

Relación de ejercicios nº 2: Reglas

NOTA: Para resolver estos ejercicios no pueden emplearse prioridades, módulos ni funciones procedimentales (salvo, en todo caso, la función *bind*).

1. En una granja hay animales de los siguientes tipos: perros, gatos, patos, vacas, ovejas y gallos. Escribe una serie de reglas de manera que en base a un hecho que indique el sonido que hace un animal identifique qué clase de animal es. Por ejemplo, si se afirma un hecho como
(sonido kikiriki)

al ejecutar el programa debería imprimirse por pantalla un mensaje como
Se trata de un gallo

2. Repite el ejercicio anterior pero utilizando ahora una plantilla como esta
(deftemplate animal
 (slot tipo)
 (slot sonido))

y una sola regla, de manera que al afirmar un hecho como
(sonido kikiriki)

la regla muestre por pantalla un mensaje como
kikiriki -> gallo

3. Supongamos ahora que se quiere mantener información también sobre el nombre propio de cada uno de los animales, por lo que usa una plantilla como esta
(deftemplate animal
 (slot nombre)
 (slot tipo)
 (slot sonido))

Ahora el objetivo es escribir un conjunto de reglas de manera que cuando se afirme un hecho como

(animal (nombre Toby) (tipo perro))

una regla modifique ese hecho rellenando de manera adecuada la casilla sonido, y además muestre por pantalla un mensaje como
Toby dice guau

4. Repite el ejercicio anterior utilizando un conjunto de hechos iniciales y una sola regla (si es que no lo has hecho ya así).
5. Haz un programa que calcule el factorial de un número, de manera que ante un hecho como
(factorial 3)
muestre por pantalla el siguiente mensaje
El factorial de 3 es 6

6. Haz un programa para calcular el factorial de un número pero que funcione cuando haya afirmados varios hechos *factorial* a la vez, de manera que si en la lista de hechos se encuentran los hechos

(factorial 3)

(factorial 4)

(factorial 5)

al ejecutar el programa dé la salida

El factorial de 5 es 120

El factorial de 4 es 24

El factorial de 3 es 6

7. Una planta industrial tiene diez sensores identificados por un código numérico entre 1 y 10. Cada sensor puede encontrarse en un estado *correcto* o *incorrecto*. Escribe una plantilla que permita representar la información relativa a los sensores y un conjunto de reglas que imprima un mensaje de advertencia si tres o más sensores se encuentran en un estado *incorrecto*. Sólo debe mostrarse un mensaje de error aunque haya más de tres sensores en estado incorrecto.

8. Define una plantilla para almacenar fechas, y escribe una regla que compruebe si la fecha no es válida. (Nota: un año es bisiesto si es múltiplo de 4 pero no de 100, salvo cuando sea múltiplo de 400, en cuyo caso sí que es bisiesto.)

9. Escribe un programa para ayudar a una persona a decidir qué plantas podría plantar. La siguiente tabla indica las características de una serie de plantas (tolerancia al frío, tolerancia a la sombra, tolerancia al clima seco, tolerancia al suelo húmedo, tolerancia al suelo ácido, si puede vivir en la ciudad, si puede vivir en una maceta, si es fácil de cuidar y si es de crecimiento rápido). La entrada al programa debe consistir en un conjunto de hechos del tipo

(característica-deseada <característica>)

que indiquen características que se desee que tenga una planta. El programa debe mostrar por pantalla el nombre de las plantas que cuenten exactamente con las características indicadas, ni más ni menos.

Planta	Frío	Sombra	Sequedad	Suelo húmedo	Suelo ácido	Ciudad	Maceta	Cuidado fácil	Crece rápido
Hortensia		X				X	X		X
Adelfa						X	X	X	X
Laurel	X	X	X	X		X		X	X
Madreselva						X	X	X	X
Gardenia		X			X		X		
Enebro	X		X		X	X		X	
Pimentero	X	X		X	X			X	
Escaramujo	X	X		X		X		X	
Aucuba		X	X				X	X	
Azalea		X		X	X		X		

10. Haz un programa en base a la información del ejercicio anterior, pero con las siguientes diferencias con respecto al anterior:

- El programa indica qué plantas pueden plantarse mediante hechos del tipo (`deftemplate puede-plantarse (slot planta)`), no mediante mensajes por pantalla.
- El programa señalará todas las plantas que cumplan todas las características deseadas, aunque aparte de esas cuenten con otras características.

11. Modifica el programa del ejercicio anterior para que la lista de plantas que pueden plantarse se actualice (previa ejecución de *run*) cada vez que el usuario modifique la lista de características deseadas.

A continuación se da un ejemplo de cómo debería comportarse el programa

```
CLIPS> (reset)
CLIPS> (assert (caracteristica-deseada frio))
<Fact-11>
CLIPS> (run)
CLIPS> (facts 11)
f-11    (caracteristica-deseada frio)
f-12    (puede-plantarse (planta escaramujo))
f-13    (puede-plantarse (planta pimentero))
f-14    (puede-plantarse (planta enebro))
f-15    (puede-plantarse (planta laurel))
For a total of 5 facts.
CLIPS> (assert (caracteristica-deseada crecimiento-
rapido))
<Fact-16>
CLIPS> (run)
CLIPS> (facts 11)
f-11    (caracteristica-deseada frio)
f-15    (puede-plantarse (planta laurel))
f-16    (caracteristica-deseada crecimiento-rapido)
For a total of 3 facts.
CLIPS> (retract 11)
CLIPS> (run)
CLIPS> (facts 15)
f-15    (puede-plantarse (planta laurel))
f-16    (caracteristica-deseada crecimiento-rapido)
f-17    (puede-plantarse (planta hortensia))
f-18    (puede-plantarse (planta adelfa))
f-19    (puede-plantarse (planta madreselva))
For a total of 5 facts.
```

12. Escribe un programa que convierta un conjunto de hechos (numero <número>) en un único hecho (numeros <número>+) en el que se encuentren los mismos valores que en los hechos anteriores ordenados de menor a mayor de izquierda a derecha. Por ejemplo, los hechos

(numero 5)

(numero 9)

(numero 2)

se transformarían en el hecho

(numeros 2 5 9)

13. En una residencia de estudiantes se desea mantener información sobre los estudiantes alojados y las habitaciones de la residencia que ocupan. Se tendrán en cuenta las siguientes consideraciones:

- Hay habitaciones de cuatro tipos: simples, dobles, triples y cuádruples.
- Es más económico llenar las habitaciones más grandes.
- Todos los ocupantes de una habitación deben de ser del mismo sexo.
- Todos los ocupantes de una habitación deben ser fumadores o no fumadores.

Define las plantillas necesarias para poder almacenar la información necesaria.

Escribe las reglas necesarias para alojar a un estudiante de manera que sea asignado a una habitación que esté ya parcialmente ocupada y sea compatible o si no que sea asignado a la mayor habitación libre disponible.

Si no hay ninguna habitación disponible para un estudiante, deberá mostrarse un mensaje por pantalla indicándolo.

14. Se utiliza la plantilla

```
(deftemplate producto
  (slot nombre)
  (slot pvp))
```

para almacenar información referente a los productos a la venta en un supermercado. La información relativa a los productos que desea comprar un cliente se indica mediante un conjunto de hechos basados en la siguiente plantilla:

```
(deftemplate compra
  (slot producto)
  (slot unidades))
```

Escribe un programa que calcule el importe total de la compra de un cliente, teniendo en cuenta que el supermercado tiene una promoción 3x2 para todos sus productos. El importe total se indicará afirmando un hecho.