

Prueba Técnica para Desarrollador Fullstack - Croper

Objetivo:

Esta prueba técnica busca evaluar claramente tus habilidades técnicas, capacidad de razonamiento lógico y resolución de problemas, así como tu habilidad para escribir código de forma independiente, evitando depender exclusivamente de soluciones generadas por inteligencia artificial.

Ejercicio 1: Implementación Backend (NestJS y MongoDB)

Tarea:

Desarrolla una API REST básica utilizando NestJS y MongoDB para gestionar un catálogo simple de productos con los siguientes campos:

- id
- nombre (obligatorio)
- descripción
- precio (debe ser mayor que 0)
- categoría

Requisitos:

- Implementar operaciones CRUD completas (Crear, Leer, Actualizar, Eliminar).
- Incluir validaciones básicas en los datos de entrada.
- Documentar claramente tu API utilizando Swagger/OpenAPI.
- Proteger los endpoints con autenticación JWT.

Ejercicio 2: Implementación Frontend (Angular/React/Vue)

Tarea:

Crea una aplicación frontend que interactúe con la API desarrollada en el Ejercicio 1.

Requisitos:

- Permitir listar, crear, editar y eliminar productos.
- Mostrar los productos en una tabla con paginación.
- Manejar errores de la API mostrando mensajes amigables al usuario.

- Implementar gestión del estado (NgRx para Angular, Redux para React, o Pinia/Vuex para Vue).
- Asegurar que la interfaz sea responsive y adaptativa a dispositivos móviles.

Ejercicio 3: Razonamiento y Diseño Arquitectónico

Tarea:

Sin necesidad de implementarlo, detalla claramente y justifica tus decisiones al diseñar una arquitectura de microservicios para una plataforma de comercio electrónico agropecuaria.

Tu explicación debe incluir:

- Los microservicios que crearías y sus responsabilidades principales.
- Los métodos de comunicación entre microservicios (por ejemplo: APIs REST, brokers de mensajes como Kafka o RabbitMQ, o gRPC) y por qué los elegirías.
- Estrategias específicas para garantizar comunicaciones seguras entre servicios.

Nota: Entrega un documento claro y estructurado. Esta sección debe reflejar claramente tu razonamiento personal y tu proceso de toma de decisiones.

Ejercicio 4: Optimización del Rendimiento Web (Respuesta escrita)

Escenario: Un cliente reporta que la aplicación carga lentamente en dispositivos móviles.

Tarea:

Detalla claramente al menos tres técnicas concretas que aplicarías para mejorar el rendimiento y la experiencia de usuario (UX/UI) en una aplicación web usando el framework frontend que elegiste (Angular/React/Vue).

Nota: Tu respuesta debe demostrar conocimientos específicos sobre optimización frontend.

Ejercicio 5: Seguridad y Desarrollo en la Nube (Respuestas breves)

Responde brevemente y con claridad las siguientes preguntas:

1. ¿Cuáles son las diferencias fundamentales entre OAuth y JWT?

2. ¿Cómo asegurarías una API desplegada en Azure contra ataques de fuerza bruta?
3. ¿Qué buenas prácticas de seguridad seguirías al manejar datos sensibles en MongoDB?
4. Describe brevemente cómo gestionarías el despliegue en la nube de una aplicación basada en microservicios.

Entregables:

- Todo el código fuente debe enviarse mediante un repositorio GitHub con commits claramente estructurados (no adjuntar archivos ZIP).
- Incluye un documento PDF con tus respuestas escritas y razonamientos.

Criterios de Evaluación:

- Calidad, claridad y estructura del código.
- Correcta implementación del Backend y Frontend.
- Razonamiento independiente y bien estructurado en decisiones arquitectónicas.
- Soluciones prácticas y profundas en rendimiento y seguridad.
- Respuestas escritas claras, concisas y técnicamente sólidas.

Nota Importante:

Las soluciones que dependan excesivamente de código generado por inteligencia artificial sin mostrar claramente entendimiento propio o razonamiento personal reducirán considerablemente tu calificación. Demuestra claramente tu entendimiento técnico y habilidades independientes de resolución de problemas.

¡Éxitos!