

# Algoritmos y Programación I

## 75.40

### Trabajo práctico N°1

Segundo cuatrimestre 2016

Federico del Mazo

Legajo: 100029

Práctica: Alan

Corrector: Agustín Castaño

## Diseño y flujo del programa

A grandes rasgos el programa se basa en una función **main**, que llama a la función **menu**, y este menú que llama a jugar los niveles en alguno de los dos **modos**, aplicando la función de **juego**. Cualquier otra función del programa son externas y dependen de este núcleo, siendo llamadas solo por este.



### Main:

- Saluda al usuario la primera vez que entra al juego
- Ejecuta el menú

### Menu:

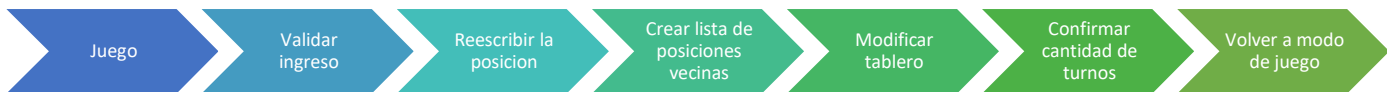
- Permite al usuario salir del juego
- Permite al usuario leer las instrucciones enteras
- Le pregunta al usuario qué modo de juego quiere jugar, si el "Random" o el "Normal"
- Si el usuario no ingresa ninguna de las opciones dadas, se le pedirá que ingrese de nuevo

### Modos de juego:

Las funciones de los modos de juego son las encargadas de comprobar si el nivel que se está jugando se ganó o perdió y de avisar la cantidad de puntos cuando sea necesario. Esto decidí hacerlo con un simple ciclo definido, donde si el usuario perdió el nivel, simplemente se rompe el ciclo y se vuelve al menu, pero si el usuario gana el nivel actual, sigue con el siguiente de la lista de niveles. Una vez terminados los 5 niveles, vuelve al menú. Para el puntaje, elegí usar un acumulador de los puntos totales al que se le vayan sumando los puntos ganados en el nivel actual.

El modo normal va a hacer uso del import de Python e ira a buscar los niveles a otro archivo, así si, eventualmente, se quieren hacer cambios en los 5 niveles predeterminados, no hay que modificar el archivo del juego principal.

El modo random es más complicado que el modo normal, por el asunto de tener que crear los niveles completamente al azar. Lo primero que tiene que hacer es pedirle al usuario que elija la dimensión del tablero que quiera, para luego jugar 5 niveles de esa dimensión, en un tablero cuadrado. Luego, con esa dimensión, va a ir a otra función, a crear un tablero con luces prendidas/apagadas al azar. Esta operación decidí dividirla en dos funciones para mayor comprensión. La primera función crea un tablero de todas luces apagadas. La segunda función es la encargada de, con la ayuda del módulo random, prender luces al azar. Para esto elegí un numero arbitrario de los límites de cuantas luces prender por nivel. Elegí entre 2 veces la dimensión y la dimensión al cuadrado menos ella misma, dejando un número que, a mi parecer, es bastante razonable. Por ejemplo, para un tablero de 5x5 prende entre 10 y 20 luces, mientras que para uno de 10x10 prende entre 20 y 90.



### **Función Juego:**

Esta es la función más “pesada” de todo el programa, y será la que más ayuda de funciones externas necesite. La función va a recibir un nivel que le pasará el modo de juego, y va a ir modificándolo prendiendo y apagando las luces, también va a tener la variable de los turnos y la de los puntos del nivel actual, para que estos se le sumen al puntaje total.

En un ciclo indefinido que depende de la variable de turnos, lo primero que hace la función de juego va a ser imprimir el tablero de forma que sea amigable al usuario, es decir, con filas y columnas de puntos, en vez de con listas de unos y ceros. Luego, la función le pregunta al usuario qué luces quiere prender/apagar o si quiere volver al menú, a las instrucciones, resetear el tablero, o ver el estado

actual del nivel. Para esto va a ir a otra función, una que valide lo que el usuario introdujo.

Una vez que se valide que el usuario introdujo una posición correcta que esté dentro del tablero, esta segunda función entrará a una tercera función, una que reescriba la posición que introdujo el usuario de forma que el programa pueda leerla. Para esto decidí convertir la posición a una lista de dos números, que refieren a la fila y la columna del nivel. Estos números tienen que ser uno menos de la posición que introdujo el usuario porque el programa está escrito en filas y columnas del 0 al 9 y no del 1 al 10 (por convención de Python). Entonces, por ejemplo, la posición D3, el programa la “convertirá” a la lista [3, 2], y así.

Una vez que el programa tenga la lista de la posición, tendrá que ir a una cuarta función que sea la que cree una lista de luces que sean las luces vecinas de la selecta. Y luego, finalmente, la función juego utilizará de una quinta función que sea la que modifique el nivel cambiando el estado de las luces vecinas de prendidas a apagadas y viceversa, siempre fijándose que no haya entrado en una luz que esté afuera del tablero, como sería el caso de los bordes del nivel.

Terminadas de prenderse/apagarse las 5 luces, el acumulador de turno bajará una unidad, la función juego confirmará que sigue estando dentro del ciclo indefinido, y finalmente, devolverá dos valores al modo de juego, uno booleano indicando si ganó o perdió, y un número que sea los puntos ganados o perdidos en el nivel.

Una última función externa del núcleo principal del programa, es la de confirmar cuántas luces prendidas quedan en el tablero. Esta función tiene dos utilidades, saber si el usuario ganó el nivel (cuando el acumulador de luces prendidas sea igual a 0) y el de calcular los puntos perdidos en la función de resetear el nivel a su estado original (que varía dependiendo de cuántas luces reseteó).

Por último, otros módulos de los que hice uso fueron el módulo time, para su función sleep y hacerle esperar al usuario un segundo antes de pasar al siguiente nivel, el módulo copy, para que si el usuario resetea el tablero el nivel vuelva al original (copiando al principio de la función juego el nivel actual a una nueva variable) y el módulo sys, con su función exit para cuando el usuario quiera salir

del juego enteramente. Todos los módulos los importé haciendo uso de el "from", para que el programa procese menos y sepa directamente qué función importar, en vez de buscar dentro de todas las funciones del módulo. También para mejor comprensión del código.