

PROGRAMA DE ESTUDIOS

DESARROLLOS DE SISTEMAS DE INFORMACIÓN

**DESARROLLO FRONTEND
DE SISTEMAS DE WEB**

Tema

**MANEJO DE ERRORES EN
JAVASCRIPT: TRY-CATCH**

Manejo de errores en JavaScript: try-catch

Actividad: Presentación de un código con error sin manejo de excepciones

Se presentará un código con un error que no ha sido manejado y se observará su impacto en la ejecución del programa.

❖ Ejemplo de código sin manejo de errores:

javascript

```
console.log("Inicio del programa");
let resultado = dividir(10, 0); // Esta función no existe
console.log("Resultado:", resultado); // Nunca se ejecutará debido al
// error
console.log("Fin del programa");
```

◆ Problema:

- JavaScript generará un error (`Uncaught ReferenceError: dividir is not defined`), lo que detendrá la ejecución del programa.

Pregunta detonadora:

"*¿Qué sucede cuando un error no es manejado correctamente en JavaScript?*"

2. Breve introducción a la necesidad de gestionar errores

En aplicaciones web, los errores pueden deberse a:

- ✓ Datos de entrada incorrectos.
- ✓ Funciones que no existen o no están definidas.

- ✓ Problemas con conexiones a servidores o bases de datos.
- ✓ Errores en operaciones matemáticas o de manipulación del DOM.

Si no se manejan, los errores pueden:

- ⚠ Detener la ejecución del código.
- ⚠ Afectar la experiencia del usuario.
- ⚠ Mostrar información sensible en la consola del navegador.

3. Presentación teórica (20 min)

Tipos de errores en JavaScript

- **Errores sintácticos:**
- Se producen cuando el código tiene una sintaxis incorrecta.

☞ Ejemplo:

```
javascript
console.log("Hola" // Falta el paréntesis de cierre
```

- Mensaje de error: Uncaught SyntaxError: Unexpected end of input.

- **Errores de ejecución (runtime errors):**

- Ocurren cuando se ejecuta el código y se intenta realizar una operación inválida.

☞ Ejemplo:

```
javascript
let x = y + 10; // y no está definida
```

- Mensaje de error: Uncaught ReferenceError: y is not defined.

- **Errores lógicos:**

- No generan un error en la consola, pero producen resultados incorrectos.

☞ Ejemplo:

```
javascript
function sumar(a, b) {
    return a - b; // Error lógico, debería ser a + b
```

}

Estructura y uso de try...catch en JavaScript

JavaScript ofrece try...catch para capturar y manejar errores sin interrumpir el flujo del programa.

📌 Sintaxis de try...catch:

```
javascript
try {
    // Código que puede generar un error
} catch (error) {
    // Código que maneja el error
}
```

📌 Ejemplo con try...catch:

```
javascript

try {
    let resultado = dividir(10, 0); // Función no definida
    console.log("Resultado:", resultado);
} catch (error) {
    console.log("Ocurrió un error:", error.message);
}
```

Uso de finally y su importancia

El bloque finally se ejecuta siempre, ya sea que ocurra un error o no.

❖ Ejemplo:

javascript

```
try {  
    let resultado = 10 / 0;  
} catch (error) {  
    console.log("Error detectado");  
} finally {  
    console.log("Este mensaje se muestra siempre");  
}
```

Objeto Error y cómo obtener detalles del error

JavaScript proporciona el objeto Error para manejar errores con más detalle.

❖ Ejemplo:

```
javascript  
try {  
    throw new Error("Este es un error personalizado");  
} catch (error) {  
    console.log("Mensaje de error:", error.message);  
    console.log("Stack trace:", error.stack);  
}
```

4. Demostración práctica (20 min)

Implementación de un bloque try...catch para manejar errores en una función

javascript

```
function dividir(a, b) {  
    try {  
        if (b === 0) throw new Error("No se puede dividir entre  
cero");  
        return a / b;  
    } catch (error) {  
        console.error("Error:", error.message);  
    }  
}  
  
console.log(dividir(10, 0));
```

Uso de `console.error()` para registrar errores en la consola

💡 Ejemplo:

```
javascript  
try {  
    let obj = null;  
    console.log(obj.propiedad); // Esto generará un error  
} catch (error) {  
    console.error("Se produjo un error:", error.message);  
}
```

Uso de `throw new Error("Mensaje de error")` para generar errores personalizados

```
javascript
```

```
function validarEdad(edad) {  
    if (edad < 18) {  
        throw new Error("Debes ser mayor de edad");  
    }  
}
```

```
        return "Acceso concedido";  
    }  
  
try {  
    console.log(validarEdad(15));  
} catch (error) {  
    console.log("Error:", error.message);  
}
```

Manejo de errores en funciones asíncronas con try...catch y await

💡 Ejemplo:

```
javascript  
async function obtenerDatos() {  
    try {  
        let respuesta = await fetch("https://api.ejemplo.com/datos");  
        if (!respuesta.ok) {  
            throw new Error("No se pudo obtener los datos");  
        }  
        let datos = await respuesta.json();  
        console.log(datos);  
    } catch (error) {  
        console.error("Error en la solicitud:", error.message);  
    }  
}  
obtenerDatos();
```

5. Actividad práctica individual (20 min)

1. Corregir un código con errores utilizando try...catch

Se proporcionará el siguiente código con errores para que los estudiantes lo corrijan.

❖ Código con errores:

```
javascript
function calcularArea(base, altura) {
    return base * altura;
}
console.log(calcularArea(5)); // Falta el segundo parámetro
```

❖ Corrección con try...catch:

```
javascript
function calcularArea(base, altura) {
    try {
        if (base === undefined || altura === undefined) {
            throw new Error("Ambos parámetros son obligatorios");
        }
        return base * altura;
    } catch (error) {
        console.error("Error:", error.message);
    }
}
console.log(calcularArea(5));
```

2. Implementar una función que valide la entrada del usuario y lance un error si es inválida

❖ Ejemplo:

```
javascript
function validarNombre(nombre) {
    if (nombre.trim() === "") {
        throw new Error("El nombre no puede estar vacío");
    }
    return `Bienvenido, ${nombre}`;
}
```

```
}

try {
    console.log(validarNombre(""));
} catch (error) {
    console.log("Error:", error.message);
}
```

3. Escribir un script que capture errores en una solicitud fetch fallida y muestre un mensaje de error al usuario

⭐ Ejemplo:

```
javascript
async function obtenerUsuarios() {
    try {
        let respuesta = await
fetch("https://api.invalida.com/usuarios");
        if (!respuesta.ok) {
            throw new Error("No se pudieron obtener los datos");
        }
        let datos = await respuesta.json();
        console.log(datos);
    } catch (error) {
        document.body.innerHTML = `<p style="color:red;">Error:
${error.message}</p>`;
    }
}
obtenerUsuarios();
```

