

# PROGRAMACIÓN DECLARATIVA

Área personal / Cursos / (23666) PROGRAMACIÓN DECLARATIVA / Bloque 3  
/ Enunciado Laboratorio 8

## LABORATORIO 8 (Definición de funciones en el lenguaje Haskell y funciones de orden superior)

Para medir lo aprendido durante la Práctica 7, los alumnos deberán enviar los programas y soluciones a las preguntas que a continuación se indican. El envío se realizará en un archivo ZIP de nombre:

**gN\_Apellidos\_Lab8.ZIP**

donde N es el número del equipo de trabajo. El archivo ZIP contendrá los ficheros:

**Apellidos\_Lab8.hs** con el código Haskell de todos los programas solicitados.

**Apellidos\_Lab8.txt** con los datos personales de la persona que hace el envío y el enunciado y solución de aquellas preguntas que (no pudiéndose responder vía programa) se formulen en cada uno de los ejercicios del laboratorio.

En todos los casos, “**Apellidos**” son los apellidos del alumno que hace el envío.

### Ejercicio 30.

Definir las siguientes funciones sin usar las funciones predefinidas **ord** y **char**:

- a) El predicado **digit** que aplicado a un carácter nos dice si este es un dígito.
- b) Los predicados **uppercase** y **lowercase** que aplicados a un carácter nos dice si es una letra mayúscula o minúscula. (nota: entienda por letra los caracteres [a .. z] y [A .. Z] ).
- c) El predicado **letter** que aplicado a un carácter nos dice si es una

letra.

d) La función **charofdigit** que aplicada a un entero entre 0 .. 9 lo convierte en carácter.

### Ejercicio 31.

Como ejemplo de diseño de funciones recursivas, definir las siguientes funciones:

a) **among(x,y)**; La función among es un predicado. x es un objeto de tipo T, y es una lista formada por objetos de tipo T. among devuelve true

si x se encuentra como elemento de y, false en cualquier otro caso.

ej.: among(a,[a,b,c]) = among(a,[c,d,e,a]) = true; among(a,[b,c,d]) = false.

b) **stringcopy : Int -> String -> String**, que concatene un string un número entero de veces. Emplee parámetros formales para su definición.

c) En una segunda versión de la última función, denominada **strcop** utilizar ajuste de parámetros para su definición.

Estas funciones están definidas para números positivos, por lo que es parcial sobre el tipo de los enteros: Tratar los casos de indefinición utilizando la expresión **if** y la función **error**.

### Ejercicio 32.

Una carta puede caracterizarse por su valor y su palo. Utilizando la declaración data definir los tipos de datos: **valor**, **palo** y **carta**. Definir las funciones: **palode**: Carta -> Palo; y **valorde**: Carta -> Valor; . Y los predicados: **espalo**: Palo -> Carta -> Bool; y **esvalor**: Valor -> Carta -> Bool; .

### Ejercicio 33.

Un saco es como un conjunto, excepto que un elemento puede ocurrir mas de una vez. Un saco puede representarse mediante una lista de pares que consisten en un elemento y el número de veces que aparece el elemento en el saco. Escribe una definición polimórfica del tipo saco, junto con las funciones: **add**, que añade un elemento a un saco; **remove**, que devuelve un elemento arbitrario del saco junto con el saco al que se le ha quitado ese elemento; **isempty**, que devuelve true si un saco esta vacío.

### Ejercicio 34.

**a)** Definir las funciones **map** y **++** (concatenar dos listas) en términos de la función **reduce**. Llamad a las nuevas funciones **mapR** y **appR** respectivamente.

**b)** Comprobar que la función **among** puede definirse en términos de las funciones **lambda** y **reduce** de la siguiente forma:

**reduce (λnext -> (λisthea -> (if next == x then True else isthea))) False (y:ys)**

La lambda expresión que se pasa como argumento funcional no deja rastro en el interprete. ¿Que tendríamos que hacer si quisiéramos evaluarla?. ¿Explicar como actúa esta función?.

### **Ejercicio 35.**

Definir una función **uncurry** que tome una función de curry de tipo **alpha -> beta -> gamma** y la convierta en una función de tipo **(alpha, beta) -> gamma**, sin currificar.

Última modificación: lunes, 11 de mayo de 2015, 13:11:11