



## Ruby - Unidad 02

### Las sentencias condicionales

Staff Pedagógico 42 [pedago@42.es](mailto:pedago@42.es)

*Resumen: En esta unidad vamos a ver cómo funcionan las sentencias condicionales.*

# Índice general

I.	Preámbulo	2
II.	Instrucciones Generales	4
III.	Ejercicio 00: ¿Soy nulo?	5
IV.	Ejercicio 01: ¿Soy negativo?	6
V.	Ejercicio 02: Contraseñas	7
VI.	Ejercicio 03: Multiplicación	8

# Capítulo I

## Preámbulo

El juego del Sirope según Perceval, sacado de la serie Kaamelott:  
"Bueno, os voy a enseñar las reglas simplificadas, porque las de verdad tienen miga. A ver, la única pega es que normalmente se juega a tres. Pero no importa, nos las vamos a apañar.

El objetivo es anunciar valores. Por ejemplo, pongamos que aquí somos tres, se distribuyen tres valores. Digamos: sirope de ocho, sirope de catorce y sirope de veintiuno. De momento no os preocupéis por los siropes. Lo que hay que entender son los valores. Si apostáis un valor al principio de una ronda, digamos un sirope de ocho, para empezar por lo bajo, los demás pueden elegir entre dejar pasar la apuesta o envidar con un sirope de catorce. Se gira en el sentido de los valores. Por eso, hay que comprender bien el sistema de valores; después todo va sobre ruedas.

Vale, digamos que abro con un sirope de ocho.  
Si sois vosotros los que habéis sorbido en la ronda anterior, entonces gira en vuestro sentido. En ese caso, o dejáis pasar diciendo "paso sirope" u os animáis a lanzar y anunciáis un sirope de catorce. Como hemos iniciado los anuncios, el segundo jugador no tiene derecho a pasar. Podéis o bien apostar un sirope de veintiuno o abandonar vuestro turno diciendo "sirope echado" o "sirope Juanito", depende de la región. Después, o se sigue la partida o ¡hago un "contrasirope"! Y a partir de ahí, sirope de manzana sobre veintiuno y entonces jugamos la partida en cuatro rondas hasta que haya uno que sorba.

Para ganar solo hay tres posibilidades: o anunciáis vuestro sirope de ocho diciendo "buen sirope" y volvemos a contar, o anunciáis vuestro sirope de catorce diciendo "buen sirope, sirope ganador" y se os añade la mitad, o anunciáis vuestro sirope de veintiuno diciendo "buen sirope, medio sirope, sorbido, sirope ganador, sirope cascabel, pasamontañas, sirope rico".

Normalmente se juega con cartas, pero si solo tenéis dados, también se puede jugar con dados porque lo que cuentan son los valores."

Al menos uno de los ejercicios siguientes no tiene ninguna relación con el juego del Sirope.

# 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, no debe dejar en su directorio ningún 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: ¿Soy nulo?


	Ejercicio : 00
¿Soy nulo?	
Directorio de entrega : <i>ex00/</i>	
Ficheros a entregar : <b>iszero.rb</b>	
Funciones autorizadas : Todas	

- Cree el programa **iszero.rb**
- Tiene que ser un programa ejecutable. (No se olvide de los permisos.)
- Cuando ejecutemos el programa nos tendrá que pedir que introduzcamos un número.
- Si el número es igual a cero, el programa mostrará *Este número es igual a cero*".
- Si el número es distinto de cero, el programa mostrará *Este número es distinto de cero*".

```
?> ./iszero.rb
42
Este número es diferente de cero
?> ./iszero.rb
0
Este número es igual a cero
?>
```

# Capítulo IV

## Ejercicio 01: ¿Soy negativo?


	Ejercicio : 00
¿Soy negativo?	
Directorio de entrega : <i>ex00/</i>	
Ficheros a entregar : <i>isneg.rb</i>	
Funciones autorizadas : Todas	

- Cree el programa `isneg.rb`
- Tiene que ser un programa ejecutable.
- Cuando ejecutemos el programa nos tendrá que pedir que introduzcamos un número.
- Si el número es negativo, el programa mostrará `.Este número es negativo`.
- Si el número es positivo, el programa mostrará `.Este número es positivo`.
- Si el número es igual a cero, el programa mostrará `.Este número es positivo y negativo`.

```
?> ./isneg.rb
42
Este número es positivo
?> ./isneg.rb
-42
Este número es negativo
?> ./isneg.rb
0
Este número es negativo y positivo
?>
```

# Capítulo V

## Ejercicio 02: Contraseñas

	Ejercicio : 02
Access denied	
Directorio de entrega : <i>ex02/</i>	
Ficheros a entregar : <b>password.rb</b>	
Funciones autorizadas : Todas	

- Cree el programa `password.rb`
- Tiene que ser un programa ejecutable.
- El programa debe tener una variable que contenga una contraseña.

```
password = "ruby mola"
```


- Cuando ejecutemos el programa nos tendrá que pedir que introduzcamos una contraseña.
- Si la contraseña es correcta, el programa mostrará `^ACCESO PERMITIDO` si no `^ACCESO DENEGADO`.

```
?> ./password.rb
1234
ACCESO DENEGADO
?> ./password.rb
ruby mola
ACCESO PERMITIDO
?>
```



# Capítulo VI

## Ejercicio 03: Multiplicación

	Ejercicio : 03
Multiplicación	
Directorio de entrega : <i>ex03/</i>	
Ficheros a entregar : <i>mult.rb</i>	
Funciones autorizadas : Todas	

- Cree el programa `mult.rb`
- Tiene que ser un programa ejecutable.
- Cuando ejecutemos el programa nos tendrá que pedir que introduzcamos 2 números.
- El programa mostrará si el resultado de la multiplicación de los 2 números es positivo, negativo o nulo.
- Después, el programa mostrará el resultado de la multiplicación.

```
$ ./mult.rb
Introduzca un primer número:
42
Introduzca un segundo número:
42
El resultado es positivo
42 x 42 = 1764

$ ./mult.rb
Introduzca un primer número:
78
Introduzca un segundo número:
-1
El resultado es negativo
78 x -1 = -78

$ ./mult.rb
Introduzca un primer número:
72
Introduzca un segundo número:
0
El resultado es positivo y negativo
72 x 0 = 0
```