depuración eficiente y refactoring automático.

Roles y Responsabilidades del Equipo

Project Leader (PL)

- Definición del producto, recopilación de requisitos, priorización de funcionalidades.
- Coordinación y **acuerdos** con equipos relacionados
- Gestión del backlog, comunicación con stakeholders.
- Gestión del equipo

Tech Leader (TL)

- Definición de la arquitectura técnica, elección del stack, aseguramiento de la calidad del código.
- Mentoría del equipo de desarrollo, resolución de impedimentos técnicos.

Backend Developers (2)

- Diseño e implementación de la API (endpoints, lógica de negocio, manejo de datos)
- Integración con fuentes de datos..
- Pruebas unitarias, integración.y regresiones/

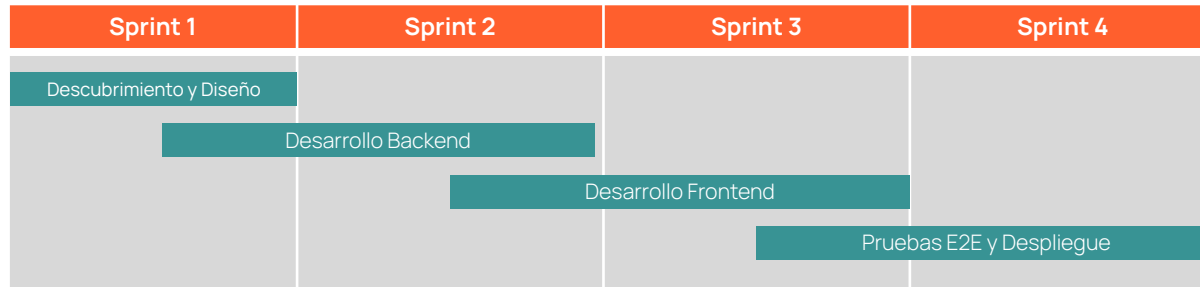
Frontend Developers (3)

- Desarrollo de la interfaz de usuario (componentes React, estilos).
- Consumo de la API backend, manejo del estado de la UI.
- Aseguramiento de la responsividad y la experiencia de usuario.
- Pruebas unitarias y E2E

UX (Ad-hoc)

- Diseño de wireframes, mockups y flujos de usuario.
- Validación de usabilidad (tests con usuarios)

Roadmap, hitos clave y dependencias



*Sprints de 2 semanas

| Fase | Hitos | Dependencias | Sptint |
|-------------------------|--|---|--------|
| Descubrimiento y Diseño | <ul style="list-style-type: none">Definición de requisitos y User Stories.Diseño de la arquitectura de la API y modelos de datos.Diseño de UI/UX. | <ul style="list-style-type: none">Colaboración entre PL, TL y UX | 1 |
| Desarrollo Backend | <ul style="list-style-type: none">API con endpoint GET para productos.Implementación de persistencia de datos. | <ul style="list-style-type: none">Diseño de la arquitectura de la API y modelos de datos | 1, 2 |
| Desarrollo Frontend | <ul style="list-style-type: none">Página de comparación funcional con datos dummy.Integración del frontend con la API backend.Implementación de diseño responsivo y mejora de UX | <ul style="list-style-type: none">API de Backend | 2, 3 |
| Pruebas y Despliegue | <ul style="list-style-type: none">Pruebas de integración y end-to-end.Despliegue en entorno de staging.Feedback de stakeholders y ajustes finales. | <ul style="list-style-type: none">Backend y frontend funcionandoInfra y pipelinesDisponibilidad de stakeholders | 3, 4 |

Consideraciones adicionales

Métricas y Monitoreo: Definir desde el inicio métricas clave (ej. tiempo de carga de página, errores de API, uso de funcionalidades de comparación) y herramientas de monitoreo (ej. Prometheus, Grafana, New Relic).

Escalabilidad: Considerar el diseño de la API y la base de datos para manejar un volumen masivo de productos y usuarios.

Seguridad: Análisis de vulnerabilidades, validación de inputs y protección contra ataques comunes.

CI/CD: Automatizar el proceso de build, test y despliegue para asegurar entregas rápidas y confiables.

Observabilidad: Logging y tracing distribuido para facilitar la depuración y el monitoreo en producción.

Feedback Loop: Establecer mecanismos para recolectar feedback de usuarios y usarlo para futuras iteraciones del producto.



¡Gracias!