

Algoritmos de caminos para peatones que reducen tanto el acoso callejero como la distancia

Samuel Aguilar Villada

saguiarv@eafit.edu.co

Angy Sánchez Moreno

ajsanchezm@eafit.edu.co

Resumen

El presente informe tiene por objetivo analizar y solucionar la búsqueda eficiente de rutas en una ciudad tan concurrida como lo es Medellín. Lo que se pretende es encontrar una estructura de datos que permita brindar varias opciones de rutas en las cuales se evaluará: la reducción en la distancia a recorrer y el acoso callejero que se pueda vivenciar.

El principal enfoque del proyecto es permitirle al usuario la mejor elección de ruta para llegar a su destino evaluando como se mencionó anteriormente dos características principales, con esto se quiere brindar un poco de confianza y tranquilidad a la hora de movilizarse por la ciudad debido a que se espera solucionar o menguar un problema que ha estado en auge en los últimos años.

Palabras clave

Ruta; mapeo; dirección; estructura de datos; movilidad; listas enlazadas; lambda, librerías.

Palabras clave de la clasificación de la ACM

Data structures design and analysis, sorting and searching, design and analysis of algorithms, Operating systems.

1. Introducción

La evolución tecnológica a la que la sociedad ha estado expuesta en las últimas décadas ha brindado alternativas para hacer la movilidad en espacios sumamente concurridos un poco más práctica, este avance se ha logrado empleando diversos modelos de geolocalización que permiten generar rutas alternas evitando así trancones, accidentes automovilísticos, vías cerradas, entre otros problemas a los que se está expuesto en la vía, además de calcular el tiempo promedio en el desplazamiento. Sin embargo, un problema que aún no se ha abordado desde esta perspectiva – o era muy poco su grado de interés – es el acoso que se vive en las calles por las que los individuos se movilizan; así pues, esta deficiencia en la calidad de movilidad por la ciudad fue lo que dio el inicio a este proyecto.

2. Problema

El problema al cual nos enfrentamos es el aumento en el acoso callejero lo que

restringe la tranquilidad en el libre desplazamiento por ciertas áreas.

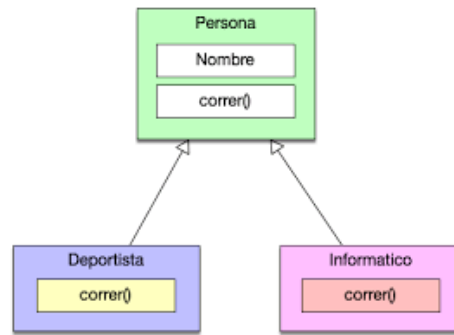
Para ello se hará uso de listas y archivos en los cuales se almacenarán los datos que brindarán información acerca de la mejor ruta, para la elección de la ruta más factible se tendrá en cuenta lo eficaz que sea el desplazamiento – menor tiempo posible – y la seguridad con la que se pueda transitar.

3. Trabajos relacionados

3.1 Herencia y polimorfismo

La herencia es la capacidad que tiene una clase de heredar los atributos y métodos de otra, algo que nos permite reutilizar código y hacer programar mucho más óptimos. Así pues la idea de la herencia es identificar una clase base (la superclase) con los atributos comunes y luego crear las demás clases heredando de ella (las subclases) extendiendo sus campos específicos (es posible utilizar el comportamiento de una superclase sin definir nada en la subclase).

El polimorfismo es una propiedad de la herencia por la que objetos de distintas subclases pueden responder a una misma acción. La polimorfía es implícita en Python, ya que todas las clases son subclases de una superclase común llamada Object. Ya que todas las clases son subclases de una superclase común llamada Object.



3.2 Librerías

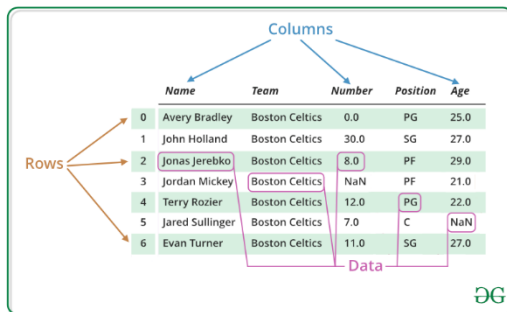
A grandes rasgos, cabe destacar que en programación una librería responde al conjunto de funcionalidades que permiten al usuario llevar a cabo nuevas tareas que antes no se podían realizar. Es decir, las librerías de Python responden al conjunto de implementaciones que permiten codificar este lenguaje, con el objeto de crear una interfaz independiente.

Librerías usadas en el proyecto:

- **Pandas:** Pandas es muy usada en Data Science. Los datos son fáciles de trabajar, ya que sus estructuras están conformadas por una serie de datos, así como el Data Frame para dos dimensiones. Sus características principales: Define nuevas estructuras de datos basadas en los arrays, permite leer y escribir fácilmente ficheros en formato CSV, Excel y bases de datos SQL, permite acceder a los datos mediante índices o nombres para filas y columnas, ofrece métodos para reordenar, dividir y combinar conjuntos de datos.

Pandas cuenta con tipos de datos, en este

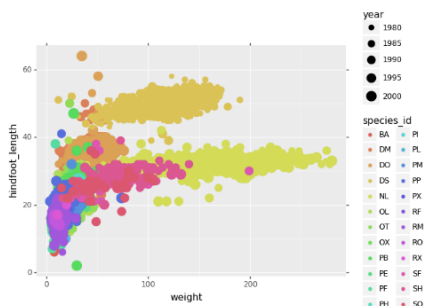
proyecto como es evidente trabajamos con DataFrame.



The diagram illustrates a DataFrame structure. It shows a table with columns labeled 'Name', 'Team', 'Number', 'Position', and 'Age'. The rows are indexed from 0 to 6. A pink box highlights the entire table, labeled 'Data'. A blue box highlights the columns, labeled 'Columns'. An orange box highlights the rows, labeled 'Rows'.

	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

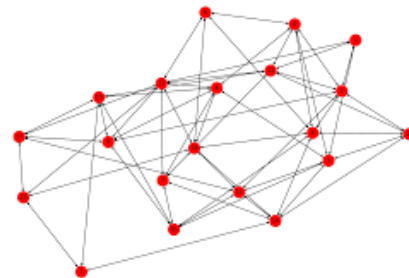
- **gmaplot:** Nos permite trazar datos en Google Maps. gmaplot tiene una interfaz similar a matplotlib para generar HTML.



- **Web browser:** En Python, el módulo de navegador web es un controlador de navegador web conveniente. Proporciona una interfaz de alto nivel que permite mostrar documentos basados en la Web a los usuarios. Web browser también se puede utilizar como una herramienta CLI. Acepta una URL como argumento con los siguientes parámetros opcionales: -n abre la URL en una nueva ventana del navegador, si es posible, y -t abre la URL en una nueva pestaña del navegador.

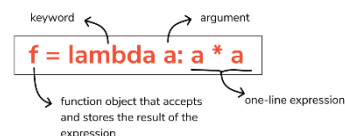
3.3 Nodos

Al igual que los bloques de código, los nodos de Python son una interfaz de secuencias de comandos dentro de un entorno de programación visual.



3.4 Lambda

Las funciones lambda son también conocidas como funciones anónimas porque se definen sin un nombre, características: Son funciones que pueden definir cualquier número de parámetros pero una única expresión, esta expresión es evaluada y devuelta, se pueden usar en cualquier lugar en el que una función sea requerida, estas funciones están restringidas al uso de una sola expresión, se suelen usar en combinación con otras funciones, generalmente como argumentos de otra función. Unas de las aplicaciones más comunes de esta función son map(), filter() y reduce().



4. Estructura de datos

4.1 Diseño de la estructura de datos

Dado que una estructura de datos es una forma particular de organizar datos en una computadora para que puedan ser utilizados de manera eficiente, se

convirtió en el modelo perfecto que se adapta a las necesidades de este proyecto, es así como al hacer uso de esta estructura de datos se logra satisfacer todos los objetivos que tiene el proyecto de manera correcta; permitiéndonos:

- Analizar el archivo de direcciones registradas.
- Dividir por secciones dichas direcciones en 7 categorías.
- Obtener individualmente cada información de las categorías para los subprocesos.
- Comparar dos direcciones tomando el atributo de coordenadas.
- Imprimir por pantalla direcciones guardadas.
- Situarse en el archivo de direcciones para verificar si posee la dirección en el mapa guardado.

4.2 Operaciones de la estructura de datos

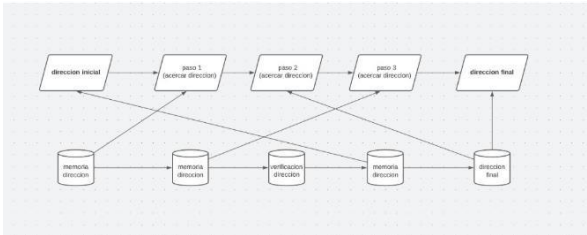


Tabla 1

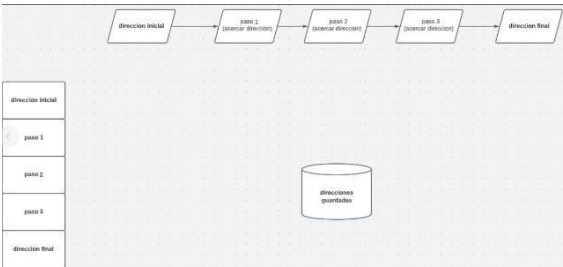


Tabla 2



Tabla 3

Tabla 1, 2 y 3: A partir del mapa la estructura de datos recorre el archivo de coordenadas agregando direcciones según la información suministrada

4.3 Análisis de complejidad

metodo	complejidad
vaciar	O(1)
AgregarDireccion	O(1)
AgregarPosicion	O(n*m)
BuscarDireccion	O(n*2)
NuevaPosicion	O(n*m)

Tabla 4: Tabla para reportar la complejidad del algoritmo

4.4 Memoria

conjunto de datos	1	2	3	4
conjunto de memoria	4823400B	6432412B	7954371B	9674321B

Tabla 5: Consumo de memoria en la estructura de datos

4.5 Tiempo

```
-----
Time: 4.8393895626068115
```

```
-----
Time: 309.1835572719574
```

```
-----
Time: 285.02525997161865
```

Tablas 6: Tiempo empleado para realizar la acción del programa desde varias direcciones.

Referencias

1. Los conceptos de encapsulación, herencia, polimorfismo y composición de la programación orientada a objetos. (2021, March 31). *Blog Bitix*. <https://picodotdev.github.io/blog-bitix/2021/03/los-conceptos-de-encapsulacion-herencia-polimorfismo-y-composicion-de-la-programacion-orientada-a-objetos>
2. Immune, M. (2022, March 17). Librerías de Python, ¿qué son y cuáles son las mejores? *Immune Technology Institute*. <https://immune.institute/blog/librerias-python-que-son/>
3. *Gmplot*. (n.d.). PyPI. Retrieved November 9, 2022, from <https://pypi.org/project/gmplot/>
4. *webbrowser* — Controlador de navegador web conveniente — documentación de Python - 3.11.0. (n.d.). Python.org. Retrieved November 9, 2022, from

<https://docs.python.org/es/3/library/webbrowser.html>

5. *Funciones lambda en Python: map(), filter() y reduce()*. (2020, April 15). J2LOGO. <https://j2logo.com/python/funciones-lambda-en-python/>