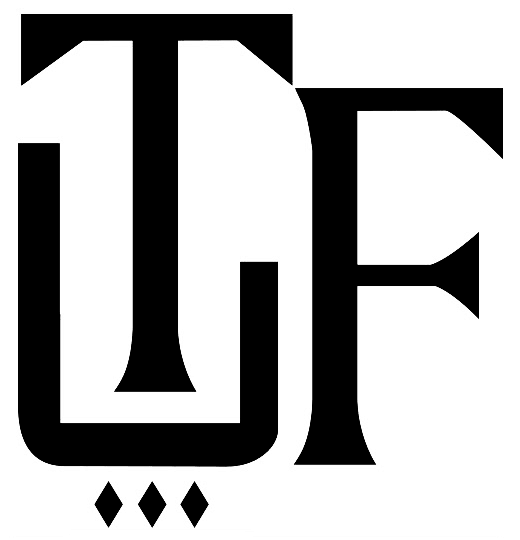
***TRABAJO PRÁCTICO Nº1***

***UNIVERSIDAD NACIONAL DE TRES DE FEBRERO***

***ESTRUCTURAS DE DATOS***

***13/11/2017***

******

**INTEGRANTES:**

Andreoli, Maximiliano

De Alesandro, Franco

Ramos, Andrés

***Decisiones de Diseño:***

El programa está dividido en cuatro archivos que interactúan entre sí.

Por un lado, tenemos la *consola* (*consola.py*). Esta, se ocupa de garantizar interacción entre el usuario y el programa mediante el despliegue de un menú cuyas opciones se derivan de la funcionalidad aportada por los otros tres archivos restantes. A su vez, la consola interactúa con el módulo de *persistencia* (*persistencia.py*) para permitirle al usuario guardar una partida en disco y poder retomarla más tarde.

Tanto el modo de juego normal, como el de vidas estáticas se valen del módulo de tableros (*tablero.py*) para funcionar. Los tableros poseen un constructor que es quien genera la matriz correspondiente según el modo de juego ingresado, así como también poseen un método "\_\_eq\_\_" que hace las veces del "compare\_to" en java, es decir, le permite a cada tablero poder compararse con aquél que lo preceda. También se compone de un método para actualizar las células del tablero según las reglas del juego especificadas en la presentación de este trabajo. Tanto las células como los modos de juego se interpretan dentro del programa mediante el uso de Enumerables.

En el modo vidas estáticas se utiliza el archivo combinations.py para poder obtener la cantidad de combinaciones posibles para una "n" cantidad de células vivas en un tablero de "i" dimensiones.

Para detectar los patrones osciladores (ya sean de nivel 1 o de nivel 2), implementamos un esquema que consiste en dos atributos booleanos: *finalizo* y *estatico*. Dichos atributos se inicializan en el constructor y luego, su valor se actualiza con cada una de las iteraciones llevadas a cabo en el método actualizar\_celulas. Cada vez que una iteración se completa exitosamente el atributo *finalizo* pasa a ser verdadero. Por su parte, el atributo *estatico* certifica que la distribución de células vivas de la generación anterior (patrón oscilador de orden 1) o de la precedente a esta última (patrón oscilador de orden 2) sean diferentes a la de la generación actual; de no serlo, el atributo estático será verdadero y el programa habrá detectado un patrón oscilador.

No está de más aclarar que se utilizó un esquema de excepciones tipificadas en el archivo *exepciones.py*.

Finalmente, en lo que respecta a la persistencia, se utilizó la librería *shelve*. Para el diccionario shelve, se usó como clave el nombre de la partida ingresado por el usuario mientras que su valor, queda definido por una instancia de la clase tablero (que es la partida que la persona estaba jugando).

***Instrucciones de uso:***

El menú consiste en tres submenús a saber: "Nuevo juego", "Cargar Partida" y "Salir". Al seleccionar "Nuevo juego", el programa solicitará el ingreso de un número de filas y de columnas que deberá ingresarse con el formato "filaxcolumna". La cantidad mínima de filas y columnas que se pueden ingresar es igual a 3 unidades mientras que la máxima para la primera es de 30 unidades y 60 unidades respectivamente para la segunda. Luego de haber ingresado las filas y las columnas, el programa consultará al usuario por el modo de juego en el que desea jugar (modo normal, paso a paso o vida estática).

*Modo normal:* en este modo se deberá indicar el método mediante el cual se asignarán las células vivas del tablero. En método de generación manual, el usuario ingresa las coordenadas de las células vivas en formato "filaxcolumna" mientras que, en el método de generación aleatorio, la ubicación de las células vivas se determina al azar. Luego de haber seleccionado el método de generación, el tablero comenzará a iterarse indefinidamente y solo se detendrá cuando se encuentre un patrón oscilador (si hay vida estática), cuando todas las células vivas hayan muerto, cuando el usuario guarde el tablero o cuando este cierre el programa.

*Modo paso a paso:* este modo funciona exactamente igual al modo normal con la salvedad de luego de cada iteración el programa se detiene y le pregunta al usuario si desea seguir (además de permitirle guardar la partida).

*Modo vidas estáticas:* bajo esta modalidad solo se permite un ingreso manual de las células vivas. Luego de haberse ingresado dichas células, el tablero comienza a iterarse hasta encontrar un patrón de vida estática o hasta alcanzar unas 30 iteraciones. Durante la ejecución, el programa le irá mostrando al usuario al usuario cada uno de los tableros que se generan y finalmente le indicara, en caso de que se haya detectado, que hay un patrón de vida estática.