



# Listado de Preguntas Técnicas

## HTML

1. **¿Cuál es la forma correcta de crear un enlace que abra en una nueva pestaña?**
  - a) `<a href="url" new="tab">`
  - b) `<a href="url" target="_blank">`
  - c) `<a href="url" open="new">`
  - d) `<a href="url" window="new">`
2. **¿Qué etiqueta semántica HTML5 se usa para contenido independiente?**
  - a) `<section>`
  - b) `<div>`
  - c) `<article>`
  - d) `<aside>`

## CSS

3. **¿Qué propiedad CSS se usa para crear un diseño de cuadrícula?** a) `display: flex`
  - b) `display: grid`
  - c) `display: table`
  - d) `display: inline-grid`
4. **¿Cuál es la especificidad más alta en CSS?**
  - a) Elemento
  - b) Clase
  - c) ID



**d) Estilos inline**

**JavaScript**

5. ¿Qué método se usa para agregar un elemento al final de un array? a)

array.add()

**b) array.push()**

c) array.append()

d) array.insert()

6. ¿Cuál es la diferencia principal entre let y const?

a) let es para números, const para strings

**b) let permite reasignación, const no**

c) const es más rápido que let

d) No hay diferencia

**React**

7. ¿Qué hook se usa para manejar efectos secundarios en React? a)

useState

b) **useEffect**

c) useContext

d) useReducer

8. ¿Cómo se pasan datos de un componente padre a hijo en React?

a) A través de state

**b) A través de props**

c) A través de context

d) A través de refs



## Vue

9. ¿Qué directiva de Vue se usa para renderizado condicional?

- a) v-show
- b) v-if
- c) v-display
- d) v-render

10. ¿Cómo se define una propiedad reactiva en Vue 3 Composition API? a)

reactive()

- b) ref()
- c) computed()
- d) Todas las anteriores pueden usarse

## Angular

11. ¿Qué decorador se usa para definir un componente en Angular?

- a) @NgModule
- b) @Injectable
- c) @Component
- d) @Directive

12. ¿Qué sintaxis se usa para enlazar propiedades en Angular? a)

{property}

- b) (property)
- c) [property]
- d) {{property}}



## Node.js

13. ¿Qué objeto global de Node.js permite acceder a variables de entorno?

- a) process.env
- b) global.env
- c) system.env
- d) node.env

14. ¿Qué módulo de Node.js se usa para trabajar con rutas de archivos? a)

fs

b) path

c) url

d) file

## Express

15. ¿Qué método de Express se usa para manejar peticiones POST? a)

app.post()

b) app.get()

c) app.send()

d) app.request()

16. ¿Cómo se define un middleware en Express que se ejecute en todas las rutas?

a) app.middleware()

b) app.use()

c) app.all()



d) `app.apply()`

## Django

17. ¿Qué comando se usa para crear migraciones en Django?

- a) `python manage.py migrate`
- b) `python manage.py makemigrations`
- c) `python manage.py createmigrations`
- d) `python manage.py db migrate`

18. ¿Qué ORM usa Django por defecto?

- a) SQLAlchemy
- b) Sequelize
- c) Django ORM
- d) TypeORM Flask

19. ¿Qué decorador se usa para definir una ruta en Flask?

- a) `@app.path()`
- b) `@app.route()`
- c) `@app.url()`
- d) `@app.endpoint()`

20. ¿Cómo se habilita el modo debug en Flask?

- a) `app.debug = True`
- b) `app.run(debug=True)`
- c) `flask.debug()`
- d) Ambas a y b son correctas



## Spring Boot

21. ¿Qué anotación se usa para marcar una clase como controlador REST en Spring Boot?

- a) @Controller
- b) @RestController**
- c) @RequestMapping
- d) @Service

22. ¿Qué archivo se usa para configurar propiedades en Spring Boot? a)

config.properties

- b) application.properties**
- c) settings.properties
- d) spring.properties

## API REST

23. ¿Qué código de estado HTTP indica que un recurso fue creado exitosamente?

- a) 200
- b) 201**
- c) 204
- d) 202

24. ¿Qué método HTTP se usa para actualizar completamente un recurso?

- a) POST
- b) PATCH
- c) PUT**
- d) UPDATE



## SQL

25. ¿Qué cláusula SQL se usa para filtrar resultados agrupados?

- a) WHERE
- b) FILTER
- c) **HAVING**
- d) GROUP BY

26. ¿Qué tipo de JOIN devuelve todas las filas de ambas tablas?

- a) INNER JOIN
- b) LEFT JOIN
- c) RIGHT JOIN
- d) **FULL OUTER JOIN**

## Ejercicio Práctico: Sistema de Gestión de Tareas

Tiempo estimado: 2 horas

### Problema

Eres parte del equipo de desarrollo de TaskFlow. Tu misión es construir un MVP simplificado que permita a los usuarios gestionar sus tareas personales de forma eficiente.

### Lo que Debes Construir

Una aplicación web full-stack donde:

1. Los usuarios puedan registrarse e iniciar sesión
2. Los usuarios puedan crear y gestionar sus tareas personales
3. Las tareas se organicen en un tablero simple con estados: "Por hacer", "En progreso", "Completada"



4. Las tareas tengan título, descripción y prioridad

## **Funcionalidades Requeridas**

### Autenticación, Usuarios y Tareas

- Registro con email y contraseña
- Login con JWT (solo access token)
- Ver perfil personal básico

### Tareas

- Crear tarea (título, descripción, prioridad: alta/media/baja)
- Listar todas mis tareas
- Cambiar estado de tarea (Por hacer → En progreso → Completada)
- Eliminar tarea
- Filtrar tareas por estado

## **Stack Tecnológico Requerido**

### **Frontend**

Elige UNA opción:

- React
- Vue
- Angular

### **Backend**

Elige UNA opción:

- Node.js + Express
- Django
- Flask





- Spring Boot

## Base de datos

Elige UNA opción:

- MySQL
- PostgreSQL
- MongoDB

## Requisitos Técnicos Obligatorios

### Backend

1. Autenticación JWT con access token
2. Middleware de autenticación en rutas protegidas
3. Hash de contraseñas con bcrypt
4. Validación básica de datos
5. Manejo de errores con códigos HTTP apropiados
6. Variables de entorno (.env)

### Frontend

1. Rutas protegidas
2. Agregar JWT en peticiones HTTP
3. Estados de carga básicos
4. Validación de formularios
5. Diseño responsive básico

## Entregables

1. **Repositorio Git** con:
  - Código fuente (frontend + backend)
  - .gitignore



- Commits descriptivos
- 2. **README.md** que incluya:
  - Descripción breve
  - Tecnologías utilizadas
  - Instrucciones de instalación
  - Variables de entorno necesarias (.env.example)
  - Comandos para ejecutar
- 3. **Documentación de API** (formato libre):
  - Lista de endpoints principales
  - Ejemplos básicos de request/response
- 4. **Datos de prueba:**
  - Al menos 1 usuario y 5 tareas de ejemplo