

# PyNEMO 3: SE PERDIERON TODOS.

## Una aplicación con Algoritmos Búsqueda informada y no informada

Jefferson Amado Peña Torres, Cristian Alexander Valencia Torres, Domenico Di Mambro Cortes

1425590, 1329454, 1038452

(jefferson.amado.pena,cristian.a.valencia,DOMENICO)@correounivalle.edu.co

Escuela de ingeniería de sistemas y computación (EISC)

Universidad del Valle

Santiago de Cali, Colombia

**Abstract**—PyNEMO 3 es una implementación que pone a prueba varios algoritmos de búsqueda aplicadas en IA, estos algoritmos son evaluados de acuerdo a términos de tiempo, espacio y costo de la solución obtenida. Alguno de estos algoritmos de estos búsqueda no informada, búsqueda por anchura (BFS), búsqueda por profundidad (DFS), búsqueda por costo uniforme (UCS) y algunos de búsqueda informada como Avara (Heuristic) y A estrella (A star), construyen un grafo a partir de un ambiente en el que debe encontrar en un orden específico tres de los personajes de Nemo.[1]

Este informe describe como Nemo3 permitió analizar estos algoritmos, comparando el tiempo de ejecución de todos sobre una misma entrada y la forma en que procede cada uno para hallar la solución. Este informe hace parte del curso de Inteligencia Artificial(IA) de la Universidad del valle descrito por ingenieros Carlos Delgado.<sup>1</sup>

### I. INTRODUCCIÓN

Muchos de los problemas de ingeniería puede ser relacionados o modelados con una búsqueda a través de grafos, la búsqueda es uno de estos y que ha sido ampliamente estudiado generando dos grupos de acuerdo a la forma en que hallan la solución.

Estos grupos son denominados búsqueda informada y no-informada, en los que se encuentran diferentes algoritmos. En la búsqueda no-informada Amplitud, Profundidad y Costo Uniforme y en la búsqueda informada Avara o heurística y A\* (A estrella), que son discutidos y analizados en este informe.

El problema presentado en este informe es la pérdida de tres de los personajes de la película Buscando a Nemo [1], “Nemo, Marlin y Dori” en un arrecife y para ayudar a resolver este problema se han implementado diferentes estrategias de búsqueda aplicando IA, a través de un nuevo personaje Robot, este es guiado por cada uno de los algoritmos y debe hallar los tres personajes en un orden específico.

El arrecife se carga desde un archivo de texto plano y a partir de este el robot debe “construir” la ruta pasando por una serie de obstáculos y ayudas que este posee.

Por esta razón las pruebas consisten en un conjunto de archivos que moldean arrecifes de diferentes tamaños y en los que se ejecuta cada uno de las implementaciones de algoritmos y de acuerdo a las salidas se obtiene las rutas, el costo de la ruta y los tiempo de ejecución, que son comparados y finalmente permiten concluir que una “buena” heurística, en concordancia a lo visto en clase, puede conducir a un mejor resultado en cuanto a tiempo de ejecución, espacio en memoria y costo de la búsqueda.

Este informe primero describe cada uno de los algoritmos, la forma de implementación, en la segunda sección indica el tamaño algunas de las entradas de prueba, en la tercera sección el uso de la aplicación y la codificación, en la cuarta sección algunos de los resultados obtenidos y finalmente la discusión de los resultados y las conclusiones.

### II. ALGORITMOS DE BÚSQUEDA

Los algoritmos de búsqueda realizan una exploración limitada por el espacio en los sistemas de computo, por esta razón un aspecto importante es la estructura de datos que modela la estrategias de búsqueda, aunque algunos ya poseen estructuras definidas, en esta sección se hace una descripción de la implementación para Nemo3 y algunas características que permiten que la ejecución sea un poco mas rápida sin modificar el algoritmo.

#### A. BÚSQUEDA NO-INFORMADA

- 1) Búsqueda por amplitud en Nemo3:
- 2) Búsqueda por profundidad en Nemo3:
- 3) Búsqueda con costo uniforme en Nemo3:

#### B. BÚSQUEDA INFORMADA

- 1) Búsqueda heurística en Nemo3:

<sup>1</sup>carlos.andres.delgado@correounivalle.edu.co

- 2) *Búsqueda heurística 2 en Nemo3:*
- 3) *Búsqueda A estrella en Nemo3:*

### III. ENTRADAS

El arrecife es especificado desde una archivo de texto plano, donde se encuentran especificadas con números (Tiburones, Rocas, Tortugas, Hombres con arpón y Metas), según la especificación realizada por el Ing. Carlos Delgado y como se muestra en la figura 1

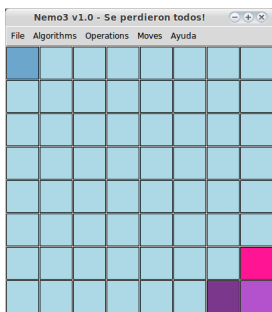
```
3
# Esta es un arrecife
# sencillo
7 6 5
4 4 2
4 0 3
```

Figure 1: Entrada simple 3x3

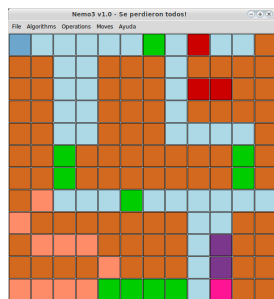
Con la finalidad de hacer rápida la ejecución de la aplicación un archivo es leído una sola vez y en esta lectura se construyen colecciones para cada uno de los elementos del arrecife, una colección de coordenadas donde se ubican los tiburones, una colección de coordenadas donde se ubican los obstáculos o rocas, etc.

El escenario que se encuentra en memoria puede ser leído por diversos algoritmos, dado que estos obtienen la posición del robot y a partir de este recorre y construye *el grafo*.

**Nemo3** Permite entradas que van desde el 2x2 hasta 20x20, aunque la ejecución de los algoritmos, tardan mucho tiempo entre mayor sea el escenario y menor cantidad de elementos (Tiburones, Rocas, Tortugas, Hombres con arpón y Metas) posea. Un ejemplo de esta situación es la que indica en las Figuras ?? una exploración en las primeras dos figuras podría costar lo mismo, mientras una exploración en la gráfica podría tardar un poco mas, claro que esto depende también de la estrategia y de condiciones de procesamiento y memoria de sistema en que se ejecuten.



(a) Entrada 8x8 con 64 espacios disponibles



(b) Entrada 12x12 con 64 espacios disponibles

Esto sera ampliado en la discusión y el análisis de los resultados. Algunas de las pruebas de entrada fueron creadas

a partir de un generador ASCII en linea, con diversas dimensiones.

### IV. PYNEMO 3: APLICACIÓN CON ALGORITMOS DE BUSQUEDA.

**PyNemo 3** es una aplicación escrita en Python [2] como se muestra en la Figura 2 que permite cargar un archivo de texto que contiene un escenario desde un archivo de texto plano con un formato como el que se especifica en la anterior sección.

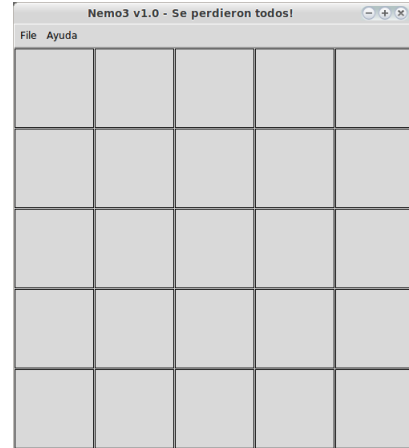


Figure 2: Interfaz grafica (GUI) de Nemo3

Una vez cargado un archivo de texto este pinta un ambiente con colores que representan cada uno de los elementos especificados en el archivo y que representan el escenario de ejecución, para la entrada especificada en la 1 la interfaz gráfica quedara como 3

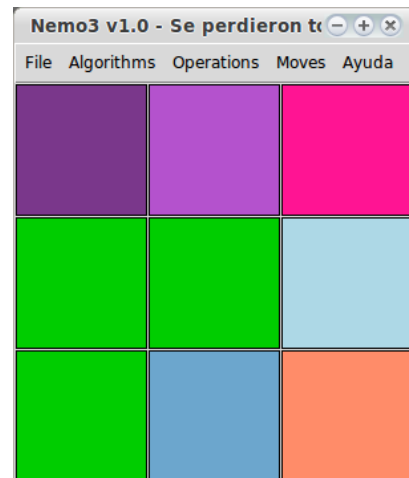


Figure 3: Interfaz grafica (GUI) de Nemo3

La representación de colores que utiliza **PyNemo 3** se encuentra en un cuadro de dialogo que puede ser abierto desde el menú ayuda 4

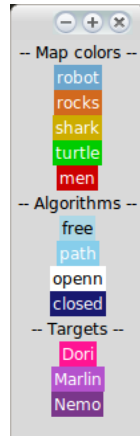


Figure 4: Ayuda de la Interfaz grafica (GUI) de Nemo3

Este menú de ayuda indica los colores que se encuentran en el mapa, los elementos que puede modificarse con el algoritmo y los objetivos. Esta aplicación es orientada a objetos, y por tal razón un algoritmo se trata como una instancia en un hilo que puede ser detenida, ralentizada o acelerada durante la ejecución, cada uno de los algoritmos implementados, puede leer ambiente, que posee varias colecciones de acuerdo a lo que se ha encontrado en el archivo de texto.

De acuerdo con lo anterior **PyNemo 3** permite selec-

cionar la estrategia que el usuario final desea ejecutar en el ambiente cargado, el orden de los movimientos que desea que el algoritmo tenga en cuenta (operadores), además un conjunto de operaciones como iniciar la ejecución, detener la ejecución, hacer la transición de estados, abiertos y cerrados lento o rápido y limpiar los cambios que ha hecho el algoritmo en la interfaz.

Una vez termine la ejecución del algoritmo se pinta la ruta que ha encontrado, además permite guardar en un archivo de texto la salida de la ejecución.

**Pasos solución:**  $(X, Y) - > (X, Y) - > \dots - > (X, Y)$

**Nodos expandidos:** X

**Nodos creados:** Y

**Costo total solución:** Z

**Factor ramificación:** W

La puesta en marcha realiza con el comando *python nemo3.py*, la interfaz esta construida con Tkinter y con python2.7 que deben estar instalados en el computador que se desea ejecutar la aplicación

#### REFERENCES

- [1] Wikipedia. Buscando a nemo — wikipedia, la enciclopedia libre, 2016.
- [2] Wikipedia. Python — wikipedia, la enciclopedia libre, 2016.