

# Desafío - Métodos

---

## Instrucciones

- Realiza los ejercicios utilizando tu editor de texto Atom o Sublime.
- Guarda los cambios y súbelos a un repositorio en tu cuenta de Github.
- Una vez enviados los últimos cambios, sube el link de tu repositorio de Github en el desafío de la sección correspondiente en la plataforma.

### 1)

El siguiente algoritmo es algo redundante, optimízalo:

```
def par(x)
  if x % 2 == 0
    return true
  else
    return false
  end
end

puts par(2)
puts par(3)
puts par(4)
puts par(5)
```

### 2)

El siguiente algoritmo debería mostrar 'sí' o 'no', sin embargo muestra error. Se pide identificar el error y corregirlo:

```
def random
  result = [true, false].sample
  puts result
end

if random == true
  puts 'sí'
elsif random == false
  puts 'no'
else
  puts 'error'
end
```

### 3)

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

```
puts check5(5) # Debería ser false
puts check5(6) # Debería ser true
```

### 4)

Crear un método que imprima un saludo. El método debe recibir un parámetro, si ese parámetro es el string "Hola" el método debe imprimir "Hola Mundo".

### 5)

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

### 6)

Un método puede llamar a otros métodos:

```
def draw_line(size)
  ! * size
end

def draw_lines(size)
  size.times { }
end

draw_line 2
draw_lines 10
```

Modifica el siguiente código para que al ejecutar un sólo método se imprima:

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

## 7)

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

```
cadena = 'Hola Mundo!'
caracter = 'o'
```

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

## 8)

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.