

# Prueba Técnica - Desarrollador Fullstack SR

## Descripción del Proyecto

Una gran empresa inmobiliaria requiere la creación de una API para obtener información sobre propiedades almacenadas en una base de datos, y una página web para mostrar esta información. El objetivo es crear una aplicación full-stack utilizando .NET, MongoDB, C# y ReactJS o Next.js. El proyecto consistirá en construir un conjunto de servicios para gestionar los datos de las propiedades y crear una interfaz de usuario para mostrarlos.

## Tecnologías Requeridas

- Backend: .NET 8 o 9, C#
- Base de Datos: MongoDB
- Frontend: ReactJS o Next.js
- Pruebas: NUnit para pruebas unitarias

## Desglose de Tareas

### 1. Desarrollo del Backend (API)

- Crear una API en C# usando .NET 8 o 9 para obtener los datos de las propiedades desde la base de datos MongoDB.
- Implementar filtros en la API para obtener una lista de propiedades basada en parámetros como el nombre, la dirección y el rango de precio.
- Definir los campos de los DTOs, como:
  - IdPropietario
  - Nombre
  - Dirección de la propiedad
  - Precio de la propiedad
  - Una sola imagen

## 2. Desarrollo del Frontend (Página Web)

- Crear una página web usando ReactJS o Next.js para mostrar los datos de las propiedades.
- La página debe incluir:
  - Una lista de propiedades obtenidas de la API.
  - Filtros para buscar propiedades (nombre, dirección y rango de precio).
  - Opción para ver más detalles de propiedades individuales.
- Asegurar que el frontend sea responsivo, proporcionando una experiencia de usuario fluida en diferentes dispositivos.

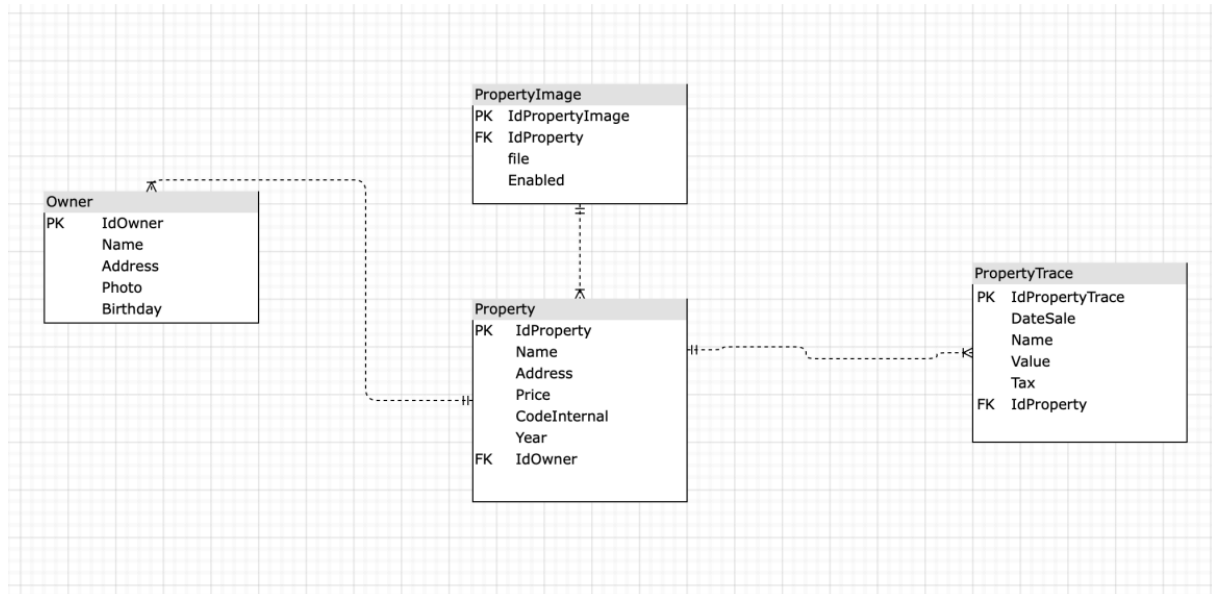
## Criterios de Evaluación

- **Arquitectura del Backend y Frontend:** Implementar una arquitectura limpia y eficiente tanto para la API del backend como para la página web del frontend.
- **Estructura del Código:** Organizar el código de manera modular y mantenible.
- **Documentación:** Proporcionar documentación clara y concisa tanto para la API como para el código del frontend.
- **Mejores Prácticas:** Seguir las mejores prácticas en el desarrollo tanto del backend como del frontend, incluyendo:
  - Arquitectura limpia
  - Manejo adecuado de errores
  - Consultas optimizadas a la base de datos (si es necesario)
- **Rendimiento:** Asegurarse de que tanto la API como el frontend estén optimizados para el rendimiento, especialmente para manejar grandes conjuntos de datos o múltiples filtros.
- **Pruebas Unitarias:** Escribir pruebas unitarias tanto para la API del backend como para los componentes del frontend utilizando NUnit u otros marcos de pruebas apropiados.
- **Código Limpio:** Asegurarse de que el código sea legible, mantenible y siga las convenciones de codificación establecidas.

## Entrega

Al completar este proyecto, demostrarás tu capacidad para construir una aplicación full-stack con una arquitectura limpia, buen rendimiento y pruebas y documentación adecuadas.

Esta versión organiza los requisitos de forma más estructurada y aclara las tareas y los criterios de evaluación.



Envía la solución con el proyecto en un archivo zip o súbelo a GitHub u otro sitio similar. Adjunta una copia de seguridad de la base de datos. Si es necesario, especifica los pasos para ejecutar el proyecto después de descargarlo.