



# Proyecto Semestral, Algoritmos II

## Título: Uber.

(equipos de 2 estudiantes solamente)

### Objetivos Generales

- Desarrollar una aplicación para implementar la movilidad de personas como un servicio.
- Desarrollar mecanismos eficientes para encontrar las mejores rutas dentro de un mapa.

### Introducción

Uber es una empresa que brinda la movilidad como un servicio. A partir de una aplicación, el mapa de una ciudad y una dirección dentro del mapa es posible llevar de un punto a otro a una persona.

Por ello, la aplicación consiste de 4 componentes fundamentales:

<mapa>

<direcciones>

<ubicaciones fijas>

<ubicaciones móviles>

A continuación se describen cada uno de los componentes.

**Mapa:** Consiste de dos conjuntos **<E,C>**. **E** es un conjunto de esquinas  $\{e_1, e_2, e_3, \dots\}$  y **C** las calles que conectan dichas esquinas. **C** es un conjunto de ternas ordenadas  $\{<e_1, e_2, c>, <e_3, e_4, c>, <e_2, e_1, c>\}$  que expresa la dirección y el largo de las calles ( $c$  = distancia entre  $e_1$  y  $e_2$ ).

**Dirección:** Consiste de un par de tuplas  $\{<e_x, d>, <e_y, d>\}$  donde **e** es una esquina dentro del mapa y **d** es un entero que representa la distancia de esa dirección con la esquina **e**.

Ej: la dirección  $d_1 = \{<e_x, 10>, <e_y, 5>\}$  indica que  $d_1$  esta en la siguiente posición dentro del mapa  $e_1$  10  $d_1$  5  $e_2$

**Ubicación fija:** Consiste de una tupla **<Nombre, Dirección>**. Donde **Dirección** representa una dirección posible en el mapa y **Nombre** puede ser cualquiera de estos caracteres.

- **H: hospital**
- **A: almacén**
- **T: tienda**
- **S: supermercado**
- **E: escuela**
- **K: kiosco**
- **I: iglesia**

**Ubicación móvil:** Consiste de una 3-tupla <Nombre, Dirección, Monto>. Donde **Dirección** representa una dirección posible en el mapa, **Monto** un valor numérico para indicar saldo/tarifa y **Nombre** puede ser cualquiera de estos caracteres.

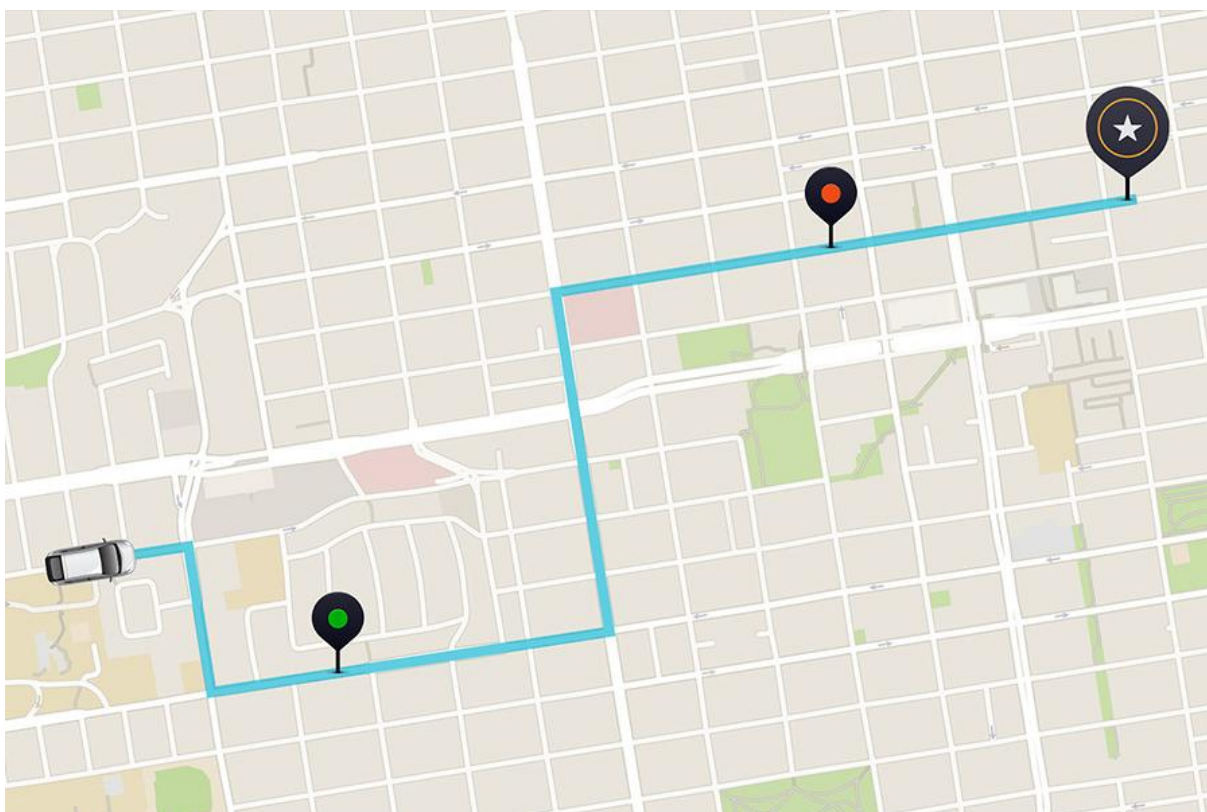
- **P: persona**
- **C: auto**

## Indicaciones

Si dada la configuración inicial de un mapa y la ubicación en el mismo de lugares fijos y componentes móviles (autos y personas). Implementar las siguientes funcionalidades:

1. Cargar nuevos lugares, personas y autos dentro del mapa
2. Conocer, dado un lugar, persona o auto la dirección del mismo.
3. Conocer, dado una persona que se encuentra en el mapa cuáles son los 10 autos más cercanos que esa persona puede pagar.
4. Conocer, dado dos direcciones en el mapa cual es el camino más cercano para llegar de uno a otro.

**Figura 1: Ejemplo ilustrativo del recorrido más corto para llegar al destino.**



## Requerimientos

- Lograr el cumplimiento de los objetivos a través de una aplicación (script) utilizando el lenguaje de programación python3.
- A través de la aplicación desarrollada, permitir la movilidad de las personas cargadas dentro del mapa.
- Para lograr la movilidad de las personas es necesario la creación de un mapa y para

ello se utilizará el siguiente comando: `python uber.py -create_map <local_path>`. Donde `<local_path>` es un fichero que contiene la información inicial del mapa.

- Una vez creado el mapa los elementos dentro del mapa deberán ser cargados utilizando los siguiente comando:
  - `python uber.py -load_fix_element <nombre,dirección>`
  - `python uber.py -load_movil_element <nombre,dirección, monto>`
- Los nombres de los elementos dentro del mapa deben ser únicos por lo que si se intenta agregar un elemento con nombre repetido la aplicación debe notificarlo. Para los nombres solo se permiten combinaciones de caracteres alfanuméricos. Ej: H1 para indicar el hospital 1, C10 para indicar el auto 10 y P2 para indicar la persona 2.
- Una vez cargados los elementos dentro del mapa crear la siguiente funcionalidad a través del comando:

1. `python uber.py -create_trip <persona>  
<direccion>/<elemento>`

- a. Devuelve ranking de los 3 autos que puede pagar más cercanos a la dirección actual de la persona. El precio del viaje estará definido por: *(costo del camino más corto (auto <-> persona) + tarifa del auto)/4*. Una persona puede pagar un viaje si su monto supera el costo del viaje.
  - b. Devuelve el camino más corto para llegar a destino
  - c. Panel interactivo para indicar si se acepta o no el viaje. En caso de aceptar el viaje indicar el auto que realizará el recorrido. Una vez que se acepta el viaje los elementos involucrados (auto, persona) quedan dentro del mapa en la posición de destino del viaje y el monto de la persona se le resta el costo del viaje.
- Para el desarrollo de la aplicación se puede utilizar cualquiera de las estructuras y funciones nativas de python u otra estructura de datos o funciones definidas por ustedes.
  - Garantizar la persistencia de los datos. Esto significa que la estructura que compone el índice de los elementos dentro del mapa tiene que ser recuperable a través de consultas en todo momento.
  - Los equipos de trabajo deben estar compuestos por 2 estudiantes. No se permiten trabajos individuales y en caso de que el número total de estudiantes sea impar se conformará solamente un equipo de 3 estudiantes.

## Evaluación del proyecto

- Para la evaluación del proyecto entra en consideración los siguientes factores:
  1. Perfecto entendimiento de cada integrante del equipo de todo el código del proyecto.
  2. Perfecto entendimiento de cada integrante del equipo de los problemas surgidos y soluciones generadas durante toda la fase de desarrollo de la aplicación.
  3. Correcto funcionamiento de la aplicación acorde a los objetivos planteados.
  4. Claridad y documentación del código.
  5. Correcta elección de las estructuras de datos y algoritmos utilizados.
  6. Eficiencia de la aplicación relacionada al costo temporal y espacial.



## Creación de un Mapa

- Para la creación del índice de la biblioteca se utilizará el siguiente comando: **python uber.py -create\_map <local\_path>**
- **<local\_path>** representa la dirección local de la carpeta que contiene el documento con la información de las esquinas y sus calles. El documento contiene dos líneas, la primera son las esquinas y la segunda las calles.
  - $E = \{e1, e2, \dots\}$
  - $A = \{<e1, e2, c>, <e3, e4, c>, <e2, e1, c>, \dots\}$
- Una vez finalizado el proceso de creación de mapa, la aplicación devolverá el texto ***"map created successfully"***. A partir de este momento se puede iniciar a interactuar con el mapa.
- La información deberá persistir de manera que se pueda acceder a la información del mapa y crear viajes en todo momento. Esto significa que no se deberá volver a crear tal índice de elementos en cada búsqueda, sino que se realizará sobre una estructura persistente en disco, que se levantará a memoria cada vez que se requiera hacer una consulta.

## Estructura de la Aplicación a realizar

- Se implementará un script en python utilizando la versión 3. El script tendrá el nombre **uber.py**. Sobre ese script se realizarán las operaciones de creación y búsqueda de información. El manejo de errores, excepciones y posibles valores de entrada corren a cargo de los desarrolladores de la aplicación. Dicho script será utilizado para realizar las pruebas para evaluar el desempeño de la aplicación.