

Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-3661 Laboratorio de Lenguajes de Programación I
Trimestre Septiembre - Diciembre 2018

INFORME
PROYECTO III

Integrantes:

Aurivan Castro 14-10205

Sandra Vera 14-11130

Sartenejas, Diciembre de 2018

INTRODUCCIÓN

De entre los muchos juegos que se han popularizado por medio de series de televisión, es notable mencionar “Piedra, Papel, Tijeras, Lagarto, Spock”, que fue llevado a la fama por la serie “The Big Bang Theory”. El mismo consiste en una modificación el juego tradicional “Piedra, Papel o Tijeras” añadiendo más elementos. El enfoque de este proyecto es realizar una implementación de dicho juego por medio de clases específicas pedidas en el enunciado usando Ruby como lenguaje de programación.

En las siguientes partes del informe se presentarán, en orden: las maneras de generar la documentación y la forma de jugar por medio del programa; una breve descripción de la organización del programa, así como una explicación breve de las decisiones de diseño, y por último las conclusiones, con recomendaciones para proyectos parecidos al presentado en este trabajo.

DE LA INTERACCIÓN CON EL PROGRAMA

La documentación

Para generar la documentación, es necesario correr el siguiente comando desde la carpeta que contiene el proyecto en la terminal:

```
rdoc Jugadas.rb Estrategia.rb Juego.rb
```

Esto generará una carpeta en el mismo directorio llamada “doc”, con toda la documentación de las diferentes clases del proyecto y sus métodos.

La ejecución del programa

El cliente del juego se ejecuta al correr el siguiente comando en la terminal desde la carpeta que contiene el proyecto:

```
ruby Juego.rb
```

En primer lugar, se pide que se ingresen los nombres de cada jugador, así como la estrategia de cada uno con su configuración si es necesaria. Estos nombres y estrategias no pueden cambiarse durante una partida, para hacerlo se debe salir del juego y volver a ejecutar el cliente. Una vez especificados los jugadores, se muestran las siguientes opciones:

- Jugar por rondas: permite jugar un número de rondas especificado, generando errores si se especifica un número no positivo.

- Jugar por puntos: permite jugar hasta alcanzar un número de puntos especificado, generando errores si se especifica un número no positivo.
- Reiniciar la partida: reinicia la partida y las estrategias de cada jugador.
- Mostrar jugadores con sus estrategias: muestra en consola los nombres de los jugadores junto con sus estrategias y configuración asociada.
- Salir del juego

DE LA IMPLEMENTACIÓN

En este punto, es importante mencionar que el enunciado del proyecto fue bastante claro en las partes del programa, clases, subclases, métodos, entre otros, por lo cual el equipo de trabajo decidió apegarse lo más posible a lo indicado en el mismo. Sin embargo, existen detalles que son importantes de la implementación, y que serán indicados separadamente a continuación.

De la clase Estrategia

Se asume que el parámetro *j* de las funciones prox de las subclases de Estrategia corresponde a la jugada de la ronda anterior del contrincante. En el caso de la subclase Pensar, la primera jugada siempre es Piedra por la manera en la que se toman las decisiones de esta estrategia que se definió en el enunciado del proyecto.

De la clase Partida

En la clase Partida en el método rondas, cuando se están empezando las rondas, si la estrategia es Copiar o Pensar, se pasa como argumento en *j* la jugada que está guardada en el atributo de instancia primera, pues ahí está definida la primera jugada a realizar.

Además, por la manera en la que se guardan las jugadas anteriores en Partida, se pasan a las estrategias para que calculen su próxima jugada una vez se haya realizado otro movimiento, por lo que para la estrategia Pensar no se registra la última jugada al llamar al método rondas o alcanzar. Sin embargo, si por ejemplo se llaman en orden a los métodos rondas y alcanzar, se registra la última jugada de rondas, más no la de alcanzar.

De la interfaz gráfica

Algo que es importante señalar de este proyecto es que *no* hay interfaz gráfica, y esto se debe a la *imposibilidad* de conectar la terminal con la interfaz creada por medio de *Ruby Shoes*. El equipo de trabajo intentó por varias formas, pero finalmente, decidió no continuar con dicha tarea, y por cuestiones de tiempo, se ha dejado sin implementar.

Otros detalles de implementación pueden encontrarse en la documentación del programa.

CONCLUSIONES

El presente proyecto tenía de objetivo la creación de un programa que permitiera jugar el juego “Piedra, Papel, Tijeras, Lagarto, Spock” por medio de clases, métodos y otros detalles, indicados en el enunciado del mismo. El equipo de trabajo se encargó de realizar cada una de las clases de manera que cumplieran lo indicado en el enunciado, así como un cliente que permitiera la interacción con el usuario.

De entre las dificultades encontradas, la más problemática y la que tomó más tiempo y no permitió la completitud del proyecto, fue el uso de Ruby Shoes para construir la interfaz del programa, debido a que, como se mencionó anteriormente, no se halló una manera de conectar la interfaz con la terminal en el tiempo dado para la realización del proyecto, sobre todo en las semanas ajetreadas para el equipo de trabajo.

Para futuros proyectos se recomienda la utilización de una librería diferente a Shoes que realmente facilite la realización del proyecto en el tiempo estipulado, y que la misma haya sido probada con anticipación por el personal correspondiente.