

# JavaScript: Pila de Ejecucion y Event Loop

## ¿Qué es la pila de ejecucion (Call Stack)?

La pila de ejecucion es una estructura FILO (First In, Last Out). Es como una pila de platos: el ultimo que se pone arriba es el primero que se retira.

## Visualizacion de la pila

```
+-----+
| prueba1() |
+-----+
| prueba2() |
+-----+
| Global Context |
+-----+
```

## Ejemplo practico

```
function prueba1() {
  console.log("Hola desde prueba1");
}

function prueba2() {
  prueba1();
  console.log("Volviendo a prueba2");
}

let temp = "algo temporal";
prueba2();
```

Paso a paso:

1. Se declara temp (Contexto Global).
2. Se llama prueba2(), se apila.
3. Dentro, se llama prueba1(), se apila.
4. prueba1 imprime y se desapila.
5. prueba2 continua, imprime, y se desapila.

## Recursion y la pila

```
function crearNuevoContexto(n) {
  if (n === 0) return;
  console.log("Contexto:", n);
  crearNuevoContexto(n - 1);
}
```

## JavaScript: Pila de Ejecucion y Event Loop

```
crearNuevoContexto(3);
```

Visualizacion de la pila:

```
+-----+
| crearNuevoContexto(3) |
+-----+
| crearNuevoContexto(2) |
+-----+
| crearNuevoContexto(1) |
+-----+
| crearNuevoContexto(0) |
+-----+
```

### El Bucle de Eventos (Event Loop)

JavaScript es single-threaded, así que usa el Event Loop para manejar tareas asincronas.

### Como funciona

```
Codigo sincrono
|
[ Call Stack ]
|
[ Task Queue ]
|
[ Event Loop ]
|
[ Call Stack ]
```

### Ejemplo: setTimeout

```
console.log("Primer log");

setTimeout(() => {
  console.log("Segundo log");
}, 0);

console.log("Tercer log");
```

## JavaScript: Pila de Ejecucion y Event Loop

Flujo:

1. Primer log se ejecuta.
2. setTimeout se registra, su callback va a la cola.
3. Tercer log se ejecuta.
4. Cuando la pila esta vacia, se ejecuta el callback.

### Problemas comunes

```
function procesoPesado() {  
  let i = 0;  
  while (i < 1e9) {  
    i++;  
  }  
  console.log("Proceso pesado terminado");  
}  
  
document.getElementById("boton").addEventListener("click", () => {  
  console.log("Clic detectado");  
});  
  
procesoPesado();
```

El listener se registra, pero procesoPesado bloquea la pila.

Los clics no se procesan hasta que termine. Mala experiencia de usuario.

### Resumen visual del flujo

```
[Codigo Principal]  
  |  
[Pila de Ejecucion]  
  |  
[Cola de Mensajes (FIFO)]  
  |  
[Event Loop]  
  |  
[Pila de Ejecucion]
```