Ruben Cuadra

Manual

22 August 2018

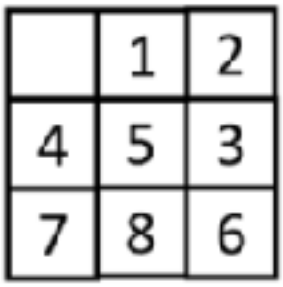
Proyecto 1 - Puzzle con DFS y BFS

El proyecto consta de 2 archivos:

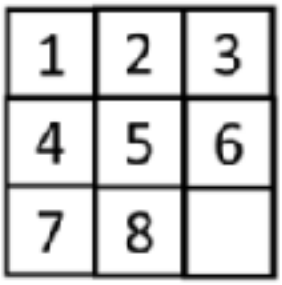
- Proyecto1.py: Core del programa, contiene las clases *PuzzleNode, Puzzle* y la función *busquedaNoInformada*
- main.py: Contiene un ejemplo para correr el programa

El proyecto funciona sin necesidad de instalar alguna librería externa siempre y cuando se tenga la version de Python 3.x instalada (Probado con Python 3.6.5)

El ejemplo se basa en estos estados:



Estado Inicial



Estado Final

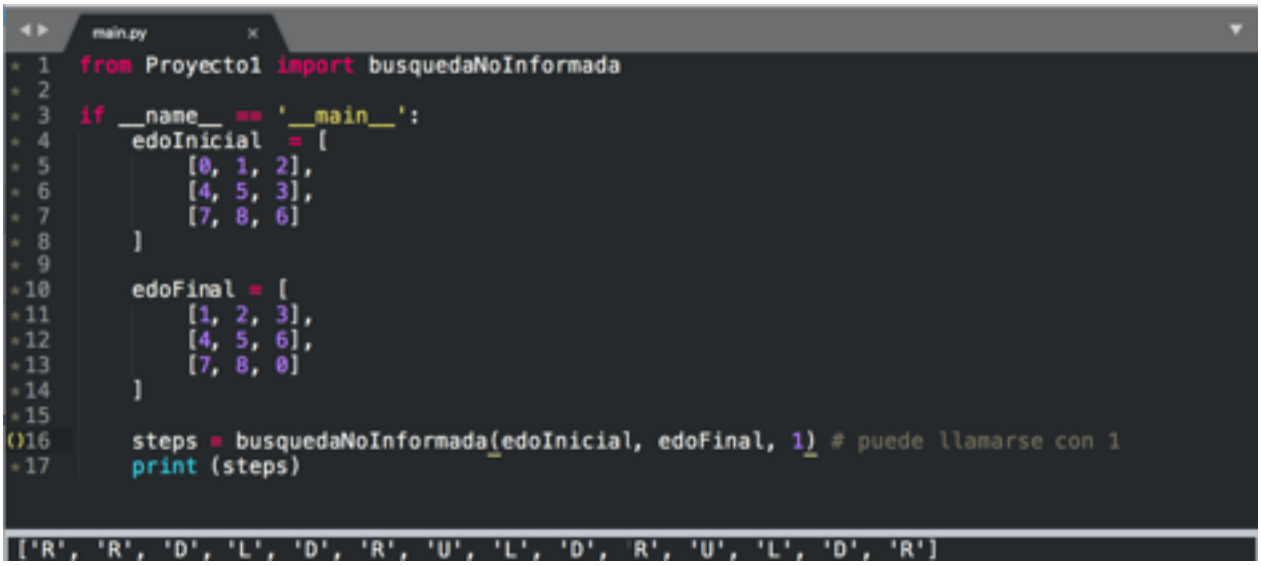
Los cuales se representan en el código de la siguiente manera:

```
main.py
1  from Proyecto1 import busquedaNoInformada
2
3  if __name__ == '__main__':
4      edoInicial  = [
5          [0, 1, 2],
6          [4, 5, 3],
7          [7, 8, 6]
8      ]
9
10     edoFinal = [
11         [1, 2, 3],
12         [4, 5, 6],
13         [7, 8, 0]
14     ]
15
16     steps = busquedaNoInformada(edoInicial, edoFinal, 0) # puede llamarse con 1
17     print (steps)

['R', 'R', 'D', 'D']
```

En el ejemplo de arriba vemos el import de la función de búsqueda, la cual recibe un estado inicial, un estado final y una bandera que nos indica que algoritmo sera usado, donde **0** es *Breadth First Search* y **1** *Depth First Search,* en esa imagen se ejecutó con el algoritmo BFS.

La función regresa un arreglo de movimientos en caso de que exista un conjunto de movimientos que nos lleva del estado inicial al final, en caso de no ser posible nos regresa un arreglo vacío. Ej. En la imagen de arriba los movimientos serian *Right, Right, Down, Down* considerando que el **numero 0** es el que se mueve



Esta segunda imagen nos muestra el mismo caso pero ahora cambiamos el algoritmo a DFS, dándonos como resultado una cadena mas larga (profunda).

Para ejecutar el código es suficiente con escribir desde la terminal

python3 main.py