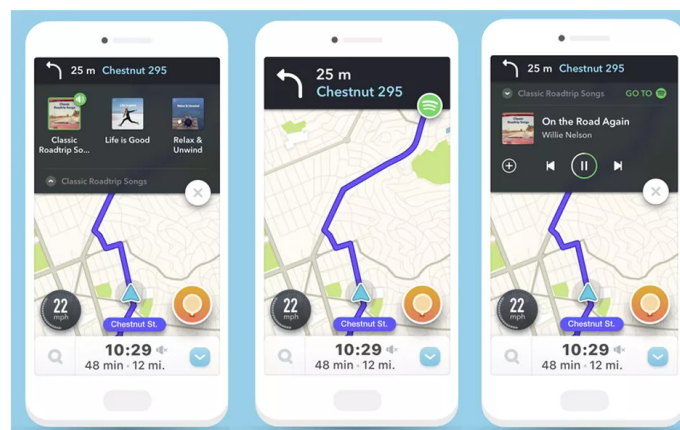


## **Proyecto Programado #2 - MiniWaze**

### ***1. Introducción***

Actualmente existen varias aplicaciones que asisten a las personas en cómo llegar a un destino haciendo uso de servicios de localización de nuestros dispositivos móviles visualizando esto con mapas y el sistema vial (ver Figura 1), estas aplicaciones pueden ayudar al conductor en orientarlo sin infringir las leyes de tránsito o incurrir en un accidente.

Para efectos de este proyecto, nuestro mapa será una matriz que deberá ser leído desde un archivo externo con el formato csv, y con ello llevarlo a una representación gráfica de cuadras, calles, avenidas, bloqueos, accidentes entre otros.



**Figura 1** Pantallas de la aplicación Waze

### ***2. ¿Qué se busca con este proyecto?***

El objetivo general de este proyecto es facilitar un acercamiento con aplicaciones de posicionamiento y mapas desde la perspectiva de la programación, de manera que las reglas que lo enmarcan sean analizadas minuciosamente para desarrollar un diseño que sea programado mediante el lenguaje Python. Con esto se busca:

1. Practicar las habilidades de aplicaciones de software.
2. Ejercitar la toma de decisiones sobre el dominio del problema y de la solución.
3. Aplicar los conceptos de programación, manejo de archivos y de matrices.
4. Manejo de componentes para el uso de interfaz gráfica.

### 3. Proyecto por desarrollar

Se requiere que el estudiante desarrolle un pensamiento crítico y fortalezca su perfil en la investigación acerca de las funcionalidades detalladas sobre aplicaciones de localización y orientación por ejemplo Waze, GoogleMaps, Maps, Bing, entre otros, donde un usuario en una posición determinada necesita llegar a un destino X, pero para ello se analiza diferentes rutas y como resultado muestra la trayectoria más efectiva.

Asegúrese de investigar:

1. Visualización del mapa
2. Interfaz gráfica con TK Inter
3. Análisis de rutas

La aplicación debe disponer de las funcionalidades que a continuación de describen.

#### a. Ventana de inicio

Será necesario una autenticación (usuario / contraseña) cada vez que ingrese a la aplicación, y todo lo que se cree durante su instancia será vinculado a ese usuario. Los usuarios deben de estar guardados en un archivo llamado Usuario.txt bajo el siguiente formato:

ccampos;123

hperez;456

Se debe desarrollar una ventana en la cual el usuario tiene las opciones:

- a) Cargar mapa
- b) Seleccionar destino
- c) Planificar destino
- d) Guardar destino
- e) Borrar destino
- f) Modificar mapa
- g) Salir

#### b. Crear y Cargar mapa

Este deberá ser cargado desde un archivo de texto en formato csv, en donde por medio de una representación de una matriz por defecto y mínimo (10x10), pero podrá ser personalizada, una vez definido este podrá crear las avenidas, cruces, calles, cuadras. La calles y avenidas tendrán una dirección definida, en la tabla siguiente se muestra los símbolos a utilizar para definir el mapa base

Tabla 1 Símbolos para definición del mapa

Valor	Descripción
0	Cuadras

Valor	Descripción
N	Son avenidas donde su dirección de navegación será de sur a norte (North)
S	Son avenidas donde su dirección de navegación será de norte a sur (South)
L	Son calles donde su dirección de navegación será de derecha a izquierda (Left)
R	Son calles donde su dirección de navegación será izquierda a derecha (Right)
C	Son las intersecciones entre las calles y avenidas
ND	Son aquellas calles donde no tiene definido su dirección, en otras palabras, se puede manejar en doble sentido

A continuación, en las siguientes imágenes se ilustran el cómo sería el contenido del archivo (csv, separado por ;) y lo que sería su representación gráfica.

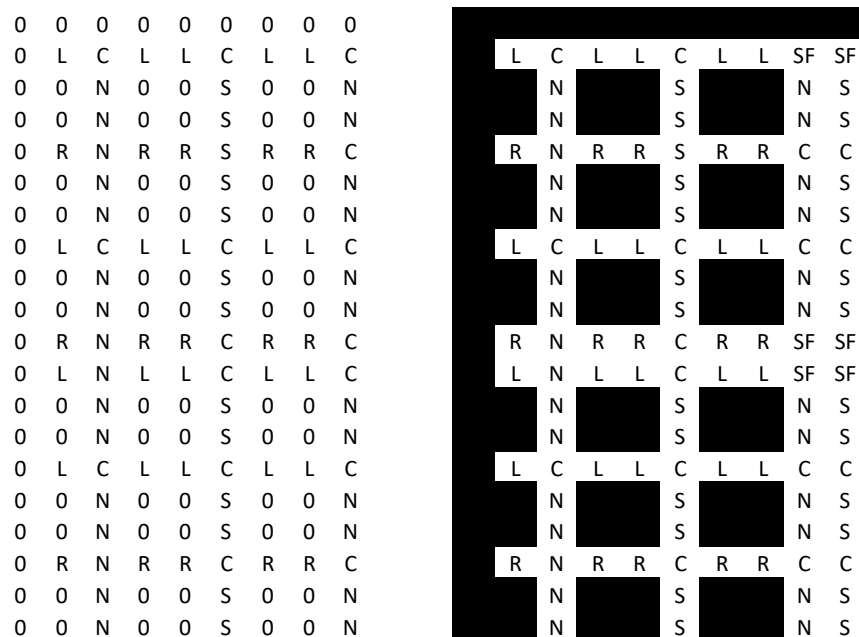


Figura 2 Representación en texto del mapa y su representación gráfica

```

0;0;0;0;0;0;0;0;0
0;L;C;L;L;C;L;L;C
0;0;N;0;0;S;0;0;N
0;0;N;0;0;S;0;0;N
0;R;N;R;R;S;R;R;C
0;0;N;0;0;S;0;0;N
0;0;N;0;0;S;0;0;N
0;L;C;L;L;C;L;L;C
0;0;N;0;0;S;0;0;N
0;0;N;0;0;S;0;0;N
0;R;N;R;R;C;R;R;C
0;L;N;L;L;C;L;L;C
0;0;N;0;0;S;0;0;N
0;0;N;0;0;S;0;0;N
0;L;C;L;L;C;L;L;C
0;0;N;0;0;S;0;0;N
0;0;N;0;0;S;0;0;N
0;R;N;R;R;C;R;R;C
0;0;N;0;0;S;0;0;N
0;0;N;0;0;S;0;0;N

```

Figura 3 Representación del archivo csv

### c. *Seleccionar destino*

Una vez cargado el mapa, debe permitir al usuario ubicar su punto inicial y su destino en el mapa, para ello depende de la creatividad del programador el cómo representarlo, se le recomienda el uso del **mouse**.

La aplicación debe permitir al usuario insertar una hora, esto es para poder calcular la trayectoria ya sean en “horas pico” o fuera de ella

Con los puntos de inicio y fin definido, el usuario podrá indicar al sistema que calcule la ruta o rutas, una vez realizado esto la aplicación le muestra la ruta de menor costo como primera opción, de lo contrario, el usuario puede visualizar cada una de las rutas y seleccionar el que más le convenga.

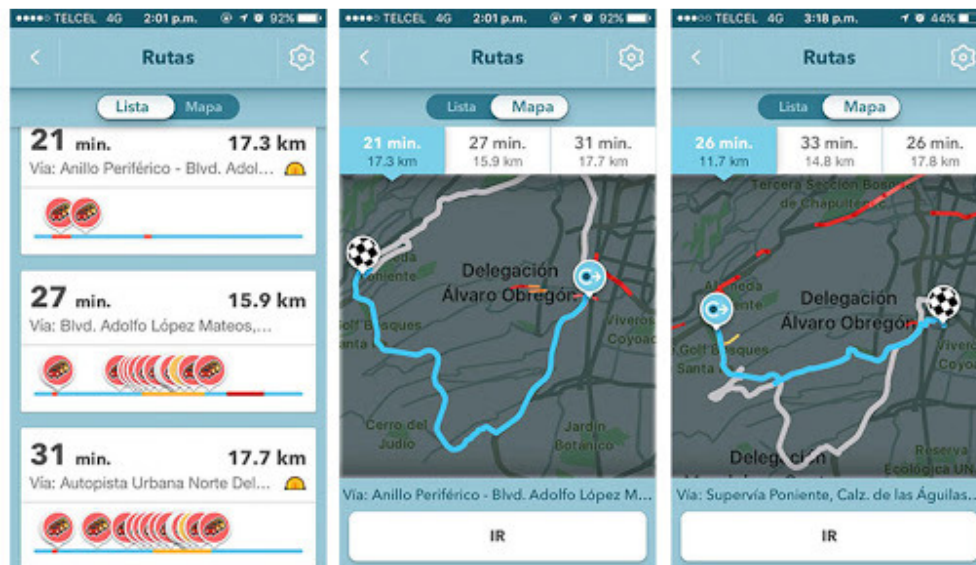


Figura 4 Ejemplo de cómo Waze muestra las rutas calculadas

#### ***d. Guardar destino***

- El usuario puede guardar el destino que seleccionó, se debe de tomar en cuenta que en un mismo mapa el usuario podría guardar más de un destino por vez.
- El programador debe crear una forma de guardar el o los destinos haciendo referencia un mapa.
- El usuario posteriormente puede volver a cargar el mapa y mostrar los destinos de este por si en una nueva ubicación actual, él vuelva hacer uso de un destino.

#### ***e. Planificar destino***

- El usuario puede programar su hora de salida con una ruta específica, este deberá calcular su duración dado la hora de salida de este.

#### ***f. Borrar destino***

- Dado que un mapa puede tener más de un destino, el usuario puede prescindir de ellos con el uso de esta funcionalidad, solo tendría que seleccionarlos y posteriormente borrarlos. La aplicación debe mostrar un mensaje de confirmación de esta acción.

#### ***g. Modificar mapa (Extra)***

Una vez con el mapa cargado por primera vez, el usuario puede personalizarlo agregando nuevos elementos a incorporar a las cuadras, a continuación, se detalla las simbologías para estos nuevos edificios. Estos nuevos elementos deben ser apreciables a la vista del usuario.

Tabla 2 Simbología de edificios o ubicaciones

Valor	Símbolo
P	Parque
E	Escuelas / Colegios
H	Hospital
Z	Zonas deportivas
B	Calle / Avenida bloqueada (no se puede transitar)
T	Oficentros / Zona francas

### ***h. Cálculo de rutas***

Cuando el usuario calcule la duración de su trayecto debe tomar en cuenta que su valor puede variar según la hora del día en que realiza esta operación. A continuación, se muestra los valores para cada uno de los elementos del mapa.

Tabla 3 Pesos de elementos del mapa según hora pico o no del día

Símbolo	Descripción	Hora pico <sup>1</sup> , valor	Hora normal
L / R	Calles	2	2
N / S	Avenidas	4	1
C	Cruces	3	2

## ***4. Puntos Extra***

Se otorgarán puntos extra al permitir agregar las funcionalidades adicionales:

- **(5 puntos)** Animación: Realizar el recorrido del vehículo desde la posición inicial a su destino
- **(5 puntos)** Modificar Mapa

## ***5. Aspectos técnicos***

El proyecto deberá estar escrito en el lenguaje de programación Python y se deben desarrollar las funcionalidades de usuario por medio de interfaz gráfica (**Tkinter**). Además, considerar lo siguiente:

- Se deben manejar mensajes claros al usuario.
- Realizar validaciones de captura de campos y movimientos.

---

<sup>1</sup> Las horas serán de 6 a 9am, 12 a 1pm y de 5 a 8pm. La aplicación de permitir establecer la hora.

- **Restricción de las usuales** funciones bult-in de Python que deseen utilizar debe ser validada con el profesor (no incluye las relacionadas a interfaz).
- En el desarrollo del programa se deberá utilizar **iteración**.
- Deben utilizar nombres de variables, argumentos y funciones significativas.

## 6. Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de estos.

Para la **documentación interna**, deberán incluir comentarios descriptivos para cada función, con sus **entradas, salidas, restricciones**. La **documentación externa** deber ser en **PDF** (si lo suben en otro formato, **no será leído**) y deberá incluir:

- Portada.
- **Instrucciones de ejecución**.
- Un enlace a un video en **youtube o te MStTeams** donde muestre la funcionalidad de su aplicación. Su duración no de ser mayor a los **15 minutos**.
- Descripción del problema.
- Diseño del programa: decisiones de desarrollo, algoritmos usados.
- Librerías usadas: creación de archivos, etc.
- Análisis de resultados: lista detallada de objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).
- **Bitácora** en GitHub con los **commit** por usuario incluyendo su descripción al momento de hacerlo.
- Conclusión (es), su punto de vista sobre el programa desarrollado

## 7. Evaluación

La evaluación se va a centrar en dos elementos: programación y documentación. El proyecto programado **tiene un valor de 20% de la nota final**, en el rubro de Proyectos.

Desglose de la evaluación del proyecto programado:

- Documentación interna 10 puntos.
- Documentación externa 5 puntos.
- Funcionalidad 80 puntos (ver detalle en Software a Desarrollar)
- Revisión del proyecto 5 puntos. (No solo consiste en estar presente, sino saber responder las preguntas dadas por el profesor)

## 1. Rúbrica de evaluación

FUNCIONALIDAD		80%	
Rubro	Valor	Obtenido	Observaciones
Interfaz gráfica	10		
<b>Opciones Administrativas</b>			
Control de acceso	10		
Crear y Cargar Mapa	20		
Seleccionar Destino	10		
Guardar Destino	10		
Planificar Destino	10		
Borrar Destino	10		
Calcular rutas	20		
<b>Extras</b>			
Modificar mapa	5		
Animación del recorrido	5		
<b>TOTAL</b>	<b>110</b>	<b>0</b>	

### 8. Forma de trabajo

El trabajo se debe realizar de forma **parejas o individual**.

### 9. Aspectos administrativos

Crear 2 carpetas llamadas **documentación** y **programa**, en la primera deberá incluir el documento **PDF** solicitado y en la segunda los archivos y/o carpetas necesarias para la implementación de este proyecto programado.

#### **NO SUBIR ARCHIVOS EN FORMATO ZIP o SIMILARES**

Deben modificar el archivo llamado **README.md**, este archivo debe contener la siguiente información:

- Nombre del Estudiante y Número de carné del estudiante
- Estatus de la entrega (debe ser **CONGRUENTE** con la solución entregada):  
[Deplorable | Regular | Buena | MuyBuena | Excelente | Superior]

Con este enlace <https://docs.github.com/es/github/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax> le ayudará a dar formato al archivo README.md

### 10. Entrega

Será el **22 de mayo de 2023**. Después de esa fecha con un **máximo de 2 días** de retraso se le **restará 10 puntos**, después de eso no es aceptado.

Los archivos fuentes pueden ser revisados en el sistema de Control de Plagio del TEC Digital. **Todo el código de cada proyecto debe ser 100% original y en caso de plagio se asignará nota cero.**