Materia:	1S7B-PROGR.LOG.FUNC
Nombre Informe:	Ejercicio: Kata FizzBuzz
Alumno(s):	Pizano Zamora Leonardo Gabriel

Objetivo

Marco Teórico

Desarrollo

Módulo y Importación

haskell

```
module Main where

import Data.List (intercalate)
```

module Main where: Define el módulo principal del programa.
import Data.List (intercalate): Importa la función intercalate del módulo Data.List, aunque no se utiliza en el código proporcionado.

Función fizzBuzz

haskell

```
fizzBuzz :: Int -> String
fizzBuzz n
    | n `mod` 15 == 0 = "FizzBuzz!"
    | n `mod` 5  == 0 = "Fizz!"
    | n `mod` 3  == 0 = "Buzz!"
    | otherwise    = numberToWords n
```

Propósito: Determina la cadena de texto correspondiente para un número dado según las reglas de FizzBuzz.
Entrada: Un número entero (Int).
Salida: Una cadena de texto (String):
"FizzBuzz!" si el número es múltiplo de 15.

"Fizz!" si el número es múltiplo de 5.
"Buzz!" si el número es múltiplo de 3.
La representación en palabras del número si no es múltiplo de 3 ni de 5 (usando numberToWords).

Función numberToWords

haskell

```
numberToWords :: Int -> String
numberToWords n
                        | n == 0        = "Zero"
                        | n < 0         = "Negative " ++ convert (-n)
                        | otherwise = convert n
  where
                        units = ["", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine"]
                        teens = ["Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen",
"Eighteen", "Nineteen"]
                        tens = ["", "", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"]
                        convert n
                        | n < 10        = units !! n
                        | n < 20        = teens !! (n - 10)
                        | n < 100   = let (t, u) = n `divMod` 10
                        in tens !! t ++ if u /= 0 then " " ++ units !! u else ""
                        | n == 100  = "One Hundred"
                        | otherwise = "Number out of range"
```

Propósito: Convierte un número entero en su representación en palabras en inglés.
Entrada: Un número entero (Int).
Salida: Una cadena de texto (String) que representa el número en palabras.
Detalles:
Define listas de cadenas para unidades (units), números entre 10 y 19 (teens) y decenas (tens).
La función interna convert maneja la conversión según el rango del número:
Para números menores que 10, utiliza la lista units.
Para números entre 10 y 19, utiliza la lista teens.
Para números entre 20 y 99, divide el número en decenas y unidades, y utiliza las listas tens y units para construir la cadena.
Para el número 100, devuelve "One Hundred".
Para números fuera del rango 0-100, devuelve "Number out of range".

Función main

haskell

```
main :: IO ()
main = do
                        mapM_ (putStrLn . fizzBuzz) [0..100]
```

Propósito: Ejecuta el programa FizzBuzz para los números del 0 al 100 y muestra los resultados en la consola.

Detalles:
Utiliza mapM_ para aplicar la función fizzBuzz a cada número en el rango [0..100].
putStrLn se usa para imprimir cada resultado en una nueva línea en la consola.

Resultado

[?2004l
FizzBuzz!
One
Two
Buzz!
Four
Fizz!
Buzz!
Seven
Eight
Buzz!
Fizz!
Eleven
Buzz!
Thirteen
Fourteen
FizzBuzz!
Sixteen
Seventeen
Buzz!
Nineteen
Fizz!
Buzz!
Twenty Two
Twenty Three
Buzz!
Fizz!
Twenty Six
Buzz!
Twenty Eight
Twenty Nine
FizzBuzz!
Thirty One
Thirty Two
Buzz!
Thirty Four

Fizz!
Buzz!
Thirty Seven
Thirty Eight
Buzz!
Fizz!
Forty One
Buzz!
Forty Three
Forty Four
FizzBuzz!
Forty Six
Forty Seven
Buzz!
Forty Nine
Fizz!
Buzz!
Fifty Two
Fifty Three
Buzz!
Fizz!
Fifty Six
Buzz!
Fifty Eight
Fifty Nine
FizzBuzz!
Sixty One
Sixty Two
Buzz!
Sixty Four
Fizz!
Buzz!
Sixty Seven
Sixty Eight
Buzz!
Fizz!
Seventy One
Buzz!
Seventy Three
Seventy Four
FizzBuzz!
Seventy Six
Seventy Seven
Buzz!
Seventy Nine
Fizz!
Buzz!
Eighty Two

Eighty Three
Buzz!
Fizz!
Eighty Six
Buzz!
Eighty Eight
Eighty Nine
FizzBuzz!
Ninety One
Ninety Two
Buzz!
Ninety Four
Fizz!
Buzz!
Ninety Seven
Ninety Eight
Buzz!
Fizz!
[?2004h

Evaluación

Conclusiones}