

MANUAL
DE RUBY
(PARTE III)

Luis José Sánchez González

1. BUCLE WHILE

En ruby disponemos de iteradores como es el caso de **each**, con lo que la mayoría de las veces es innecesario utilizar bucles **while** o **for**.

Utilizaremos **each** siempre que sea posible con el fin de evitar introducir variables innecesarias en el código. No obstante vamos a ver cómo funcionan los bucles.

El formato del bucle **while** es el siguiente:

```
while expr [do]
  ...
end
```

Se ejecuta el código entre **while** y **end** mientras **expr** sea verdadero.

Ejemplo:

```
hay_luz = true

while hay_luz do
  puts "Estoy leyendo..."
  puts "¿Sigue habiendo luz?"
  hay_o_no = gets.chomp
  if hay_o_no == "no" then hay_luz = false end
end

puts "Ya no leo más"
```

2. BUCLE FOR (AL ESTILO C)

La sintaxis del bucle **for** es la siguiente:

```
for var in expr [do]
  ...
end
```

Ejecuta el código para cada elemento de **expr** asignándoselo a **var**.

Veamos un ejemplo:

```
puts "Tabla de multiplicar del 5"
for i in (0..10) do
  puts "5 x #{i} = #{5*i}"
end
```

3. BUCLE FOR (AL ESTILO EACH)

El rango del **for** puede ser un array:

```
puts "Mira cuantas cosas tengo"
for x in ["peluche", 5, 77, "helado de vainilla", 0] do
  puts "Tengo un #{x}"
end
```

En este último ejemplo, el uso del **for** es equivalente al del **each**.

```
puts "Mira cuantas cosas tengo"
["peluche", 5, 77, "helado de vainilla", 0].each do |x|
  puts "Tengo un #{x}"
end
```

4. EJECUCIÓN DE COMANDOS EXTERNOS

La función predefinida **system()** permite ejecutar un comando como si lo tecleásemos en una ventana de terminal.

Ejemplo:

```
system("clear") # Borra la pantalla
system("date") # Muestra la fecha actual
puts "Introduzca el nombre del directorio que desea crear: "
d = gets.chomp
system("mkdir #{d}")
```

5. NÚMEROS ALEATORIOS

La función **rand(max)** genera un número aleatorio entre **0** y **max-1**. Por ejemplo **rand(8)** genera un número aleatorio entre 0 y 7.

Si no se especifica ningún parámetro, es decir, si escribimos **rand()**, se genera un número en punto flotante entre 0 y 1 (sin llegar nunca a 1).

Ejemplo:

```
puts "Al tirar el dado ha salido el #{rand(6)+1}"
```

Ejercicios

1. Muestra la tabla de multiplicar de un número introducido por teclado.
2. Escribe un programa que diga si un número introducido por teclado es o no primo.
3. Crea un lanzador de las 5 o 6 aplicaciones que más utilices en Linux. Será simplemente un menú que llame a la aplicación correspondiente eligiendo una de las opciones.
4. Programa el juego de las 21. Tanto las cartas de la banca (el ordenador) como las cartas del jugador deben generarse de forma aleatoria. Empezará jugando la banca, sacando dos cartas de las que mostrará sólo la primera. A continuación, el jugador sacará dos cartas y, a partir de ahí podrá ir pidiendo cartas o plantarse. Cuando el jugador se plante, continuará la banca. Gana quien más se aproxime sin pasarse a 21 puntos. El as vale un punto u once, según nos interese, las figuras valen diez puntos y las demás cartas valen el número que tengan. Para saber más sobre el funcionamiento del juego, se puede jugar en *Aplicaciones -> Juegos -> Blackjack*.