

## Resumen de los ejercicios

### 1. Palabras repetidas en un texto

- Entrada: cadena de texto.
- Salida: lista de palabras que aparecen más de una vez.
- Pista: usar un Map o un objeto para contar ocurrencias.

**Enunciado:** Implementa una función que reciba un texto y devuelva las palabras que se repiten.

**Ejemplo de ejecución:**

Introduce un texto: Recordar es fácil para quien tiene memoria, olvidar es difícil para quien tiene corazón

Se repiten las siguientes palabras: es, para, quien, tiene

---

### 2. Diferencia entre dos arrays

- Entrada: dos arrays de números.
- Salida: elementos que están en el primero pero no en el segundo.
- Pista: recorrer el primer array y filtrar con `includes`.

**Enunciado:** Implementa una función que reciba dos arrays y devuelva los elementos que están en el primero pero no en el segundo.

**Ejemplo:**

Array 1: [1, 2, 3, 4, 5]

Array 2: [3, 5]

La diferencia de ambos conjuntos es [1, 2, 4]

---

### 3. Frecuencia de caracteres

- Entrada: cadena de texto.
- Salida: Map con cada carácter y su frecuencia.
- Pista: recorrer la cadena y actualizar el Map.

**Enunciado:** Implementa una función que reciba una cadena y devuelva un Map con la frecuencia de cada carácter.

**Ejemplo:**

Introduce un texto: Hola mundo

Mapa de frecuencias: Map { 'h' => 1, 'o' => 2, 'l' => 1, 'a' => 1, ' ' => 1, 'm' => 1, 'u' => 1, 'n' => 1, 'd' => 1 }

#### 4. Anagramas con Map

- Entrada: dos palabras.
- Salida: indicar si son anagramas.
- Pista: comparar frecuencias de caracteres.

**Enunciado:** Escribe una función que reciba dos cadenas y verifique si son anagramas utilizando un Map.

**Ejemplo:**

Introduce una palabra: amor

Introduce otra palabra: roma

Las palabras son anagramas.

---

#### 5. Invertir palabra con pila (stack)

- Entrada: palabra.
- Salida: palabra invertida.
- Pista: usar un array como pila (push y pop).

**Enunciado:** Implementa una función que invierta el orden de una palabra usando una pila.

**Ejemplo:**

Introduce una palabra: amor

La palabra invertida es "roma".

---

#### 6. Bubble Sort con 100.000 elementos

- Entrada: array aleatorio grande.
- Salida: tiempo de ejecución.
- Pista: medir con `console.time` y `console.timeEnd`.

**Enunciado:** Implementa quicksort y mide el tiempo con 100.000 elementos.

**Ejemplo:**

El algoritmo quicksort ha tardado 0.6 segundos en ordenar 100000 elementos.

---

#### 7. Quicksort con 100.000 elementos

- Similar al ejercicio 6, pero implementando quicksort.
- Comparar tiempos con bubble sort.

**Enunciado:** Implementa quicksort y mide el tiempo con 100.000 elementos.

**Ejemplo:**

El algoritmo quicksort ha tardado 0.6 segundos en ordenar 100000 elementos.

---

## 8. Número primo

- Entrada: número.
- Salida: indicar si es primo.
- Pista: comprobar divisores hasta la raíz cuadrada.

**Enunciado:** Implementa una función que reciba un número y devuelva si es primo.

**Ejemplo:**

Introduce un número: 1009

1009 es primo.

---

## 9. Paréntesis balanceados

- Entrada: texto con paréntesis.
- Salida: indicar si están balanceados.
- Pista: usar una pila para abrir/cerrar.

**Enunciado:** Implementa una función que indique si los paréntesis están balanceados.

**Ejemplo:**

Introduce un texto: ()()

Están balanceados.

---

## 10. Secuencia válida de (), [], {}

- Entrada: secuencia de paréntesis, corchetes y llaves.
- Salida: indicar si es correcta.
- Pista: pila con reglas de apertura/cierre.

**Enunciado:** Verifica si una secuencia de paréntesis, corchetes y llaves es válida.

**Ejemplo:**

Introduce una secuencia: [(){}()

Secuencia correcta.

---

## Entrega y evaluación

- Cada ejercicio debe entregarse en un archivo .js.
- Al menos **un diagrama de flujo** en imagen (Draw.io, LucidChart, etc.).
- Se evaluará:
  - Que el código funcione.
  - Que cumpla el enunciado.
  - Que esté limpio y bien indentado.
  - Que el diagrama sea claro y fiel al código.

👉 Dado tu objetivo de construir un **portafolio freelance**, estos ejercicios son perfectos para mostrar dominio de algoritmos básicos y estructuras de datos. Además, puedes subirlos a GitHub y acompañarlos con diagramas para dar más profesionalidad.

¿Quieres que te prepare un **plan paso a paso** para resolverlos y documentarlos en tu portafolio (con ejemplos de código y diagramas sugeridos), o prefieres que empecemos directamente con el código del **Ejercicio 1**?