

# Guía de ejercicios de arreglos en ruby

Crear una carpeta, dentro de la carpeta guardar cada archivo .rb con el número de la pregunta, subir los ejercicios completados a Github y a la plataforma empieza

# Ejercicios de Arrays

## Ejercicio 1: Array Básicos

---

Dado el array `[1, 2, 3, 9, 1, 4, 5, 2, 3, 6, 6]`

1. Mostrar en pantalla el primer elemento.
2. Mostrar en pantalla el último elemento.
3. Mostrar en pantalla todos los elementos.
4. Mostrar en pantalla todos los elementos junto con un índice.
5. Mostrar en pantalla todos los elementos que se encuentren en una posición par.
6. Determinar con el método aprendido en clases si un elemento ingresado pertenece al array o no, mostrar el resultado en pantalla

## Ejercicio 2: Operaciones de push y pop en arrays

---

Dado el array `[1, 2, 3, 9, 1, 4, 5, 2, 3, 6, 6]`

1. Eliminar el último elemento.
2. Eliminar el primer elemento.
3. Eliminar el elemento que se encuentra en la posición media, si el arreglo tiene un número par de elementos

entonces remover el que se encuentre en la mitad izquierda, ejemplo, en el arreglo [1,2,3,4] se removería el elemento 2.

4. Borrar el último elemento mientras ese número sea distinto a 1.
5. Utilizando un arreglo vacío auxiliar y operaciones de push and pop invertir el orden de los elementos en un arreglo.

## Ejercicio 3: Array, operaciones numéricas y métodos.

---

Dado un array como el siguiente  
`[1,2,3,9,1,4,5,2,3,6,6]` :

1. Crear un método para eliminar todos los números pares del arreglo.
2. Crear un método para obtener la suma de todos los elementos del arreglo Utilizando `.each`
3. Crear un método para obtener el promedio de un arreglo.
4. Crear un método que incrementa todos los elementos en una unidad y devuelva un arreglo nuevo.

## Ejercicio 4: Manipulación de múltiples arrays

---

Se tienen dos arreglos

El primero es del tipo [1,2,3,0,1,2,2,1,2,1,2,0,3] y el

segundo es del tipo [:azul, :verde, :amarillo]

Se pide cambiar todas las apariciones del número que aparece en el arreglo 1 por el elemento con el mismo índice del arreglo 2, en caso de no existir el elemento deberá ser remplazado por nil.

El resultado de este ejercicio debería quedar:

```
[ :verde, :amarillo, nil, :azul, :verde, :amarillo ....]
```

## Pregunta 5

---

Se tiene un string del tipo "1,2,7,1,3,5,3.4,9,1"

Se pide:

- Calcular el promedio
- Calcular la moda (o sea mostrar el número que más se repite)

