



SOCIEDAD DE CAPACITACIÓN TECNOLÓGICA DE CHILE  
OTEC SOCATECH

# Glosario de Términos

## Bootcamp Full Stack JavaScript

### 1. Fundamentos de JavaScript

#### JavaScript y Node.js

- **JavaScript:** Lenguaje de programación interpretado utilizado para desarrollar aplicaciones web interactivas.
- **Node.js:** Entorno de ejecución de JavaScript en el servidor que permite construir aplicaciones escalables.

#### Datos Primitivos y Estructuras de Control

- **Tipos de datos primitivos:** Incluyen string, number, boolean, null, undefined, symbol y bigint.
- **Estructuras de control:** Instrucciones como if-else, switch, for, while que controlan el flujo de ejecución del programa
- **Operador ternario:** Atajo para escribir condiciones de forma compacta:  
`condicion ? valor_si_verdadero : valor_si_falso.`
- **Template literals (Strings Literals):** Permiten insertar variables dentro de cadenas de texto usando backticks () y `${}`:

```
Hola, ${nombre}!
```

### 2. Funciones, Arrays y Objetos

#### Funciones

- **Funciones de Flecha:** Sintaxis concisa de funciones anónimas con `=>`. Ejemplo: `const suma = (a, b) => a + b;`
- **Funciones de Callback:** Funciones que se pasan como argumento a otra función y se ejecutan luego de una operación.

## Arrays y Operaciones de Arrays

- **Array:** Estructura de datos que almacena elementos en un solo objeto. Ejemplo: `const frutas = ['manzana', 'pera', 'uva'];`
- **Métodos de Arrays:**
  - `.map()`: Itera y retorna un nuevo array transformado.
  - `.filter()`: Filtra elementos según una condición.
  - `.reduce()`: Acumula valores en una operación.
  - `.forEach()`: Itera sobre cada elemento del array.

## Objetos y Operaciones con Objetos

- **Objeto:** Colección de pares clave-valor. Ejemplo:  
`const persona = { nombre: 'Ana', edad: 25 };`
- **Deestructuración:** Extraer valores de un objeto o array en variables independientes. Ejemplo:  
`const { nombre, edad } = persona;`
- **Spread Operator (...):** Permite copiar elementos de arrays y objetos sin modificarlos. Ejemplo:  
`const copiaPersona = { ...persona, ciudad: 'Santiago' };`

## 3. Asincronía y Eventos

### Try Catch y Spread Operator

- **Try...Catch:** Manejo de errores en JavaScript para prevenir fallos en la ejecución del código.

```
try {
  let resultado = 10 / 0;
} catch (error) {
  console.log('Error:', error.message);
}
```

## SetInterval, SetTimeout y Promesas

- **setTimeout():** Ejecuta un código después de un tiempo determinado. Ejemplo:

```
setTimeout(() => console.log('Hola luego de 2 segundos'), 2000);
```

- **setInterval():** Ejecuta un código repetidamente cada cierto intervalo de tiempo.
- **Promesa (Promise):** Objeto que representa la terminación (o falla) de una operación asíncrona.

```
const promesa = new Promise((resolve, reject) => {  
  let exito = true;  
  exito ? resolve('Operación exitosa') : reject('Error en la operación');  
});
```

## DOM, Eventos y Local Storage

- **DOM (Document Object Model):** Representación estructurada de una página web en JavaScript.
- **Eventos:** Acciones detectadas por el navegador, como clics, teclado, o carga de página.

```
document.getElementById('boton').addEventListener(  
'click', () => alert('Botón presionado'));
```

- **Local Storage:** Almacena datos en el navegador de manera persistente.

```
localStorage.setItem('usuario', 'Juan');  
console.log(localStorage.getItem('usuario')); // 'Juan'
```