



## Práctica 1: Introducción a ML

1. ¿Cuál es la respuesta de ML a las siguientes expresiones?
  - a. `1+2*3`
  - b. `15.0-4.2/1.4`
  - c. `11 div 2 mod 3`
  - d. `"foo"~"bar"~""`
  - e. `3>4 orelse 5<6 andalso not(7<>8)`
  - f. `if 6<10 then 6.0 else 10.0`
2. Las siguientes expresiones en ML contienen errores. Indique cuál es el problema con cada una de ellas.
  - a. `8/4`
  - b. `if 2<3 then 4`
  - c. `1<2 and 5>3`
  - d. `6+7 DIV 2`
  - e. `4.+3.5`
  - f. `1.0<2.0 or 3>4`
  - g. `#"a"~#"b"`
3. Exprese:
  - a. `E orelse F`
  - b. `E andalso F`como expresiones `if-then-else`.
4. Escriba expresiones para realizar cada una de las siguientes conversiones.
  - a. Convierta 123.45 al más próximo menor entero
  - b. Convierta -123.45 al más próximo menor entero
  - c. Convierta 123.45 al más próximo mayor entero
  - d. Convierta -123.45 al más próximo mayor entero
  - e. Convierta al caracter Y en un entero
  - f. Convierta a 120 en un caracter
  - g. Convierta al caracter N en un real
  - h. Convierta 97.0 en un caracter
  - i. Convierta al caracter Z en una cadena de caracteres
5. Las siguientes expresiones tienen errores de tipo. ¿Cuáles son esos errores y cómo podrían corregirse?
  - a. `ceil(4)`
  - b. `if true then 5+6 else 7.0`
  - c. `chr(256)`
  - d. `chr(~1)`
  - e. `ord(3)`
  - f. `chr("#a")`

- g. `if 0 then 1 else 2`
- h. `ord("a")`

6. ¿Cuáles son los valores de las siguientes expresiones?

- a. `#2(3,4,5)`
- b. `hd([ 3, 4, 5])`
- c. `tl([3,4,5])`
- d. `explode("foo")`
- e. `implode(["f", # "o", # "o"])`
- f. `"c"::["a","t"]`
- g. `["c", "o"]@["b", "o", "l"]`
- h. `concat(["c", "a","t"])`

7. ¿Cuál es el error en las siguientes expresiones? Si lo cree posible, proponga una solución apropiada.

- a. `#4(3,4,5)`
- b. `hd([])`
- c. `#(1)`
- d. `explode(["bar"])`
- e. `implode("#a", # "b")`
- f. `["r"]::["a", "t"]`
- g. `tl([])`
- h. `1@2`
- i. `concat([#"a",#"b"])`

8. ¿Cuál es el tipo de las siguientes expresiones?

- a. `(1.5, ("3", [4,5]))`
- b. `[[1,2],nil, [3]]`
- c. `[(2,3.5), (4, 5.5), (6,7.5)]`
- d. `([#"a",#"b"], [nil, [1,2,3]])`

9. Responda las siguientes preguntas.

- a. ¿`(1,2)` y `(1,2,3)` tienen el mismo tipo?
- b. ¿`[1,2]` y `[1,2,3]` tienen el mismo tipo?

10. Dé ejemplos apropiados para cada uno de los siguientes tipos. No use la lista vacía (`nil` o `[]`) como valor.

- a. `int list list list`
- b. `(int * char) list`
- c. `string list * (int * (real * string)) * int`
- d. `((int * int) * (bool list) * real) * (real * string)`
- e. `(bool * int) * char`
- f. `real * int list list list list`

11. Usando dos de los operadores vistos en clase, resulta posible convertir un `string` de longitud uno a un `char`. Muestre cómo obtener esta transformación.