Trabajo Práctico N1: Haskell

Alumno: Luciano Tomás Riente

Legajo: 208.480-6

Materia: Paradigmas de Programación

Comisión: K2005

Fecha de Entrega: 21/04/22



**CONSIGNA**

Sabemos que:

• La edad de una persona se calcula a partir de su nombre, como la cantidad de caracteres por 3; en caso de empezar con X o K, 5 años más y 10 en cualquier otro caso.

• Dos personas son compatibles cuando tienen una diferencia de edad de menos de 5 años y ambos son mayores de edad. Además, si la última letra de ambos nombres coincide, no son compatibles.

• Definir la función sonCompatibles que reciba dos nombres de persona y devuelva si es cierto o no. Mostrar ejemplos de consulta donde se vean algunas variantes

empiezaConXoK :: String -> Bool **{- Devuelve True si la persona empieza con X o K -}**

empiezaConXoK persona = head persona == 'K' || head persona == 'X'

edadPersona :: String -> Int **{- Calcula la edad en base a la cantidad de caracteres y si empieza o no con X o K-}**

edadPersona persona = 3 \* length persona + fromEnum (empiezaConXoK persona) \* 5 + fromEnum (not (empiezaConXoK persona)) \* 10

-- **fromEnum** sirve para convertir el resultado booleano en un valor entero (1 si es verdadero y 0 si es falso)

diferenciaEdadMenos5anios :: String -> String -> Bool **{- Devuelve True si la diferencia de edad es menor a 5 años -}**

diferenciaEdadMenos5anios persona1 persona2 = abs (edadPersona persona1 - edadPersona persona2) <= 5

mayoresDeEdad :: String -> String -> Bool **{- Devuelve True si ambas personas son mayores de edad -}**

mayoresDeEdad persona1 persona2 = (edadPersona persona1 >= 18) && (edadPersona persona2 >= 18)

ultimaLetraCoincidente :: String -> String -> Bool **{- Devuelve True si la última letra de ambos nombres coincide -}**

ultimaLetraCoincidente persona1 persona2 = last persona1 == last persona2

sonCompatibles :: String -> String -> Bool **{- Devuelve True si las dos personas son compatibles -}**

sonCompatibles persona1 persona2 = diferenciaEdadMenos5anios persona1 persona2 && mayoresDeEdad persona1 persona2 && not (ultimaLetraCoincidente persona1 persona2)

**OTRA ALTERNATIVA**

empiezaConXoK :: Char -> Int **{- Devuelve 5 si la letra es X o K, 10 en cualquier otro caso -}**

empiezaConXoK 'K' = 5

empiezaConXoK 'X' = 5

empiezaConXoK letra = 10

edadPersona :: String -> Int **{- Calcula la edad en base a la cantidad de caracteres y si empieza o no con X o K-}**

edadPersona persona = length persona \* 3 + empiezaConXoK (head persona)

diferenciaEdadMenos5anios :: String -> String -> Bool **{- Devuelve True si la diferencia de edad es menor a 5 años -}**

diferenciaEdadMenos5anios persona1 persona2 = abs (edadPersona persona1 - edadPersona persona2) <= 5

mayoresDeEdad :: String -> String -> Bool **{- Devuelve True si ambas personas son mayores de edad -}**

mayoresDeEdad persona1 persona2 = (edadPersona persona1 >= 18) && (edadPersona persona2 >= 18)

ultimaLetraCoincidente :: String -> String -> Bool **{- Devuelve True si la ultima letra de ambos nombres coincide -}**

ultimaLetraCoincidente persona1 persona2 = last persona1 == last persona2

sonCompatibles :: String -> String -> Bool **{- Devuelve True si las dos personas son compatibles -}**

sonCompatibles persona1 persona2 = diferenciaEdadMenos5anios persona1 persona2 && mayoresDeEdad persona1 persona2 && not (ultimaLetraCoincidente persona1 persona2)

Texto

Descripción generada automáticamente

**Ejemplos de consulta**