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

Objetivo

Marco Teórico

Desarrollo

1. Función fizzBuzz

fizzBuzz :: Int -> String

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 primo.
La representación en palabras del número en español si no es primo (usando numberToWords).

2. Función isPrime

isPrime :: Int -> Bool

Propósito: Verifica si un número es primo.
Entrada: Un número entero (Int).
Salida: Un booleano (Bool) indicando si el número es primo o no.
Detalles:
Devuelve False si el número es 1.
Utiliza una lista de números desde 2 hasta la raíz cuadrada del número para verificar si algún número en este rango divide al número dado sin dejar residuo.

3. Función numberToWords

numberToWords :: Int -> String

Propósito: Convierte un número entero en su representación en palabras en español.
Entrada: Un número entero (Int).
Salida: Una cadena de texto (String) que representa el número en palabras.
Detalles:
Define listas de cadenas para unidades, "teens" (10-19), decenas, centenas y miles.
La función convert maneja la conversión según el rango del número.
La función thousands maneja la conversión para números entre 1000 y 999999.

4. Función main

main :: IO ()

Propósito: Ejecuta el programa FizzBuzz para un número dado por el usuario.
Detalles:
Pide al usuario que introduzca un número.
Lee la entrada del usuario.
Convierte la entrada a un número entero.
Si la conversión tiene éxito y el número está en el rango 0-1000000, llama a fizzBuzz y muestra el resultado.
Si la conversión falla o el número está fuera del rango, muestra un mensaje de error adecuado.

Al ejecutar las pruebas, deberías ver una salida similar a esta en la consola:

makefile

```
Cases: 4  Tried: 4  Errors: 0  Failures: 0
Cases: 4  Tried: 4  Errors: 0  Failures: 0
Cases: 5  Tried: 5  Errors: 0  Failures: 0
```

Cada línea corresponde a la ejecución de uno de los grupos de pruebas:

Pruebas de la función fizzBuzz:
Verifica si fizzBuzz devuelve las cadenas correctas para números específicos.
Casos probados: 1, 2, 178, 1000000.

Pruebas de la función isPrime:
Verifica si isPrime identifica correctamente si un número es primo.
Casos probados: 2, 3, 4, 13.

Pruebas de la función numberToWords:
Verifica si numberToWords convierte correctamente números a palabras en español.
Casos probados: 0, 15, 21, 100, 999.

Análisis Detallado de la Salida

Cases: Número de casos de prueba definidos.
Tried: Número de casos de prueba ejecutados.

Errors: Número de errores ocurridos durante las pruebas (por ejemplo, si una prueba no se pudo ejecutar por algún problema interno).
Failures: Número de fallos (cuando el resultado de la función no coincide con el esperado).

Si todos los valores de Errors y Failures son 0, significa que todas las pruebas pasaron correctamente y las funciones funcionan como se espera.
Resultado Esperado para Cada Prueba

Pruebas de fizzBuzz:
fizzBuzz 1 debe devolver "uno".
fizzBuzz 2 debe devolver "FizzBuzz!".
fizzBuzz 178 debe devolver "ciento setenta y ocho".
fizzBuzz 1000000 debe devolver "número fuera de rango".

Pruebas de isPrime:
isPrime 2 debe devolver True.
isPrime 3 debe devolver True.
isPrime 4 debe devolver False.
isPrime 13 debe devolver True.

Pruebas de numberToWords:
numberToWords 0 debe devolver "cero".
numberToWords 15 debe devolver "quince".
numberToWords 21 debe devolver "veintiuno".
numberToWords 100 debe devolver "cien".
numberToWords 999 debe devolver "novecientos noventa y nueve".

Evaluación

Conclusiones}