

Desafío - Estructuras de control

<https://github.com/DesafioLatam/taller-estructuras-control>

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.

Condicional: If and else

Ejercicio 1

Modifica la condición para que se cumpla.

```
a = 2
if a == 'saludo'
  puts 'La condición es verdadera.'
end
```

Ejercicio 2

Modifica el valor asignado a la variable 'a' para que se cumpla la condición.

```
a = 2
if a == 5
  puts 'La condición es verdadera.'
end
```

Ejercicio 3

Hacer un refactoring, transformando las últimas 3 líneas en una sola línea.

```
a = 'X9-by'

if a == 'X9-by'
  puts 'HOLA!'
end
```

Ejercicio 4

La variable 'password' no se encuentra definida.

Permitir que el usuario pueda ingresar la contraseña por teclado, almacenar esta contraseña en la variable 'password' y luego evaluar la condición.

```
if password == 'secreto'
  puts 'Acceso PERMITIDO! :)'
else
  puts 'Acceso DENEGADO! :('
end
```

Ejercicio 5

Utiliza lógica booleana para hacer un refactoring de este código. Para verificar la evaluación de condiciones puedes modificar los valores de a y b.

```
a = true
b = true

if a == true
  if b == true
    puts 'Lograste A y B!'
  else
    puts 'Lograste A! Pero no B!'
  end
else
  puts 'No lograste A ni B!'
end
```

Ejercicio 6

Utiliza álgebra booleana para hacer un refactoring de este código. Para verificar la evaluación de condiciones puedes modificar los valores de a y b.

```
a = 'verdadero'
b = 'falso'

if a == 'verdadero'
  puts ':'
else
  if b == 'verdadero'
    puts ':'
  else
    puts '('
  end
end
end
```

Ciclos Iterativos

Ejercicio 1

En el siguiente código reemplaza la instrucción 'for' por 'times'.

```
for i in 1..10 do
  puts i
end
```

Ejercicio 2

En el siguiente código reemplaza la instrucción 'while' por 'times'.

```
i = 0
while i < 10
  puts "Iteración #{i}"
  i = i + 1
end
```

Ejercicio 3

Mostrar todos los divisores del número 990 con 'while', 'for' y 'times'.

Ejercicio 4

En el siguiente código:

```
a = 5
b = ""

a.times do
  b = '<li> hola </li>'
end
```

Modifica el código para que el contenido de b sea un string que contenga:

```
<ul>
  <li> hola </li>
  <li> hola </li>
  <li> hola </li>
  <li> hola </li>
  <li> hola </li>
</ul>
```

Ejercicio 5

El siguiente código busca sumar todos los números del 1 al 10, pero no funciona porque algo falta, ¿puedes arreglarlo?

Hint: El total debería sumar 55.

```
10.times do |i|  
  suma += i  
end  
  
puts suma
```

Ejercicio 6

El siguiente algoritmo pretende calcular la multiplicación de los números de 1 a 10. (Factorial de 10)

Hint: El resultado es 3628800.

```
multiplicacion = 1  
  
10.times do |i|  
  multiplicacion *= i  
end  
  
puts multiplicacion
```

Ejercicio 7

El siguiente código:

```
a = 10  
a.times do |i|  
  puts i  
  if i.even?  
    puts 'par'  
  end  
end
```

Debería imprimir la siguiente secuencia, pero no está completo:

```
1
par
3
par
5
par
7
par
9
par
```

Ejercicio 8

Generar -utilizando un ciclo iterativo- un string con la siguiente estructura:

```
a = '1impar 2par 3impar 4par 5impar 6par 7impar 8par 9impar 10par'
```

```
a = ""
10.times do |i|
  # aquí agregar instrucciones necesarias
end

puts a
```

Ejercicio 9

Crear un algoritmo que permita generar un string con el siguiente contenido:

```
<table>
<tbody>
  <tr>
    <td> 1 </td>
    <td> 2 </td>
    <td> 3 </td>
  </tr>
</tbody>
</table>
```

Ejercicio 10

El siguiente bloque de código debería mostrar un menú e imprimirlo reiteradamente hasta que el usuario ingrese la opción número 4.

```
ready = 0
while (ready == 0){
  puts "Opción 1: blah"
  puts "Opción 2: blah"
  puts "Opción 3: blah"
  puts "Opción 4: Salir"
  opcion = get.chomp.to_i
}
```

Ciclos Iterativos Anidados

Ejercicio 1

Se pide imprimir la secuencia numérica, de la siguiente forma:

```
1 2 3 4
2 4 6 8
3 6 9 12
4 8 12 16
```

Ejercicio 2

Generar el código para imprimir un string con el siguiente contenido:

```
<table>
<tbody>
<tr>
<td> 1 </td>
<td> 2 </td>
<td> 3 </td>
<td> 4 </td>
</tr>
<tr>
<td> 5 </td>
<td> 6 </td>
<td> 7 </td>
<td> 8 </td>
</tr>
<tr>
<td> 9 </td>
<td> 10 </td>
<td> 11 </td>
<td> 12 </td>
</tr>
</tbody>
</table>
```

Ejercicio 3

Construir un programa que permita ingresar un número por teclado e imprimir la tabla de multiplicar del número ingresado.

Debe repetir la operación (volver a preguntar por un número) hasta que se ingrese un 0 (cero).

```
Ingrese un número (0 para salir): _
```