

Guía Completa de Preguntas y Respuestas para Entrevista Técnica

1. Lógica de programación

Escribe una función que invierta una cadena.

```
```javascript
function invertir(str) {
 return str.split("").reverse().join("");
}
```
```

¿Cómo contarías cuántas veces aparece una letra en una palabra?

```
```python
def contar_letra(palabra, letra):
 return palabra.count(letra)
```
```

¿Cómo imprimirías los primeros n números de Fibonacci?

```
```python
def fibonacci(n):
 a, b = 0, 1
 for _ in range(n):
 print(a)
 a, b = b, a + b
```
```

¿Cómo encontrarías el número mayor de un arreglo sin usar Math.max()?

```
```javascript
let numeros = [3, 9, 1];
let max = numeros[0];
for (let n of numeros) {
 if (n > max) max = n;
}
```
```

¿Cómo verificarías si una palabra es un palíndromo?

```
```javascript
function esPalindromo(str) {
 const limpio = str.toLowerCase().replace(/\s/g, "");
 return limpio === limpio.split("").reverse().join("");
}
```
```

¿Qué estructura usarías para buscar por clave rápidamente?

En JavaScript usaría un ``Object`` o ``Map``, en Python usaría un ``dict`` (diccionario).

2. Frontend: JavaScript, TypeScript y React

¿Qué diferencia hay entre *var*, *let* y *const*?

`var` tiene alcance de función y permite hoisting. `let` y `const` tienen alcance de bloque. `const` no permite reasignar.

¿Qué es *hoisting*?

Es el comportamiento por el cual las declaraciones (`var`, `function`) se mueven al inicio del contexto antes de ejecutarse.

¿Qué es el *Virtual DOM*?

Es una copia en memoria del DOM real que React usa para detectar y aplicar cambios de forma eficiente.

¿Qué es *useState* y *useEffect* y cuándo se usan?

`useState` se usa para manejar estado, `useEffect` para ejecutar efectos secundarios como llamadas a APIs.

¿Cómo renderizarías una lista en React?

```
```jsx
{lista.map((item, i) => {item})}
```
```

¿Qué son *props* y en qué se diferencian del *state*?

Las `props` son valores que recibe un componente, el `state` es interno del componente y puede cambiar.

¿Qué es un *componente controlado*?

Un componente de formulario cuyo valor es controlado por el estado del componente de React.

¿Cómo gestionarías formularios en React?

Usando `useState` para los valores y `onChange` para actualizarlos.

3. Backend: FastAPI o Node.js

¿Qué es una API REST?

Es una interfaz que permite comunicación entre sistemas usando HTTP con métodos como GET, POST, PUT, DELETE.

¿Qué métodos HTTP conoces y para qué se usan?

GET (leer), POST (crear), PUT/PATCH (actualizar), DELETE (eliminar).

¿Cómo defines un endpoint en FastAPI?

```
```python
@app.get("/ruta")
def funcion():
 return {"clave": "valor"}
```
```

¿Cómo recibirías datos desde el cuerpo (POST)?

```
```python
from pydantic import BaseModel
class Item(BaseModel):
 nombre: str
@app.post("/items")
def crear(item: Item):
 return item
```
```

¿Cómo harías manejo de errores en un endpoint?

Usando `try/except` o validaciones con Pydantic.

¿Qué es un middleware?

Una función que se ejecuta antes o después de las rutas para modificar la solicitud o respuesta.

¿Qué diferencias hay entre JSON y un diccionario en Python?

JSON es una cadena de texto estructurada, un diccionario es una estructura de datos en memoria.

4. SQL (tabla Empleados)

¿Cómo seleccionar todos los empleados activos?

```
```sql
SELECT * FROM Empleados WHERE EstaActivo = 1;
```
```

¿Cómo traer empleados que ingresaron en el último año?

```
```sql
SELECT * FROM Empleados WHERE FechaIngreso >= DATEADD(YEAR, -1, GETDATE());
```
```

¿Cómo contar cuántos empleados hay por cargo?

```
```sql
SELECT Cargo, COUNT(*) FROM Empleados GROUP BY Cargo;
```
```

¿Cómo harías un JOIN entre Empleados y Tiendas?

```
```sql
SELECT e.Nombre, t.NombreTienda FROM Empleados e
JOIN Tiendas t ON e.TiendaId = t.Id;
```
```

¿Cómo filtrar empleados por correo corporativo?

```
```sql
SELECT * FROM Empleados WHERE Correo LIKE '%@empresa.com';
```
```

¿Qué hace GROUP BY? ¿Y ORDER BY?

`GROUP BY` agrupa registros, `ORDER BY` los ordena.

5. Docker / DevOps

¿Qué es Docker?

Permite crear contenedores que empaquetan tu app y sus dependencias de forma aislada.

¿Qué diferencia hay entre un contenedor y una máquina virtual?

Un contenedor es más ligero y comparte el sistema operativo con el host, mientras que una VM emula un sistema completo.

¿Has trabajado con imágenes de Docker?

Sí, una imagen es una plantilla de solo lectura para crear contenedores. Puedo construir una imagen con un Dockerfile.

¿Para qué sirve un Dockerfile?

Para definir instrucciones para construir una imagen Docker (instalaciones, copias, comandos de inicio).

6. IA generativa

¿Qué sabes de la IA generativa?

Son modelos como ChatGPT que generan texto, código u otros contenidos a partir de entradas de lenguaje natural.

¿Has usado ChatGPT o APIs de OpenAI?

Sí, he usado ChatGPT y conozco la API de OpenAI para generar respuestas automatizadas o asistentes virtuales.

¿Qué es LangChain o n8n?

LangChain permite crear apps que usan LLMs con cadenas de pasos. n8n es una herramienta para automatizar flujos entre servicios.

¿En qué casos usarías una IA en una app real?

Para asistentes virtuales, generación de respuestas, clasificación de texto, resúmenes, etc.

7. Git y GitHub

¿Qué diferencia hay entre Git y GitHub?

Git es un sistema de control de versiones local. GitHub es una plataforma para alojar repositorios Git y colaborar en línea.

¿Qué hace git add, git commit y git push?

`add` agrega cambios al área de preparación, `commit` guarda una versión, `push` sube al repositorio remoto.

¿Qué es un branch y para qué sirve?

Es una rama paralela de desarrollo para trabajar sin afectar el código principal.

¿Qué es un pull request?

Una solicitud para fusionar cambios de una rama a otra, generalmente tras revisión de código.

¿Cómo resolverías un conflicto de merge?

Editaría manualmente el archivo, seleccionaría qué versión conservar, luego haría commit.

¿Cómo organizas tus commits?

Uso mensajes claros, relacionados a tareas o funcionalidades específicas. Ej: `fix: validación de formulario`

¿Qué proyecto destacarías de tu GitHub?

Mi sistema de gestión de empleados y tiendas con .NET y Angular, o mi app Postúlate con Laravel y React.

8. Preguntas personales / actitud

Cuéntame sobre un proyecto que hayas hecho.

Postúlate: app de gestión de ofertas laborales, con React frontend y Laravel backend. Incluye login, CRUD, PDF, y control de sesiones.

¿Qué parte disfrutaste más: frontend o backend?

Me gusta el backend por la lógica y manejo de datos, pero también disfruto crear interfaces funcionales.

¿Qué retos enfrentaste y cómo los resolviste?

Errores en autenticación JWT y control de sesión. Los resolví con pruebas, depuración y ajustes en el middleware/interceptores.

¿Qué haces si algo en producción falla?

Reviso logs, verifico errores y aplico soluciones temporales o parches mientras preparo la corrección definitiva.

¿Cómo te organizas para aprender una tecnología nueva?

Divido el tema en partes, busco documentación oficial, hago pruebas pequeñas y luego aplico en proyectos reales.

¿Qué harías si no entiendes una tarea asignada?

Pregunto de forma clara a quien la asignó, busco ejemplos similares, y empiezo por lo que sí comprendo.

9. Preguntas de cierre

¿Qué esperas aprender en esta pasantía?

Profundizar en desarrollo profesional real, metodologías ágiles, buenas prácticas, y colaboración con equipos técnicos.

¿Qué tipo de proyectos te interesan?

Aplicaciones web con frontend/backend bien estructurado, y con posibilidad de usar APIs, IA o automatización.

¿Tienes disponibilidad para trabajar en equipo remoto?

Sí, tengo buena organización, experiencia colaborando por Git y disponibilidad para reuniones virtuales.

¿Tienes preguntas sobre la empresa o el equipo?

Sí, me interesa saber cómo organizan el trabajo, qué herramientas usan día a día, y cómo es el proceso de mentoría.