

Prueba Técnica – Desarrollador Web Senior Backend

Objetivo

Evaluar las **habilidades técnicas**, de **diseño** y **de resolución de problemas** de candidatos para la vacante de Desarrollador Web Senior Backend, alineado con los requerimientos del área de Desarrollo en Toka.

1. Diseño y Arquitectura de Software

Objetivo: Validar conocimientos en patrones de diseño, arquitecturas modernas y buenas prácticas de diseño orientado a objetos.

Instrucción:

Diseña la arquitectura para un sistema de gestión de usuarios que permita autenticación, autorización, auditoría de accesos y gestión de roles. El sistema debe ser escalable y estar basado en microservicios.

Requisitos de entrega:

- Diagrama de arquitectura (formato visual o Markdown)
- Justificación de decisiones técnicas
- Breve explicación del flujo de datos

Evaluación:

- Claridad y cohesión arquitectónica
- Uso de patrones adecuados
- Consideración de seguridad y escalabilidad

2. Desarrollo Backend – API RESTful

Objetivo: Comprobar habilidades prácticas en desarrollo con C#, Python o Go y uso de ORM.

Instrucción:

Implementa una API RESTful para el módulo de “Gestión de Tareas” con operaciones CRUD y autenticación básica.

Tecnologías requeridas:

- Lenguaje libre: C# (.NET), Python (FastAPI/Flask) o Go
- ORM: Entity Framework, SQLAlchemy o equivalente
- Base de datos: SQL Server o SQLite

Evaluación:

- Cumplimiento funcional
- Uso correcto de ORM
- Código limpio y seguro

3. Revisión de Código (Code Review)

Objetivo: Medir criterio técnico y capacidad crítica sobre buenas prácticas.

Instrucción:

Se proporciona un fragmento de código (en C#, Python y Go) con errores de diseño, código repetido y posibles fallos. El candidato debe:

- Detectar errores o malas prácticas
- Sugerir y explicar mejoras

Evaluación:

- Entrega de análisis técnico
- Profundidad de observaciones
- Claridad en justificación

4. Desempeño bajo presión – Resolución de problema

Objetivo: Medir capacidad de tomar decisiones rápidas y resolver problemas.

Instrucción:

Simular una falla en producción: “Los usuarios no pueden guardar registros, la API responde 500”. El candidato debe:

- Formular hipótesis
- Describir un plan de diagnóstico
- Explicar cómo comunicaría el avance

Evaluación:

- Lógica del análisis
- Prioridad de acciones
- Comunicación clara

Duración Total Estimada

- Arquitectura: 60 min
- Desarrollo API: 90 min
- Code Review: 30 min
- Diagnóstico: 30 min

Total: 3.5 horas

Entregables Esperados

- Repositorio o carpeta comprimida con:
 - Código fuente y documentación
 - Diagrama(s) y justificación técnica
 - Documento de análisis y revisión de código
 - Respuesta del ejercicio de diagnóstico