

## **Informe Final Programación Declarativa**

*Integrantes:*

- *Oliva Benjamin*
- *Ameri Francisco*

*Asignatura:*

- *Programacion Declarativa*

*Docente:*

- *Maximiliano Andres Echoyez*

*Fecha:*

- *04/03/2022*

**Proyecto:** *Juego del Uno*

## **Resumen del proyecto**

El proyecto consiste en el juego “Uno” donde se reparten 7 cartas para cada jugador (2 jugadores mínimo - 4 jugadores máximo), donde cada uno puede poseer distintas cartas de diferentes colores además de cartas especiales.

El juego fue desarrollado dos veces, la primera en el lenguaje Python usando listas y tuplas utilizando programación funcional.

## **Consigna**

El objetivo de este Trabajo Final es desarrollar un programa que permita al usuario jugar un juego de cartas que implique cartas en mano, en mesa y en el mazo, tanto para naipes españoles como ingleses.

Se deberán considerar los siguientes puntos:

- El programa debe ser interactivo con el usuario.
- La interfaz de usuario será en modo texto.
- Deberá estructurarse en módulos según las funcionalidades.
- Deberá contener tipos de datos definidos específicamente para resolver la problemática. El juego deberá implementarse en los lenguajes de programación Haskell y Python. Si bien Python es multiparadigma, se deberá realizar dentro del paradigma de Programación Funcional. Al finalizar ambas versiones, se debe realizar una comparación de similitudes y diferencias entre ambos lenguajes.

## **Solución**

### **Python:**

Fue en el primer lenguaje que desarrollamos el juego ya que conocíamos este lenguaje desde antes y tenemos un mejor manejo de la sintaxis. Al ser un lenguaje que no se limita al paradigma funcional (como lo es haskell) nos fue más fácil desarrollar el programa y luego reescribir las funciones para cumplir la condición de desarrollo en paradigma funcional.

Se generaron funciones que devuelven diferentes estructuras de datos, jugador, baraja, pozo, colores, carta, cartas especiales, entre otros; y otros métodos para permitir la jugabilidad como lo son: generar mazo, cumplir reglas, escoger color, puntuar. También se utilizó algunas librerías externas para como por ejemplo hacer la mezcla de la baraja (`rd.shuffle()`).

El jugador en su turno tiene dos posibilidades, o tirar una carta de su mano que coincida (el color de la carta a tirar con la carta en el pozo o el valor de la carta a tirar con el valor de la carta del pozo), si tirar una carta que cumpla y sino deberá robar una carta del mazo. El juego termina cuando un jugador se queda sin cartas en mano.

### **Haskell:**

Se crearon dos módulos, mazo y jugador con sus respectivos data types, diccionarios, listas y funciones. Intentamos aplicar recursividad en todas las funciones para no hacerlo imperativo y respetar la consigna de programación funcional. Nos faltó terminar el circuito completo del juego.

## **Conclusion**

El problema que podemos mencionar en Python es que al ser multiparadigma, muchas veces nos vimos tentados a dejar soluciones del paradigma procedural, que nos alejaba de nuestro objetivo principal. Por esta razón hemos corregido constantemente el código para cumplir con la consigna usando los conocimientos adquiridos en la asignatura.

En cuanto a la solución en Haskell, si bien logramos hacer las estructuras principales y algunos métodos, la dificultad de la forma de juego se vio muy dificultosa para lograr el objetivo en este lenguaje.

**REPO:** [https://github.com/FranAmeri99/PD\\_FINAL\\_OLIVA\\_AMERI](https://github.com/FranAmeri99/PD_FINAL_OLIVA_AMERI)