

# Análisis de caso M3L4

Daniel Quezada Agüero

## Descripción del caso

Una pequeña startup de tecnología está desarrollando una aplicación web interactiva para gestionar listas de tareas. Actualmente, el equipo de desarrollo enfrenta dificultades en la organización y optimización del código, ya que han trabajado sin una estructura clara. Necesitan implementar soluciones eficientes utilizando los conceptos fundamentales de JavaScript aprendidos en este módulo, evitando el uso de objetos para centrarse en variables, arreglos, ciclos y funciones.

## Desarrollo

### 1. Análisis de problema:

```
1  let tareas = [];  
2  
3  let totalTareas = 0;  
4  
5  function agregarTarea(tarea) {  
6      tareas.push(tarea);  
7      totalTareas++;  
8  }  
9  
10 function mostrarTareas() {  
11     for (let i = 0; i < tareas.length; i++) {  
12         console.log(i + 1 + ". " + tareas[i]);  
13     }  
14 }  
15  
16 function eliminarTarea(indice) {  
17     tareas.splice(indice, 1);  
18     totalTareas--;  
19 }
```

El problema más evidente del código es la variable `TotalTareas` siendo innecesaria, la variable solo se llama para cambiar su valor, llevando así que un registro del total de tareas, pero puede fácilmente lograrse usando la propiedad `length` del arreglo.

El arreglo de tareas puede declararse como un `const` y aun ser mutado por las funciones `push` y `splice`.

La función `agregarTareas` y `eliminarTareas` pueden modificarse para hacer que se puedan agregar tareas en cualquier punto y no solo el final de la lista. En teoría se podría usar otro tipo de iteraciones para mostrar la lista de tareas, pero afectará también el mostrar las tareas de forma numerada.

Con estos cambios, el código queda más legible y entendible, no tiene variables innecesarias y se redujeron la cantidad de funciones que usan la misma funcionalidad, haciéndolo más ligero y reduciendo la carga de computación necesaria para funcionar.