|  |
| --- |
| **ROBOT R2E2** |
| **201807328 – Rony Ormandy Ortiz Alvarez** |

**Resumen**

El ROBOT R2E2 tiene como misión de explorar nuevos terrenos, con el único problema que tiene una cantidad limitada de combustible.

Por lo cual se realizó un software que calcula rápidamente la trayectoria en la que consumirá la menos cantidad de combustible.

Se utilizo el lenguaje de programación llamado Python, el cual recibe un archivo XML, este es procesado por el programa, el cual tiene la capacidad de generar las coordenadas en el mismo tipo de archivo.

Para la optimización de la velocidad del software se utilizaron listas, listas doblemente enlazadas para las posiciones y listas simples para las trayectorias.

El algoritmo que se utilizó para el cálculo de las trayectorias es el algoritmo de Dijkstra.

**Palabras clave**

Software, Python, XML, Listas y Algoritmo de Dijkstra

***Abstract***

*The ROBOT R2E2's mission is to explore new terrain, with the only problem being that it has a limited amount of fuel.*

*Therefore, a software was developed to quickly calculate the trajectory in which it will consume the least amount of fuel.*

*We used the programming language called Python, which receives an XML file, this is processed by the program, which has the ability to generate the coordinates in the same type of file.*

*To optimize the speed of the software, lists, double linked lists for the positions and simple lists for the trajectories were used. The algorithm used for the calculation of the trajectories is Dijkstra's algorithm.*

***Keywords***

Software, Python, XML, Lists y and Dijkstra´s algorithm.

**Introducción**

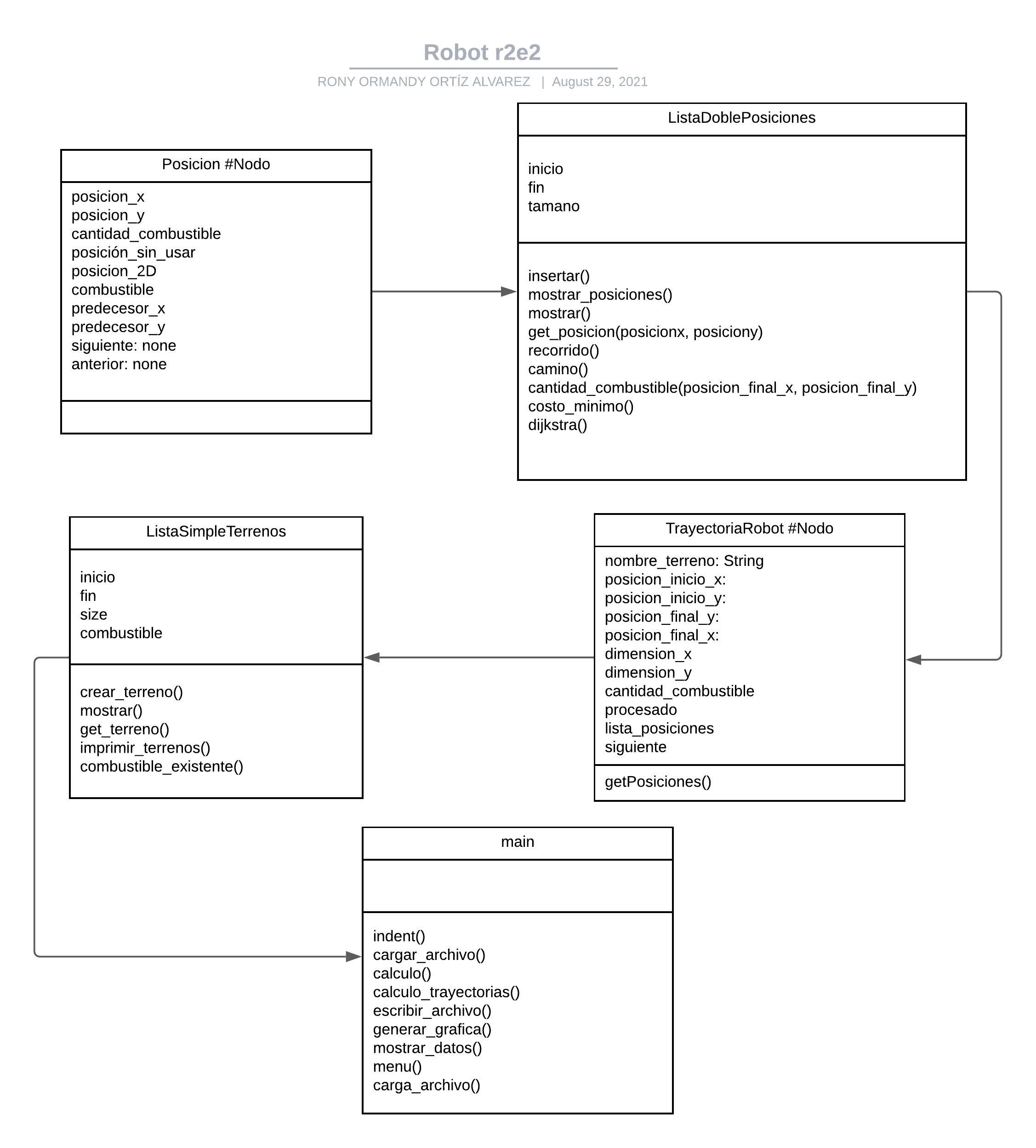
El ROBOT R2E2 tiene como misión explorar nuevos terrenos, teniendo los movimientos adelante, atrás, izquierda y derecha, nunca en diagonal el problema que afronta es la cantidad de combustible para moverse es limitada además que el terreno en el que se mueve no es homogéneo y consumirá más o menos combustible.

Por lo cual se necesita una solución de software que calcule rápidamente la trayectoria más eficiente para que se consuma la menor cantidad de combustible posible.

Se debe de construir una clase de algoritmo que calcule la trayectoria más adecuada además de ser rápido y eficiente.

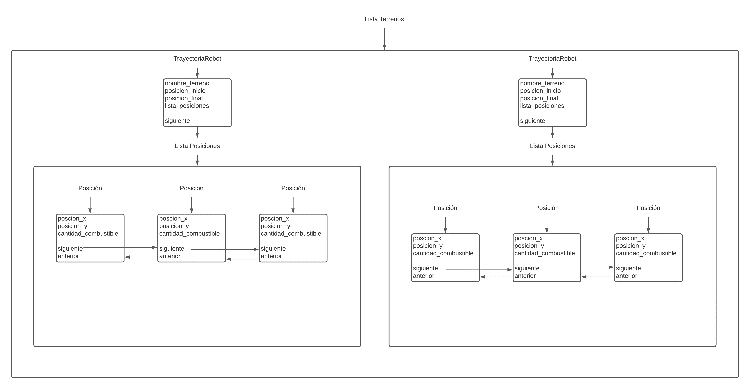
**Desarrollo del tema**

Diagrama de clases:



*Figura 1.* Diagrama de clases.

Fuente: elaboración propia.



*Figura 1.* Diagrama de estados

Fuente: elaboración propia.

Para solucionar el problema del costo de combustible se uso el algoritmo de Dijkstra:

Método que sigue el algortimo de Dijkstra

1. Asignar a cada nodo una distancia tentativa: 0 para el nodo inicial e infinito para todos los nodos restantes. Predecesor nulo para todos.

2. Establecer al nodo inicial como nodo actual y crear un conjunto de nodos no visitados.

3. Para el nodo actual, considerar a todos sus vecinos no visitados con peso w.

a) Si la distancia del nodo actual sumada al peso w es menor que la distancia tentativa actual de ese vecino, sobrescribir la distancia con la suma obtenida y guardar al nodo actual como predecesor del vecino

4. Cuando se termina de revisar a todos los vecinos del nodo actual, se marca como visitado y se elimina del conjunto no visitado

5. Continúa la ejecución hasta vaciar al conjunto no visitado

6. Seleccionar el nodo no visitado con menor distancia tentativa y marcarlo como el nuevo nodo actual. Regresar al punto 3

**Conclusiones**

**Referencias bibliográficas**

Allen. Downey, (2002). *Como Pensar como un Científico de la Computacion con Python.*

Graphviz, (2021, 29, 08). *Graphivz. https://graphviz.org/*