

# Actividad 012 - Métodos en Ruby

## INSTRUCCIONES:

- Para poder realizar esta actividad debes haber realizado los cursos previos junto con los videos online correspondientes a la experiencia 6.
- Los archivos necesarios para realizar los ejercicios los puedes encontrar en [este link](#).
- Realizar un clon del repositorio y sigue las instrucciones de manera local con Sublime o Atom.
- Luego guarda los cambios y súbelos a tu repositorio de Github.
- Luego de pusheados los últimos cambios, sube el link de Github en el desafío de la sección correspondiente en la plataforma.

## Comencemos...

## Métodos

---

### Ejercicio 1

Optimiza el código utilizando tu editor de texto.

*hint: Debes utilizar 'rubocop' y no obtener errores ni warnings.*

### Ejercicio 2

El siguiente programa debería mostrar sí o no, sin embargo mostrar error. Se pide identificar el error y corregirlo.

## Ejercicio 3

Escriba un método llamado **check5** que devuelva *true* cuando se le pase un número mayor a 5 y *false* en caso contrario.

## Ejercicio 4

Crear un método que muestre en pantalla un saludo, el método debe recibir un parámetro, si ese parámetro es el string "Hola" el método debe mostrar en pantalla "Hola Mundo".

## Ejercicio 5

Crear un método que reciba como parámetro dos números enteros positivos e imprima los números pares que existen entre esos dos números.

## Ejercicio 6

Un método puede llamar a otros métodos, modifica el siguiente código para que al llamar a un sólo método se imprima:

```
*****  
*****  
*****  
*****  
*****
```

# Arrays

---

## Ejercicio 1

Dado el array:

```
arreglo = [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 su índice.
5. Mostrar en pantalla todos los elementos que se encuentren en una posición par.

## Ejercicio 2

Dado el array:

```
a = [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

Dado el array:

```
a = [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

Se tiene un arreglo de productos, se pide que el output sea:

```
<div class='product'><p> Producto1 </p></div>
<div class='product'><p> Producto2 </p></div>
<div class='product'><p> Producto3 </p></div>
<div class='product'><p> Producto4 </p></div>
```

## Ejercicio 5

Se tiene un arreglo de productos y precios, se pide que el output sea:

```
<div class='product'><p> Producto1 </p><p> Precio: 1000 </p></div>
<div class='product'><p> Producto2 </p><p> Precio: 2000 </p></div>
<div class='product'><p> Producto3 </p><p> Precio: 1500 </p></div>
<div class='product'><p> Producto4 </p><p> Precio: 950 </p></div>
```

# Strings

---

## Ejercicio 1

Dado el siguiente string y caracter, crear un método que reciba como parámetro el string y el caracter. Luego debe buscar si existe ese caracter dentro del string.

*hint: El método .include? de un string busca si un caracter o string dado está contenido en éste.*

## Ejercicio 2

Construir un arreglo de los nombres de todos sus compañeros y en base a él:

1. Imprimir todos los elementos que excedan más de 5 caracteres.
2. Crear un arreglo nuevo con todos los elementos en minúscula.
3. Crear un método que devuelva un arreglo con la cantidad de caracteres que tiene cada nombre.

## Ejercicio 3

Dado los arrays

```
a = [1,2,3,9,12,31, "domingo"]  
b = ["lunes", "martes", "miércoles", "jueves", "viernes", "sábado", "domingo"]
```

Obtener:

1. La concatenación de a y b. (hint: Los elementos que se repiten en a y b pueden aparecer dos veces en el resultado).
2. La unión de a y b. (hint: Los elementos que se repiten en a y b NO deben aparecer repetidos en el resultado).
3. La intersección de a y b. (hint: El resultado debe estar compuesto por los elementos que se repiten en a y b).
4. Intercalar los elementos:

```
resultado = [[1, "lunes"], [2, "martes"], [3, "miércoles"], [9, "jueves"], [12, "viernes"], [31, "domingo"]]
```

