

GUÍA COMPLETA PARA PROYECTO INTEGRADOR DE DESARROLLO WEB

(El contenido anterior se mantiene igual, y añadimos la siguiente sección al final)

BANCO DE PREGUNTAS PARA EVALUACIÓN TÉCNICA

BLOQUE 1: FUNDAMENTOS DE JAVASCRIPT

Nivel Básico

1. ¿Cuál es la diferencia entre `var`, `let` y `const`?
2. Explica qué son los tipos de datos primitivos en JavaScript
3. ¿Cómo funcionan los operadores lógicos `&&`, `||` y `!`?
4. ¿Qué es una función flecha y cómo difiere de una función tradicional?
5. Describe el concepto de scope en JavaScript

Nivel Intermedio

1. Explica el concepto de closures y proporciona un ejemplo
2. ¿Qué son las promesas y cómo funcionan?
3. Describe el funcionamiento de `async/await`
4. ¿Cómo funcionan los métodos de array como `map()`, `filter()` y `reduce()`?
5. Explica la diferencia entre `==` y `===`

Nivel Avanzado

1. Describe el Event Loop de JavaScript
2. Explica cómo funciona la programación asíncrona en JavaScript
3. ¿Qué es el prototipo en JavaScript y cómo funciona la herencia?
4. Describe las diferencias entre `call`, `apply` y `bind`
5. Explica el concepto de memoización

BLOQUE 2: VUE.JS

Nivel Básico

1. ¿Qué es Vue.js y cuáles son sus principales características?
2. Explica qué es la reactividad en Vue
3. ¿Cuál es la diferencia entre `v-if` y `v-show`?

4. ¿Cómo se declaran componentes en Vue?
5. ¿Qué son las props en Vue?

Nivel Intermedio

1. Explica la diferencia entre Options API y Composition API
2. ¿Cómo se comunican los componentes en Vue?
3. Describe el ciclo de vida de un componente Vue
4. ¿Qué son los computed properties y cómo difieren de los métodos?
5. Explica cómo funcionan los watchers en Vue

Nivel Avanzado

1. Describe el sistema de reactividad de Vue en profundidad
2. ¿Cómo implementarías un estado global en Vue?
3. Explica cómo funciona la compilación de plantillas en Vue
4. ¿Qué son los mixins y cuáles son sus limitaciones?
5. Describe las estrategias de optimización de rendimiento en Vue

BLOQUE 3: AG GRID

Nivel Básico

1. ¿Qué es AG Grid y para qué se utiliza?
2. ¿Cómo se define una columna en AG Grid?
3. ¿Qué es un cellRenderer?
4. ¿Cómo se implementa la paginación en AG Grid?
5. ¿Qué tipos de filtros ofrece AG Grid?

Nivel Intermedio

1. Explica cómo personalizar el renderizado de celdas
2. ¿Cómo se manejan eventos en AG Grid?
3. Describe las estrategias de ordenación en AG Grid
4. ¿Cómo se implementa un filtro personalizado?
5. Explica las diferencias entre las versiones de AG Grid (Community vs Enterprise)

Nivel Avanzado

1. Describe estrategias de optimización de rendimiento en tablas grandes
2. ¿Cómo implementarías un filtro complejo que combine múltiples condiciones?

3. Explica las técnicas de virtualización en AG Grid
4. ¿Cómo manejarías la carga de datos asíncronos en AG Grid?
5. Describe las estrategias de exportación de datos

BLOQUE 4: TAILWIND CSS

Nivel Básico

1. ¿Qué es Tailwind CSS?
2. ¿Cómo se aplican clases en Tailwind?
3. ¿Qué son las clases utilitarias?
4. ¿Cómo se implementa un diseño responsive con Tailwind?
5. ¿Qué es el prefijo `@apply` en Tailwind?

Nivel Intermedio

1. Explica cómo personalizar el archivo de configuración de Tailwind
2. ¿Cómo se crean componentes reutilizables con Tailwind?
3. Describe las estrategias de diseño responsive
4. ¿Cómo se implementa un tema oscuro con Tailwind?
5. Explica el concepto de purge en Tailwind

Nivel Avanzado

1. ¿Cómo implementarías un sistema de diseño personalizado con Tailwind?
2. Describe las estrategias de optimización de CSS con Tailwind
3. ¿Cómo manejarías casos complejos de diseño que requieren múltiples breakpoints?
4. Explica las técnicas de extensión de Tailwind
5. ¿Cómo implementarías animaciones personalizadas con Tailwind?

BLOQUE 5: DESARROLLO WEB FULL-STACK

Desarrollo de Proyecto

1. Explica la arquitectura general de tu proyecto
2. ¿Cómo manejas el estado global en tu aplicación?
3. Describe tu estrategia de consumo de API
4. ¿Cómo implementaste el manejo de errores?
5. Explica las decisiones de diseño más importantes que tomaste

Optimización y Mejores Prácticas

1. ¿Qué técnicas de optimización de rendimiento implementaste?
2. Describe tu estrategia de manejo de estados de carga
3. ¿Cómo ensures la accesibilidad de tu aplicación?
4. Explica las estrategias de validación de datos
5. ¿Cómo manejas la seguridad en tu aplicación?

CONSEJOS PARA RESPONDER PREGUNTAS

1. **Sé Específico:** Proporciona ejemplos de código cuando sea posible
2. **Explica el Por Qué:** No solo describas QUÉ hiciste, sino POR QUÉ lo hiciste
3. **Muestra Comprensión:** Demuestra que entiendes los conceptos detrás del código
4. **Sé Honesto:** Si no sabes algo, admítelo y explica cómo lo investigarías
5. **Mantén la Calma:** Las preguntas buscan evaluar tu comprensión, no ponerte a prueba

ESTRATEGIAS DE PREPARACIÓN

1. **Práctica de Código**
 - Implementa los conceptos en proyectos pequeños
 - Experimenta con diferentes enfoques
 - Construye y reconstruye
2. **Estudio Teórico**
 - Lee documentación oficial
 - Mira tutoriales en profundidad
 - Participa en comunidades de desarrollo
3. **Simulación de Entrevista**
 - Practica explicar conceptos en voz alta
 - Grábate respondiendo preguntas
 - Pide retroalimentación a compañeros o mentores

RECUERDA: El objetivo no es memorizar respuestas, sino comprender profundamente los conceptos y ser capaz de aplicarlos de manera flexible y creativa.

¡Éxito en tu proyecto y presentación!