

PREGUNTAS DE CONOCIMIENTO

1. Fundamentos de Programación:

o Preguntas sobre tipos de datos, estructuras de control (condicionales, bucles) y funciones.
¿Cuáles conoce?

Respuesta: enteros, decimales, booleanos, cadenas de texto, estructuras de control como condicionales (if, else, switch) que permiten tomar decisiones según una condición, y bucles (for, while, do while, foreach).

2. Control de Versionamiento: o Explicar la importancia de Git, cómo realizar commits, branches, merges y resolver conflictos. Cual es el comando para crear una rama ?

Respuesta: Git es un sistema de control de versiones fundamental para el desarrollo de software, ya que permite llevar un historial de cambios, trabajar en equipo sin conflictos. Los commits se utilizan para guardar cambios de forma controlada con mensajes descriptivos, las ramas permiten desarrollar nuevas funcionalidades sin afectar la rama principal (main o master), los merges sirven para integrar cambios entre ramas y los conflictos se resuelven revisando y decidiendo qué código conservar. El comando para crear una rama es `git branch nombre-rama` o `git checkout -b nombre-rama` para crearla y moverse a ella directamente.

3. React:

o ¿Qué es un hook en React y cuál es el hook más utilizado?

o Preguntar sobre el uso de Redux y su propósito en la gestión del estado.

o ¿Qué es React Router y cómo se utiliza para la navegación?

Respuesta: Un hook en React es una función especial que permite usar estado y funcionalidades del ciclo de vida dentro de componentes funcionales, siendo `useState` el hook más utilizado para manejar el estado de un componente. Redux es una librería que se usa para la gestión global del estado cuando la aplicación es más compleja y muchos componentes necesitan compartir información de forma predecible. React Router, por su parte, se utiliza para manejar la navegación entre diferentes vistas de la aplicación sin recargar la página, definiendo rutas como `/login` o `/home` mediante componentes como `BrowserRouter`, `Routes` y `Route`.

4. CQRS:

o Explica qué es CQRS y sus beneficios en arquitecturas de software.

Respuesta: CQRS, que significa Command Query Responsibility Segregation, es un patrón de arquitectura que separa las operaciones de escritura (commands) de las operaciones de lectura (queries), permitiendo que cada una tenga su propia lógica y modelo.

5. MediatR:

o ¿Qué es MediatR y cómo facilita la comunicación entre componentes?

Respuesta: MediatR es una librería utilizada en aplicaciones .NET que implementa el patrón Mediator y facilita la comunicación entre componentes sin que estos estén directamente acoplados entre sí, permitiendo enviar comandos o consultas a través de un mediador.

6. FluentValidation:

o Define FluentValidation y cómo se usa para validar modelos en aplicaciones .NET.

Respuesta: FluentValidation es una librería para .NET que permite validar modelos de forma clara, legible y desacoplada de las entidades, mediante la definición de reglas como campos obligatorios, longitudes mínimas, formatos de correo y validaciones personalizadas.