UNIVERSIDAD TECNOLOGICA CENTROAMERICANA

UNITEC

LENGUAJES DE PROGRAMACION

DOC. KENNY MAURICIO DAVILA CASTELLANOS

MINI PROYECTO 2

INTEGRANTES:

STEPHANIE MARIELA MARTÍNEZ HANDAL

11741368

TIFFANNY ALEXA VARELA BANEGAS

11811146

19 DE SEPTIEMBRE DE 2021

Contenido

I. INTRODUCCIÓN

"La complejidad es destructiva. Chupa la sangre de los desarrolladores, hace que los productos sean difíciles de planificar, construir y probar, introduce problemas de seguridad y provoca la frustración de usuarios finales y administradores"

-- Ray Ozzie

A fin de la creación utilizando un lenguaje de programación funcional, se nos ha dado la tarea de hacer un programa en este mismo dando como objetivos el aprender, además de un nuevo lenguaje, una nueva forma de programar a la que no muchos estamos acostumbrados. Este mismo, nos ha enseñado una forma muy diferente en el uso de la programación y el cómo esta misma va cambiando tanto en el tiempo como de lenguaje y se usuario.

II. DESCRIPCIÓN DEL JUEGO

Se presenta la creación de un juego en lenguaje funcional en el cual un usuario elije su nivel de dificultad entres fácil, intermedio y difícil; después de seleccionarlo, y que cada uno de estos niveles conlleva una cantidad de dígitos diferentes, se genera un numero al azar, el cual deberá adivinar, y se le pide al jugador que ingrese una secuencia la cual se compara con la secuencia generada y se regresa una serie de caracteres dependiendo de cuantas acertó y cuantas no (“+” cuando acertó número y lugar, “-” acertó digito pero no lugar y “X” no se contiene dicho digito).

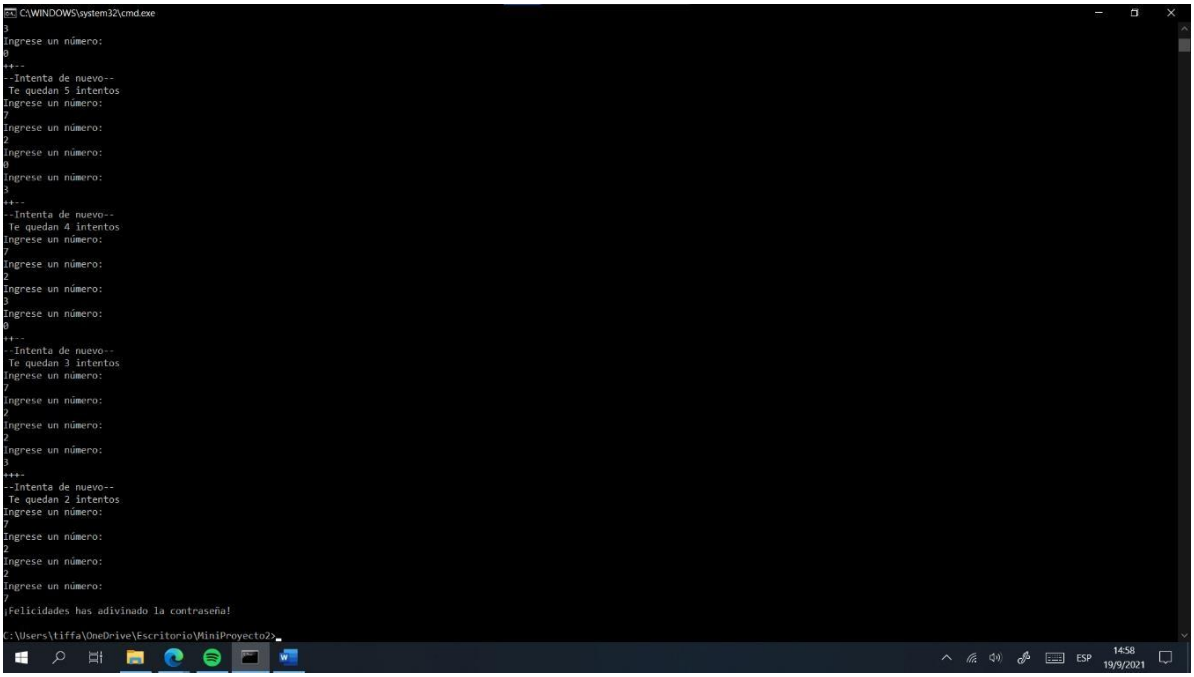
Después de todo esto, se le dará al usuario una cantidad definida de intentos para adivinar la combinación.

III. ARQUITECTURA DEL PROGRAMA

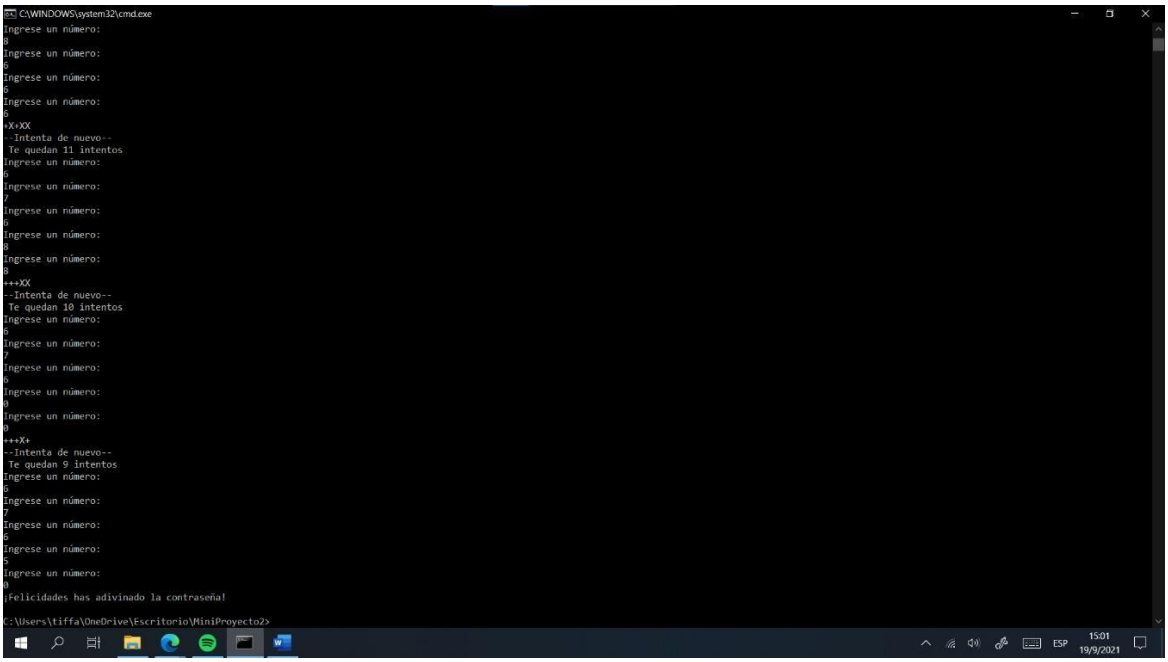
- **Main**: contiene la primera parte del programa, los llamados a las demás funciones y el llamado a la segunda parte.
- **RandomList**: creación de la lista con numero random.
- **ArregloUsuario**: crea el arreglo con los dígitos ingresados uno por uno de parte del usuario.
- **TamArreglo**: obtiene el nivel y devuelve de que tamaño será la lista.
- **NumMax**: número máximo en el rango para creación de los aleatorios.
- **NumIntentos**: número de intentos que tendrá el usuario dependiendo de la dificultad.
- **EvaluacionRetroalimentacion**: función que retorna el carácter luego de comparar el numero ingresado por usuario como el generado en el programa.
- **RetroAlimentacion**: recorre el arreglo para su evaluación.
- **Continuacion**: segunda parte del main la cual revisa si la combinación ha sido adivinada o no.

IV. EJEMPLOS

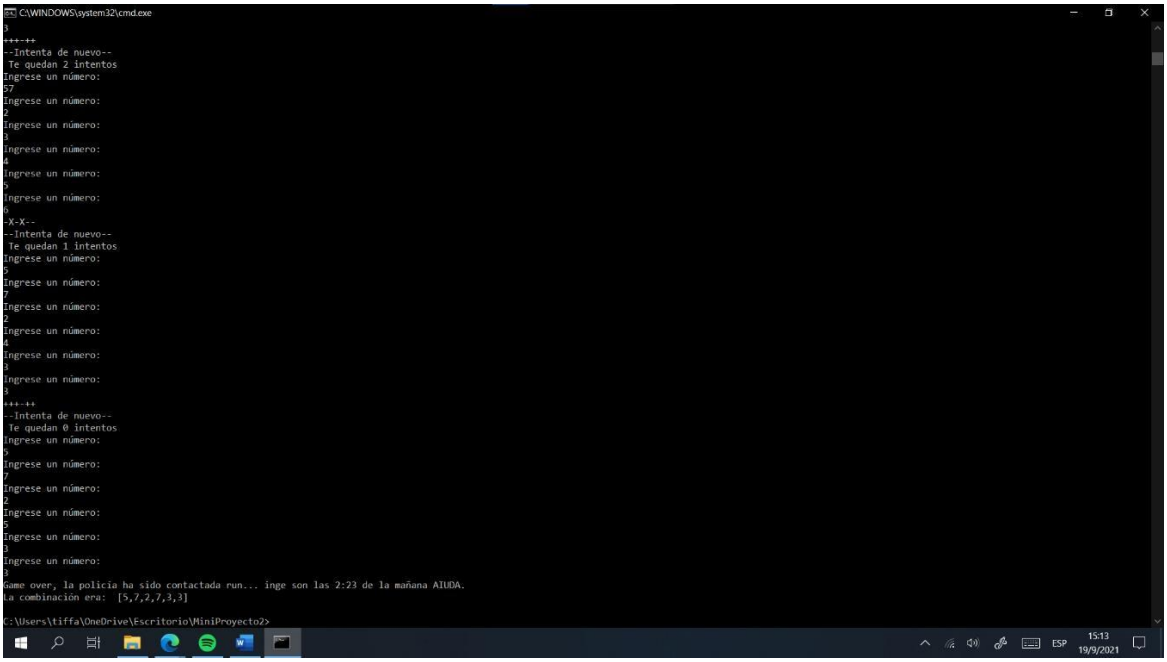
i. Fácil



ii. Intermedio



iii. Difícil


C:\WINDOWS\system32\cmd.exe
3
+++-++
--Intenta de nuevo--
Te quedan 2 intentos
Ingrese un número:
57
Ingrese un número:
2
Ingrese un número:
3
Ingrese un número:
4
Ingrese un número:
5
Ingrese un número:
6
-X-X--
--Intenta de nuevo--
Te quedan 1 intentos
Ingrese un número:
5
Ingrese un número:
7
Ingrese un número:
2
Ingrese un número:
4
Ingrese un número:
3
Ingrese un número:
3
+++-++
--Intenta de nuevo--
Te quedan 0 intentos
Ingrese un número:
5
Ingrese un número:
7
Ingrese un número:
2
Ingrese un número:
5
Ingrese un número:
3
Ingrese un número:
3
Game over, la policia ha sido contactada run... inge son las 2:23 de la mañana AIUDA.
La combinación era: [5,7,2,7,3,3]
C:\Users\tiffa\OneDrive\Escritorio\MiniProyecto2>
ESP
15:13
19/9/2021

V. ANÁLISIS

1. ¿Como compararía implementar un juego de este tipo usando Haskell vs hacer una versión en un lenguaje orientado a objetos?

   Por la diferencia ente la POO y la programación funcional, más, la poca costumbre e información clara sobre la segunda compartimos la opinión que será mucho más fácil con un lenguaje orientado a objetos en comparación de la funcional.

2. ¿Hubo partes del juego que son más fáciles de implementar en Haskell que usando lenguajes como Java?

   En general la mayor parte del programa fue más complicado de implementar en Haskell, lo único que fue más fácil de implementar fueron las funciones que hacían uso de los guards que eran similares a tener varios if's anidados.

3. Si tuviese más tiempo, ¿qué otras características le adicionarían al juego?

   Revisar si encontramos una forma fácil de implementarlo de forma gráfica.

4. Cualquier otro comentario u observación relevante.

   Programación funcional es demasiado complicada comparado con lo sencillo que es la POO en muchos aspectos.

VI. DIFICULTADES ENCONTRADAS

- Una nueva forma de programar la cual no conocíamos
- Documentación encontrada no muy entendible
- Métodos que necesitábamos se debían de hacer de una forma distinta
- La instalación de librerías por separado que se pensarían son básicas

VII. CONCLUSIONES

- La programación funcional es muy diferente a la programación imperativa, que es con la que hemos estado trabajando. La programación funcional puede presentar muchas ventajas pero en muchos casos es más conveniente usar lenguajes de programación imperativa.
- Para la programación funcional se necesita ser ordenado y crear funciones para cosas que en la programación orientada a objetos se hacen casi de forma automática.