

```
--Comentarios de línea

--Definición de una función
--Cabecera (opcional)
--Implementación

--Tipos de datos
--Numéricos enteros:      Int Integer
--Numéricos reales:       Float Double
--Carácter:               Char
--Cadena de caracteres:   String
--Booleano:               Bool

--Debemos guardar el archivo con extensión .hs
--Definición de la función cuadrado
cuadrado::Int -> Int
cuadrado x = x^2

--Definición parcializada de la función menor
menor::Int -> Int -> Int
menor n1 n2 = if n1 < n2 then n1 else n2

--Definición no parcializada de la función mayor
mayor::(Int,Int) -> Int
mayor(n1,n2) = if n1 > n2 then n1 else n2

es_par::Int -> Bool
es_par x = mod x 2 == 0

es_par_string::Int -> String
es_par_string x = if mod x 2 == 0 then "par" else "impar"

(++)::Int -> Int
(++) x = x + 1

--f(x)=4*x^2 - 4*x + 25
--y=4*x^2 - 4*x + 25
--Para evaluar la función y=f(x) en x=raíz cuadrada de 9 + 2

evaluar::Int -> String
evaluar x | x>0 = "mayor que cero"
          | x<0 = "menor que cero"
          | x==0 = "igual que cero"

evaluar_v2 x | x>0 = "mayor que cero"
             | x<0 = "menor que cero"
             | otherwise = "igual que cero"

evaluar_funcion :: Int -> String
evaluar_funcion x | y > 0 = "mayor que cero"
                  | y < 0 = "menor que cero"
                  | y == 0 = "igual que cero"
                  where y=x^2 + 5*x + 0

--evaluar_funcion_v2 :: Int -> String
evaluar_funcion_v2 x | y > 0 = "mayor que cero"
                    | y < 0 = "menor que cero"
                    | otherwise = "igual que cero"
                    where y=x^2 + 5*x + 0

dia::Int->String
dia x = case x of
    0 -> "Domingo"
    1 -> "Lunes"
    2 -> "Martes"
    3 -> "Miercoles"
    4 -> "Jueves"
    5 -> "Viernes"
    6 -> "Sabado"
    otherwise -> "No definido"

menor_cuadrado::Int -> Int -> Int
menor_cuadrado n1 n2 = menor (cuadrado n1) (cuadrado n2)

promedio::Int -> Int -> Int -> Float
promedio n1 n2 n3 = (n1 + n2 + n3) / 3

--hay_nota_menor_4 n1 n2 n3 = n1<4 || n2<4 || n3<4
--hay_nota_menor_7 n1 n2 n3 = n1<7 || n2<7 || n3<7

hay_nota_menor_a::Int -> Int -> Int -> Int -> Bool
hay_nota_menor_a n1 n2 n3 ref = n1<ref || n2<ref || n3<ref

estado_academico::Int -> Int -> Int -> String
estado_academico n1 n2 n3 | hay_nota_menor_a n1 n2 n3 4 = "Libre"
                          | hay_nota_menor_a n1 n2 n3 7 || pr < 8 = "Regular"
                          | otherwise = "Promo"
                          where pr=promedio n1 n2 n3
```