**Resumen de la Clase: Operador Condicional Ternario y Versión Corta en JavaScript**

**Introducción**

En esta clase, aprendimos sobre el **operador condicional ternario** y una forma más simplificada de usarlo. El profesor explicó cómo el ternario puede reemplazar las estructuras if/else tradicionales para realizar asignaciones rápidas, y cómo usar el operador && para manejar condiciones específicas.

**Código y Explicaciones**

**1. Estructura Tradicional con if/else**

javascript

Copiar código

const activo = true; // Variable de condición

let mensaje; // Declaración de la variable mensaje

if (activo) {

mensaje = "Activo"; // Si activo es true

} else {

mensaje = "Inactivo"; // Si activo es false

}

console.log(mensaje); // Resultado según la condición

**Explicación:**  
Aquí se utiliza if/else para asignar un valor a mensaje dependiendo del valor de activo.

**2. Simplificación con el Operador Ternario**

javascript

Copiar código

const activo = true;

const mensaje = activo ? "Activo" : "Inactivo";

console.log(mensaje);

**Explicación:**  
El operador ternario (condición ? valor\_si\_true : valor\_si\_false) reemplaza el if/else.

* Si activo es true, mensaje será "Activo".
* Si activo es false, mensaje será "Inactivo".

**3. Ternario con Caso Nulo (Sin Else)**

javascript

Copiar código

const activo = false;

const mensaje = activo ? "Activo" : null;

console.log(mensaje); // Imprime null si activo es false

**Explicación:**  
Si solo se requiere ejecutar algo cuando la condición es true, el else puede devolver null para indicar que no se hace nada.

**4. Uso del Operador && como Condicional Reducido**

javascript

Copiar código

const activo = true;

const mensaje = activo && "Activo";

console.log(mensaje); // "Activo" si activo es true, undefined si es false

**Explicación:**  
El operador lógico && devuelve el valor a la derecha solo si la condición a la izquierda es true. Esto es útil cuando no necesitamos manejar el caso false.

**Conclusión**

El operador condicional ternario y el uso de && permiten escribir código más conciso y legible, especialmente en aplicaciones como **React**. Estas herramientas serán de gran utilidad para manejar condiciones de forma rápida y efectiva.