

# Full-stack Developer

## Prueba Técnica

### Prueba Técnica: Full-stack Developer - Plataforma Educativa

#### Introducción

Esta prueba técnica está diseñada para evaluar sus conocimientos y habilidades como Full-stack Developer, con énfasis en el desarrollo de aplicaciones educativas. Se evaluarán competencias tanto en front-end como en back-end, arquitectura de software, y mejores prácticas de desarrollo.

**\*\*Tiempo estimado\*\*:** 60 minutos

**\*\*Formato\*\*:** Desarrollo práctico y preguntas conceptuales

#### Parte 1: Preguntas Conceptuales (15 minutos)

Explique brevemente cómo implementaría un sistema de autenticación seguro para una plataforma educativa, considerando diferentes roles (estudiantes, profesores, administradores).

Describa la arquitectura que utilizaría para implementar un sistema de videoconferencias en tiempo real para clases virtuales. ¿Qué tecnologías específicas elegiría y por qué?

¿Cómo abordaría el almacenamiento y la gestión eficiente de materiales educativos (videos, documentos, presentaciones) considerando escalabilidad y rendimiento?

#### Parte 2: Ejercicio Práctico (45 minutos)

##### Descripción

Desarrolle un prototipo básico de un módulo de gestión de cursos que incluya:

#### Frontend (React/Vue/Angular)

Crear una interfaz simple que muestre:

- Lista de cursos
- Formulario para crear/editar un curso
- Vista detallada de un curso

#### Backend (Node.js/Python/Java)

Implementar una API REST con los siguientes endpoints:

- GET /api/courses - Listar cursos
- POST /api/courses - Crear curso
- GET /api/courses/{id} - Obtener detalles de un curso

## Requisitos Técnicos

Utilizar una arquitectura modular y escalable

Implementar validación de datos

Incluir manejo básico de errores

Documentar el código

Incluir al menos un test unitario

## Entregables

- Código fuente del proyecto
- README con instrucciones de instalación y ejecución
- Documentación básica de la API
- Explicación breve de decisiones técnicas tomadas

## Criterios de Evaluación

### Arquitectura y Diseño (30%)

- Estructura del código
- Patrones de diseño utilizados
- Escalabilidad de la solución

### Calidad del Código (30%)

- Legibilidad

- Buenas prácticas
- Manejo de errores
- Documentación

## **Funcionalidad (25%)**

- Cumplimiento de requisitos
- Funcionamiento correcto
- Performance

## **Pruebas (15%)**

- Calidad de los tests implementados
- Cobertura de casos de uso principales

## **Notas Adicionales**

- Puede utilizar cualquier framework o librería de su preferencia
- Se valorará el uso de TypeScript
- La implementación de características adicionales relevantes suma puntos extra
- Enfóquese en la calidad sobre la cantidad de características implementadas

## **Entrega**

- Suba el código a un repositorio Git (GitHub/GitLab)
- Incluya todos los entregables solicitados
- Asegúrese de que el proyecto pueda ejecutarse siguiendo las instrucciones del README

---

Esta prueba evalúa no solo sus habilidades técnicas sino también su capacidad para:

- Tomar decisiones de diseño
- Escribir código mantenible
- Documentar adecuadamente
- Manejar requisitos en tiempo limitado
- Implementar soluciones escalables