

# Ruby - Unidad 03

Los bucles

Staff Pedagógico 42 pedago@42.es

Resumen: En esta unidad vamos a ver cómo funcionan los bucles.

# Índice general

I.	Preámbulo	2
II.	Instrucciones Generales	3
III.	Ejercicio 00: to25	4
IV.	Ejercicio 01: table_mult	5
V.	Ejercicio 02: i_got_that	6
VI.	Ejercicio 03: advanced mult	7

## Capítulo I

#### Preámbulo

He aquí la lista de los pósteres de motivación que se pueden ver en la oficina de Barney Stinson en las distintas temporadas de How I Met Your Mother:

- Awesomeness: "When I get sad, I stop being sad and be Awesome Instead. True Story. Barney Stinson"
- Conformity: Ït's the one who is different that gets left out in the cold."
- Courage: "True greatness comes when you're tested. Theodore Roosevelt"
- Challenge: "We either find a way or we make one"
- Opportunity: "You will always miss 100 per cent of the shots you don't take."
- Teamwork: Çoming together is the beginning. Keeping together is progress. Working together is success. Henry Ford."
- Teamwork: "The chain is only as strong as the weakest link"
- Perseverance: Çontinuous effort is the key to unlocking your potential. Sir Winston Churchill"
- Perfection: It is not good enough to win, everybody else should lose"
- Strength: "What the mind can conceive, it can achieve"

#### Capítulo II

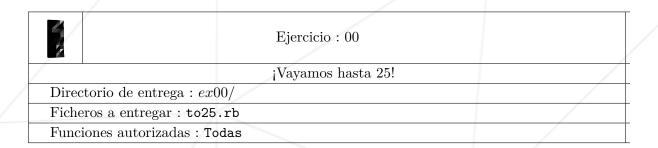
#### Instrucciones Generales

Salvo que se indique lo contrario de forma explícita, las siguientes instrucciones serán válidas durante el tiempo que dure la Piscina.

- Esta evaluación será la única referencia: no se fíe de los rumores de pasillo.
- Los ejercicios han sido ordenados con mucha precisión del más sencillo al más complejo. En ningún caso prestaremos atención a un ejercicio complejo ni lo tendremos en cuenta si no se ha conseguido realizar perfectamente un ejercicio más sencillo.
- Tenga cuidado con los permisos de sus archivos y de sus directorios.
- Sus compañeros de Piscina se encargarán de corregir los ejercicios que usted realice.
- Para la evaluación entre pares, <u>no debe</u> dejar en su directorio <u>ningún</u> archivo que no haya sido indicado de forma explícita en los enunciados de los ejercicios.
- ¿Tiene alguna pregunta? Pregunte a su vecino de la derecha. Si no, pruebe con su vecino de la izquierda.
- Todas las respuestas a sus preguntas técnicas se encuentran en los man o en Internet.
- ¡No olvide participar en el foro Piscina de su Intranet y en el Slack!
- Lea detenidamente los ejemplos ya que pueden ayudarle a identificar algún trabajo que tenga que realizar y que, a primera vista, no venga explicado en el enunciado.
- Razone. ¡Se lo suplico, por Thor, por Odín!

#### Capítulo III

#### Ejercicio 00: to25



- Cree el programa to25.rb.
- Tiene que ser un programa ejecutable.
- Este programa deberá aceptar una entrada del usuario. La entrada será un número que tendrá que almacenar en una variable con formato numérico.
- Después tendrá que crear un bucle que muestre todos los números, desde el número proporcionado hasta 25.
- Si la entrada es superior a 25, mostrará . Error "seguido de un salto de línea.

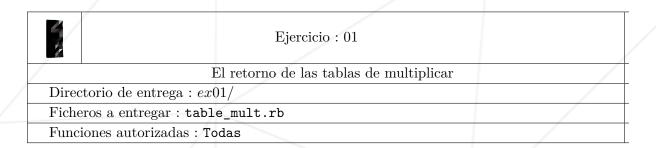
```
?> ./to25.rb
Introduzca un número superior a 25
45
Error
?> ./to25.rb
Introduzca un número superior a 25
20
En el bucle, mi variable es igual a 20
En el bucle, mi variable es igual a 21
En el bucle, mi variable es igual a 22
En el bucle, mi variable es igual a 22
En el bucle, mi variable es igual a 23
En el bucle, mi variable es igual a 24
En el bucle, mi variable es igual a 25
?>
```



While.

### Capítulo IV

# Ejercicio 01: table\_mult

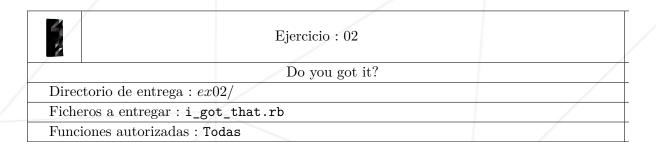


- Cree el programa table\_mult.rb
- Tiene que ser un programa ejecutable.
- Este programa deberá aceptar un input del usuario. El input será un número y tendrá que almacenarlo en una variable con formato numérico.
- El número corresponde a la tabla de multiplicar que debe mostrar. (Por ejemplo, si el input es 2, mostrará la tabla del 2)

```
?> ./to25.rb
Introduzca un número
8
0 x 8 = 0
1 x 8 = 8
2 x 8 = 16
3 x 8 = 24
4 x 8 = 32
5 x 8 = 40
6 x 8 = 48
7 x 8 = 56
8 x 8 = 64
9 x 8 = 72
?>
```

### Capítulo V

## Ejercicio 02: i\_got\_that



- Cree el programa i\_got\_that.rb.
- Tiene que ser un programa ejecutable.
- Este script tendrá que incluir un bucle while que acepte un input del usuario, devolverá una frase y se parará únicamente cuando el usuario introduzca "STOP".
- En cada iteración del bucle tendrá que aceptar un input del usuario.

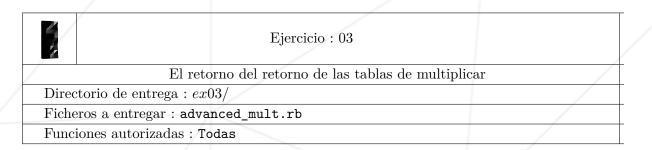
```
?> ./i_got_that.rb
¿Qué quiere decir? : Hola
Entendido. ¿Algo más? : La vida es una tómbola
Entendido. ¿Algo más? : stop...
Entendido. ¿Algo más? : STOP
?>
```



While, break.

#### Capítulo VI

### Ejercicio 03: advanced mult



- Cree el programa advanced\_mult.rb
- Tiene que ser un programa ejecutable.
- Este programa tendrá que mostrar todas la tablas de multiplicar de la siguiente forma:

```
?> ./advanced_mult.rb "yolo" | cat -e
none$
?> ./advanced_mult.rb
Tabla de 0: 0 0 0 0 0 0 0 0 0 0 0 0
Tabla de 1: 0 1 2 3 4 5 6 7 8 9 10
Tabla de 2: 0 2 4 6 8 10 12 14 16 18 20
Tabla de 3: 0 3 6 9 12 15 18 21 24 27 30
Tabla de 4: 0 4 8 12 16 20 24 28 32 36 40
Tabla de 5: 0 5 10 15 20 25 30 35 40 45 50
Tabla de 6: 0 6 12 18 24 30 36 42 48 54 60
Tabla de 7: 0 7 14 21 28 35 42 49 56 63 70
Tabla de 8: 0 8 16 24 32 40 48 56 64 72 80
Tabla de 9: 0 9 18 27 36 45 54 63 72 81 90
Tabla de 10: 0 10 20 30 40 50 60 70 80 90 100
?>
```

• Solo tiene derecho a dos bucles while.