

Constructores de tipos predefinidos

Expresiones

Una expresión *booleana* es cualquier expresión de tipo Bool

Una expresión *aritmética* es cualquier expresión de algún tipo numérico: Int, Integer...

Tipos estructurados:

Además de los tipos simples, Haskell define *tipos estructurados* que permiten representar colecciones de objetos.

Tuplas

Una tupla es un dato compuesto donde el tipo de cada componente puede ser distinto. Los elementos de tipo tupla se escriben en Haskell separando con comas cada componente y encerrando todas entre paréntesis. Una tupla, como todo objeto en Haskell tiene un tipo asociado. El tipo se escribe separando los tipos de las distintas componentes con comas y todo entre paréntesis.

Tuplas

Si v_1, v_2, \dots, v_n son valores de tipo t_1, t_2, \dots, t_n entonces
 (v_1, v_2, \dots, v_n) es una tupla de tipo (t_1, t_2, \dots, t_n)

Probar en Haskell:

1. (4, 'a', "gato")
2. :type (4, 'a', "gato")
3. :type ('a', "zapato")
4. :type ("haskell", "matematica")
5. :type (True, 5)
6. :type ('?', "pepito ")
7. :type ("cualquiera", (3.14, False))

Probar en hugs:

- fst ("jota", (1, 'b'))
- snd ("jota", (1, 'b'))

Indicar el tipo de fst y snd y verificarlo en Haskell.

Observación: el tipo arreglos lo veremos más adelante, ya que tenemos que introducir algunos conceptos previos. Por lo tanto el siguiente tema es funciones e inmediatamente curriificación.