

[Asunto]

CÁLCULO DE RUTAS



-Mediante OSM y Google Maps-



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA SUPERIOR
DE INGENIERÍA GEODÉSICA,
CARTOGRÁFICA Y TOPOGRÁFICA

Cartografía Urbana Informatizada/
Técnicas Gráficas

2013/2014

Marqués Mateu, Ángel

Autor:

Gimeno Vilanova, Enric

THIS PAGE INTENTIONALLY LEFT BLANK



TABLA DE CONTENIDOS

1.	INTRODUCCIÓN Y OBJETIVOS.....	5
2.	LISTADO LIBRERIAS NECESARIAS	5
3.	ESQUEMA FLUJO DE TRABAJO	6
4.	DESARROLLO	9
4.1.	CREACIÓN CANVAS - INTERFAZ GRAFICA.....	9
4.2.	GOOGLE MAPS.....	10
4.2.1.	OBJETO GOOGLE MAPS.....	10
4.2.2.	LISTBOX INDICACIONES	11
4.2.3.	BOUNDING BOX DE LA RUTA	12
4.2.3.1.	CENTRO DEL BOUNDING BOX.....	12
4.2.4.	CREACIÓN DEL MAPA	13
4.2.4.1.	CALCULO URL STATIC MAP	14
4.2.4.2.	GUARDADO Y CARGA DE LA IMAGEN	16
4.2.5.	DESPLAZAMIENTO DEL MAPA	17
4.2.6.	ZOOM DEL MAPA.....	17
4.3.	OPEN STREET MAP	17
4.3.1.	CALCULO DEL TILE CENTRAL.....	17
4.3.2.	CREACIÓN DE LA CACHE	18
4.3.3.	GUARDADO Y CARGA DEL MAPA	19
4.3.4.	DESPLAZAMIENTO DEL MAPA	19
5.	COMPARATIVA	19
6.	RESULTADOS.....	20
7.	CONCLUSIONES.....	22
8.	BIBLIOGRAFÍA	22
9.	ANEXOS	23
9.1.	CODIGO.....	23

LISTA DE FIGURAS

<i>Ilustración 1. Interfaz gráfica de la aplicación.</i>	<i>10</i>
<i>Ilustración 2. Extracto resultado de la función “directions”</i>	<i>11</i>
<i>Ilustración 3. Listbox indicaciones.</i>	<i>11</i>
<i>Ilustración 4. Bounding box de la ruta.</i>	<i>12</i>



<i>Ilustración 5. Ecuación del punto medio.....</i>	<i>12</i>
<i>Ilustración 6. Ejemplo gráfico de la URL.....</i>	<i>16</i>
<i>Ilustración 7. Formulación Tiles.</i>	<i>18</i>
<i>Ilustración 8. Ejemplo resultado Google Maps - Roadmap.....</i>	<i>20</i>
<i>Ilustración 9. Ejemplo resultado Google Maps - Hybrid.....</i>	<i>20</i>
<i>Ilustración 10. Ejemplo resultado Open Street Map.....</i>	<i>21</i>

1. INTRODUCCIÓN Y OBJETIVOS

La aplicación desarrollada consiste en realizar un visor de rutas. Para ello el usuario introducirá el origen y destino de su trayecto y la aplicación devolverá la ruta en un mapa así como un listado de indicaciones para realizar el trayecto.

El objetivo de la aplicación es realizar el trazado de rutas mediante dos tipos de mapas. Por una parte, realizando la ruta mediante la API de Google Maps y por otra parte mediante Open Street Map.

2. LISTADO LIBRERIAS NECESARIAS

Tkinter: Es un binding de la biblioteca gráfica Tcl/Tk para el lenguaje de programación Python. Se considera un estándar para la interfaz gráfica de usuario (GUI) para Python.

os: Nos permite acceder a funcionalidades dependientes del Sistema Operativo. Sobre todo, aquellas que nos refieren información sobre el entorno del mismo y nos permiten manipular la estructura de directorios.

tkMessageBox: El módulo se utiliza para mostrar cuadros de mensaje en las aplicaciones. Este módulo proporciona una serie de funciones que se puede utilizar para mostrar un mensaje al usuario.

GoogleMaps: Librería de Google Maps con las funciones "Geocoding", "Reverse Geocoding", "Local search" y la empleada en la aplicación "Directions" la cual devuelve paso a paso las direcciones, distancia, tiempo, etc desde el punto A al punto B.

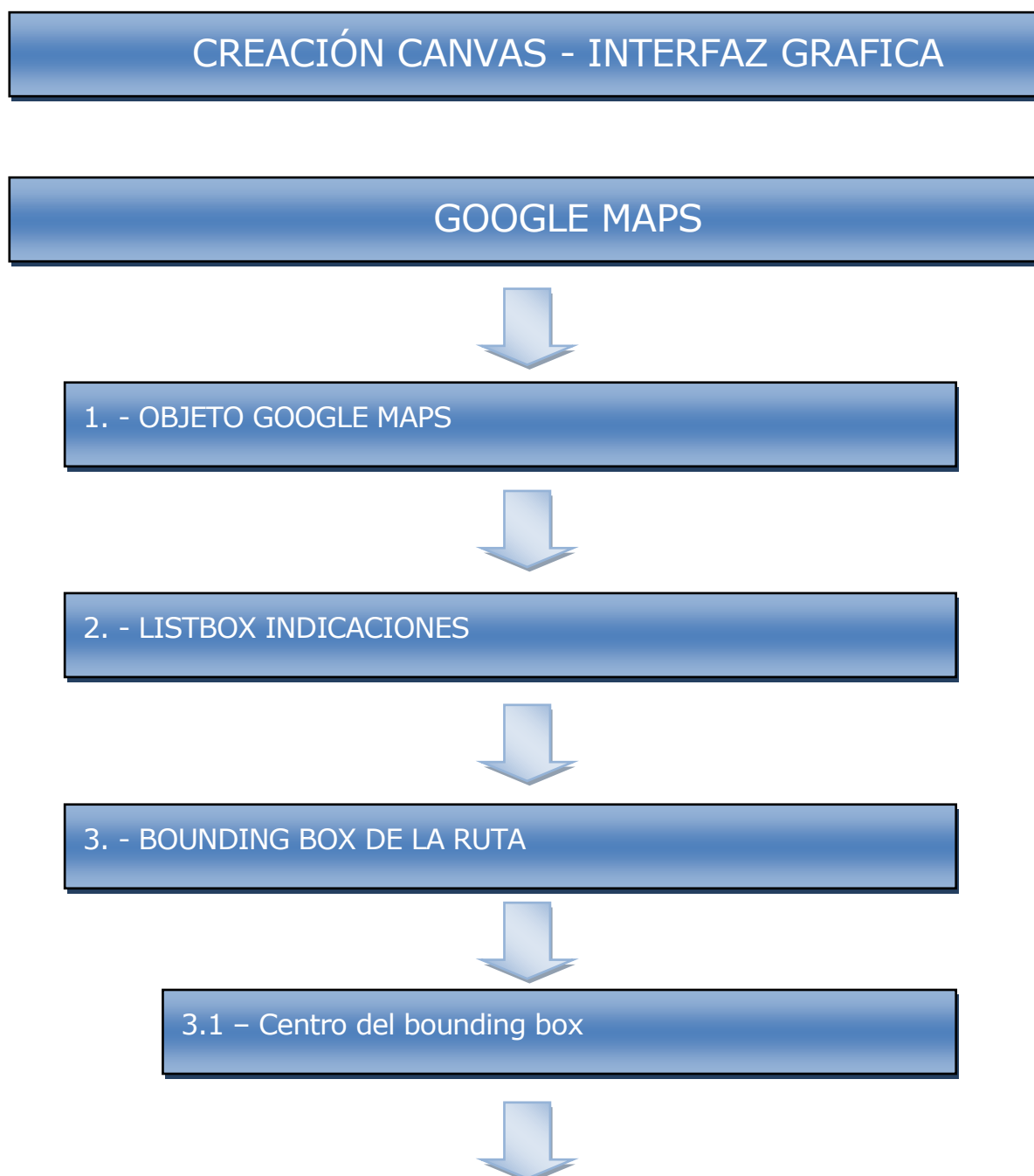
Math: Proporciona acceso a funciones matemáticas.

Urllib: Es un módulo de Python para ir a buscar las URL (localizador uniforme de recursos). Dispone de una interfaz muy sencilla, en la forma de la función urlopen. Esta es capaz de ir a buscar direcciones URL utilizando una variedad de diferentes protocolos.

cStringIO: Este módulo implementa una clase de tipo fichero, StringIO, que lee y escribe un búfer de cadena (también conocido como archivos de memoria).

ImageTk: El modulo contiene soporte para crear y modificar objetos Tkinter BitmapImage y PhotoImage de imágenes PIL.

3. ESQUEMA FLUJO DE TRABAJO



4. – CREACIÓN DEL MAPA



4.1 – Conceptos



4.2 – Calculo de la URL Static Map



4.3 – Guardado y carga de la imagen



5. – DESPLAMIENTO DEL MAPA



6. – ZOOM DEL MAPA



4. DESARROLLO

4.1. CREACIÓN CANVAS - INTERFAZ GRAFICA

De entrada, para crear nuestra aplicación de visor de rutas se debe crear la apariencia de nuestra aplicación. Para ello, se emplearan diversos objetos de la librería Tkinter el cual es un estándar para la interfaz gráfica de usuario (GUI) para Python.

Por tanto, el primer paso consiste en la creación del canvas donde se ubicara la el mapa de la ruta. Las dimensiones del canvas es de 640x640, dimensiones del mapa de Google Maps y que más adelante se detallara.

A continuación, creamos los diferentes elementos que forman parte de la aplicación.

Por una parte creamos los diferentes botones de la aplicación como son:

- Boton zoom in y zoom out: Se emplearan para acercar o alejar el mapa.
- Boton calcular mapa: Mostrara por pantalla el mapa seleccionado, asi como la ruta de indicaciones de la ruta.
- Botones desplazamiento: Desplazara el mapa creado en las cuatro direcciones norte, sur, este y oeste.
- Boton exit: La aplicación se cerrara.

Por otra parte, se crearan diferentes widget de botones de opción múltiple (radiobutton), la cual es una manera de ofrecer muchas opciones posibles para el usuario y permite al usuario elegir sólo uno de ellos. Para implementar esta funcionalidad, cada grupo de botones de radio debe estar asociado a la misma variable y cada uno de los botones debe simbolizar un solo valor. Por lo tanto ,los diferentes "radiobutton" creados son:

- Radiobutton de tipo de mapa de google maps: En ella se especifican las opciones de "Roadmap", "Satellite", "Hybrid" y "Terrain"
- Radiobutton de clase de mapa a emplear: En ella se especifican las opciones de "Open Street Map" y "Google maps".
- Radiobutton de desplazamiento: Donde se especifica el tipo de desplazamiento que se quiere emplear en el mapa. Las opciones son pequeño, mediano y grande.

Finalmente, se crean dos "widget entry" de modo que el usuario pueda introducir el origen y destino de la ruta.

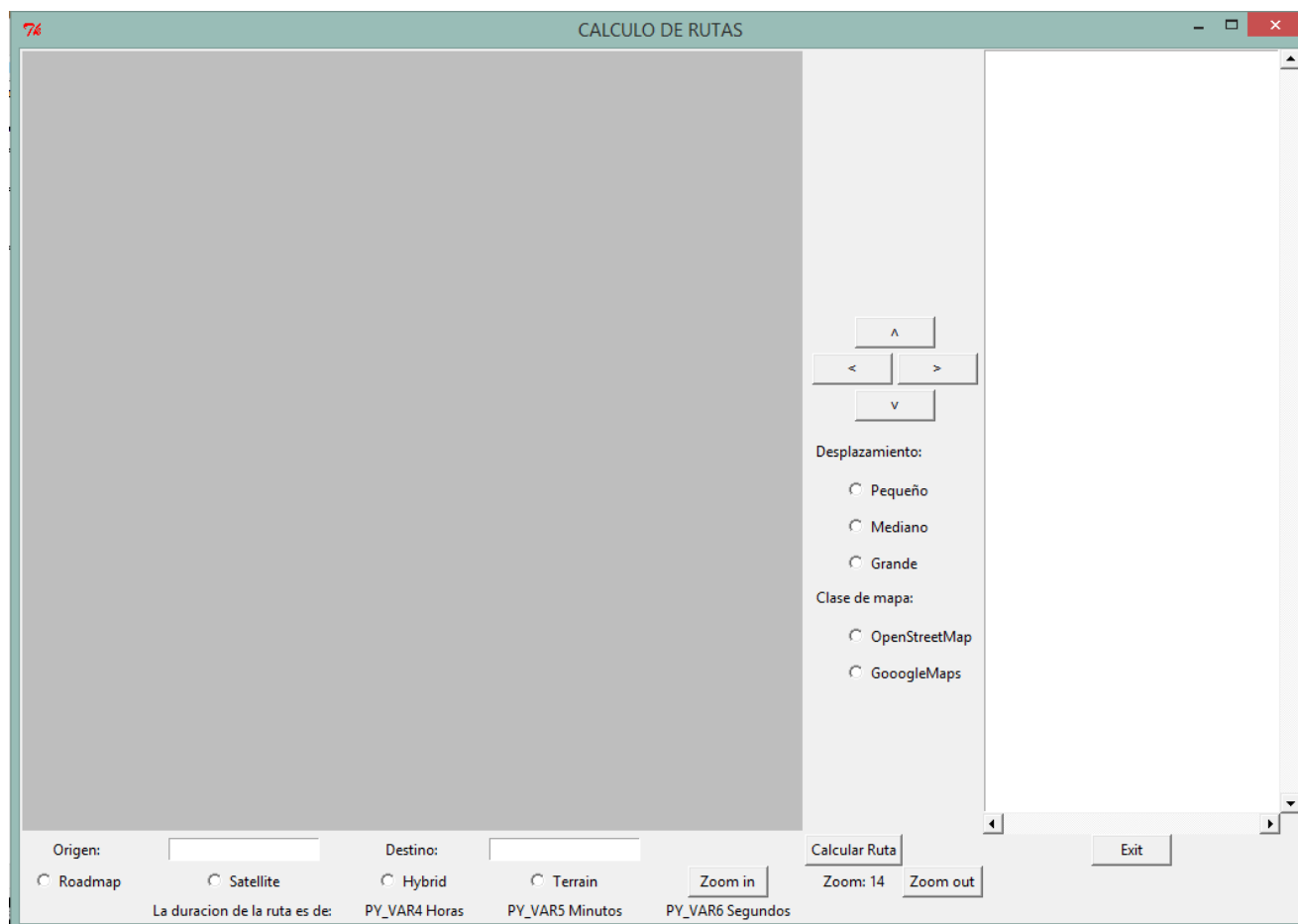


Ilustración 1. Interfaz gráfica de la aplicación.

4.2. GOOGLE MAPS

Una vez implementada la interfaz gráfica de la aplicación se prosigue con la primera de las vertientes del trabajo que es el empleo de mapas procedentes de Google Maps.

4.2.1. OBJETO GOOGLE MAPS

El primer paso consiste en el empleo de la librería de Google Maps para python denominada "googlemaps 1.0.2" en ella encontramos las funciones "Geocoding"(convierte un código postal en una latitud y longitud), "Reverse Geocoding"(encuentra la dirección mas cercana dada una latitud y longitud), "Local search" (encuentra lugares que coincidan con una consulta cerca de un lugar determinado) y la empleada en la aplicación "Directions" la cual devuelve paso a paso las direcciones, distancia, tiempo, etc desde el punto A al punto B.

El empleo de la función "directions" entre dos puntos A y B devuelve una cadena de caracteres como la siguiente:

```
{u'directions': {u'Directions': {u'Duration': {u'seconds': 952, u'html': u'16 mins'}, u'Distance': {u'html': u'9.3&nbsp;km (about 16 m...'}, u'End': {u'coordinates': [-0.375918, 39.404089]}}, u'summaryHtml': u'9.3&nbsp;km (about 16 m...'}, u'Distance': {u'html': u'9.3&nbsp;km', u'meters': 9287}, u'End': {u'coordinates': [-0.375918, 39.404089]}}, u'Duration': {u'seconds': 21, u'html': u'21 secs'}, u'Distance': {u'html': u'260&nbsp;m', u'meters': 260}, u'Point': {u'coordinates': [-0.403403, 39.404089, 0]}}, u'DescriptionHtml': u'Turn right onto <b>Av. Rambleta</b> toward <b>Calle Olivar</b>', u'Point': {u'coordinates': [-0.403403, 39.404089, 0]}}, u'Duration': {u'seconds': 64, u'html': u'1 min'}, u'Distance': {u'html': u'450&nbsp;m', u'meters': 431}, u'DescriptionHtml': u'Continue onto <b>Calle Olivar</b> toward <b>Calle San Vicente Mártir</b>'
```

Ilustración 2. Extracto resultado de la función "directions"

Tal y como se puede observar en la ilustración superior el resultado de la función "directions" es una cadena de caracteres de la cual nos interesan los campos de la duración, las coordenadas y la descripción HTML. El campo de la duración nos indica el tiempo entre los diferentes puntos del recorrido en segundos. Con este campo por lo tanto podremos indicar la duración total de la ruta. Por otra parte, el campo de coordenadas nos indica las coordenadas de los diferentes puntos de la ruta por lo que con este dato se realizara el plotado de la ruta en el mapa de Google maps, como se verá más adelante. Y finalmente, con el campo de la descripción nos aparecen las indicaciones de la ruta y que emplearemos en el listbox.

4.2.2. LISTBOX INDICACIONES

Como se ha comentado anteriormente en el campo de la descripción nos aparecen las indicaciones de la ruta y es en el listbox de la aplicación donde se ubicaran estas indicaciones, previa a una limpieza del contenido de HTML.

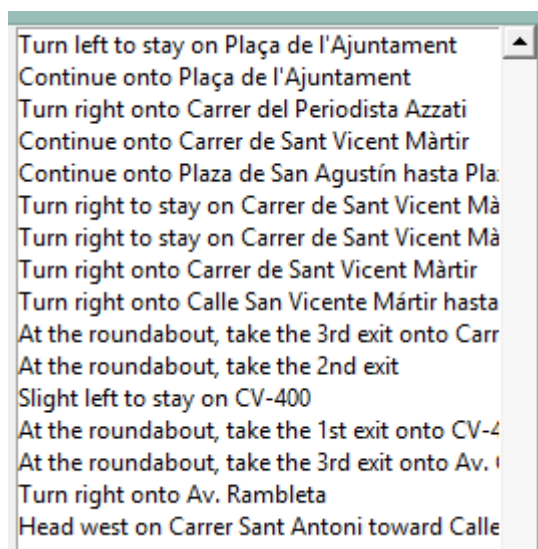


Ilustración 3. Listbox indicaciones.

4.2.3. BOUNDING BOX DE LA RUTA

El bounding box de la ruta consiste en conocer las coordenadas longitud y latitud máximas y mínimas. Para esto, se ha definido una función denominada "bbox" donde los diferentes puntos de la ruta nos determinaran las coordenadas longitud y latitud máximas y mínimas.



Ilustración 4. Bounding box de la ruta.

4.2.3.1. CENTRO DEL BOUNDING BOX

Una vez determinado el bounding box de la ruta podremos determinar cuál es el centro de este bounding box que nos servirá para saber cuál es el punto central del mapa. De modo que para conocer el centro del mapa únicamente dividimos por dos las coordenadas máximas y mínimas de la longitud y latitud.

$$\left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)$$

Ilustración 5. Ecuación del punto medio.

El cálculo de este punto central será fundamental a la hora de emplear los botones de desplazamiento del mapa.

4.2.4. CREACIÓN DEL MAPA

El API de Google Static Maps nos permite insertar imágenes de Google Maps en un sitio web sin utilizar JavaScript ni ningún sistema de carga dinámica de páginas. El servicio de Google Static Maps creará un mapa a partir de los parámetros de URL enviados a través de una solicitud HTTP estándar y generará una imagen de mapa.

Por lo tanto el API de Google Static Maps devuelve una imagen (GIF, PNG o JPEG) en respuesta a una solicitud HTTP a través de una dirección URL. Para cada solicitud se tendrá que especificar la ubicación del mapa, el tamaño de la imagen, el nivel de zoom, el tipo de mapa y la colocación de marcadores opcionales en lugares determinados del mapa.

En resumen los parámetros que podremos incluir en la URL son:

Parámetros de ubicación:

- Center: De carácter obligatorio si no hay marcadores.
- Zoom: De carácter obligatorio si no hay marcadores.

Parámetros de mapa:

- Size: Dimensiones rectangulares de la imagen. Por ejemplo 640x640 pixels
- Scale(opcional): Afecta al número de pixels que se muestran.
- Format: Define el formato de la imagen resultante.(GIF, PNG y PNG)
- Maptype: Define el tipo de mapa que se va a generar.
- Language: Define el idioma que se debe utilizar para mostrar las etiquetas de los mosaicos de mapas.
- Region: Define los límites apropiados que se deben mostrar en función de determinados factores geográficos y políticos.

Parámetros de recurso:

- Markers(opcional): Define uno o varios marcadores para adjuntarlos a la imagen en ubicaciones especificadas.
- Path(opcional): Define una única ruta de dos o más puntos conectados para superponerla en la imagen en ubicaciones especificadas.
- Visible(opcional): Especifica una o varias ubicaciones que deben estar visibles en el mapa.
- Style (opcional): Define un estilo personalizado para alterar la presentación de un recurso concreto (carretera, parque, etc.) del mapa.

Parámetros de notificación:

- Sensor(obligatorio): Especifica si la aplicación que solicita el mapa estático va a utilizar un sensor para determinar la ubicación del usuario

4.2.4.1. CALCULO URL STATIC MAP

El API de Google Static Maps es relativamente fácil de usar, ya que está formada solo por una URL con parámetros. A continuación, se explica cómo utilizar estos parámetros para construir direcciones URL.

Las URL del API de Google Static Maps deben tener el siguiente formato:

<http://maps.googleapis.com/maps/api/staticmap?parameters>

El primer parámetro que hemos incluido en la URL es el centro, para ello la latitud y longitud del centro se definen mediante números dentro de una cadena de texto separado con comas de seis posiciones decimales de precisión. Así la URL quedara:

<http://maps.google.com/maps/api/staticmap?center=42.950827,-122.108974>

El segundo parámetro que se ha definido es el zoom. Los mapas de Google Maps tienen un número entero como "nivel de zoom" que define la resolución de la vista actual. En la vista de mapas de carretera se admiten los niveles de zoom entre 0 (el más bajo, donde todo el planeta se puede ver en un mapa) y 22+ (en el que se ven edificios individuales). En caso de no indicar el nivel de zoom el servidor de Google Static Maps generará automáticamente una imagen que incluirá los marcadores proporcionados (este caso es el que se genera en nuestra aplicación la primera vez que generamos el mapa). Por lo tanto, para los otros casos la URL quedara.

<http://maps.google.com/maps/api/staticmap?center=42.950827,-122.108974&zoom=12>

A continuación, definimos los parámetros de tamaño y formato de la imagen. Para nuestra aplicación se ha empleado un tamaño de 640x640 pixels y un formato de imagen jpeg. En la siguiente tabla se observa los diferentes tamaños que se pueden emplear por la API de Google Maps, según el tipo de API empleado:

API	scale=1	scale=2	scale=4
Gratuita	640x640	640x640 (devuelve 1.280x1.280 píxeles)	No disponible
API de Google Maps for Business	2048x2048	1024x1024 (devuelve 2.048x2.048 píxeles)	512x512 (devuelve 2.048x2.048 píxeles)

Por lo tanto, la URL quedara:

<http://maps.google.com/maps/api/staticmap?center=42.950827,-122.108974&zoom=12&size=640x640&format=jpg>

El siguiente parámetro incluido en la URL es el tipo de mapa empleado. Los cuales son: "Roadmap", "Satellite", "Hybrid" y "Terrain". Según la elección del tipo de mapa, por parte del usuario la URL quedara:

<http://maps.google.com/maps/api/staticmap?center=39.4369485,-0.3924805&zoom=12&size=640x640&format=jpeg&maptype=roadmap>

Después añadimos los "markers" de origen y destino de la ruta. Los "markers" definen un conjunto de uno o varios marcadores en un conjunto de ubicaciones. De modo que la URL generada es la siguiente:

<http://maps.google.com/maps/api/staticmap?center=39.4369485,-0.3924805&zoom=12&size=640x640&format=jpeg&maptype=roadmap&markers=color:blue|label:A|39.404089,-0.403403&markers=color:blue|label:B|39.469174,-0.376037>

Finalmente, para dibujar la ruta empleamos el parámetro "path". Para ese parámetro son necesarios las coordenadas del recorrido de la ruta el cual se obtiene del objeto directions, comentado en el apartado 4.2.1. En conclusión, la URL final quedara:

<http://maps.google.com/maps/api/staticmap?center=39.4369485,-0.3924805&size=640x640&format=jpeg&markers=color:blue|label:A|39.404089,-0.403403&markers=color:blue|label:B|39.469174,-0.376037&path=color:0x0000ff|weight:5|39.404089,-0.403403|39.40476,-0.406314|39.409704,-0.405013|39.410594,-0.408924|39.440275,-0.395103|39.441696,-0.390339|39.441991,-0.388327|39.445866,-0.386579|39.447091,-0.386096|39.448083,-0.385739|39.466065,-0.381448|39.468189,-0.379737|39.468818,-0.379441|39.469808,-0.378642|39.469248,-0.376834|39.469174,-0.376037>

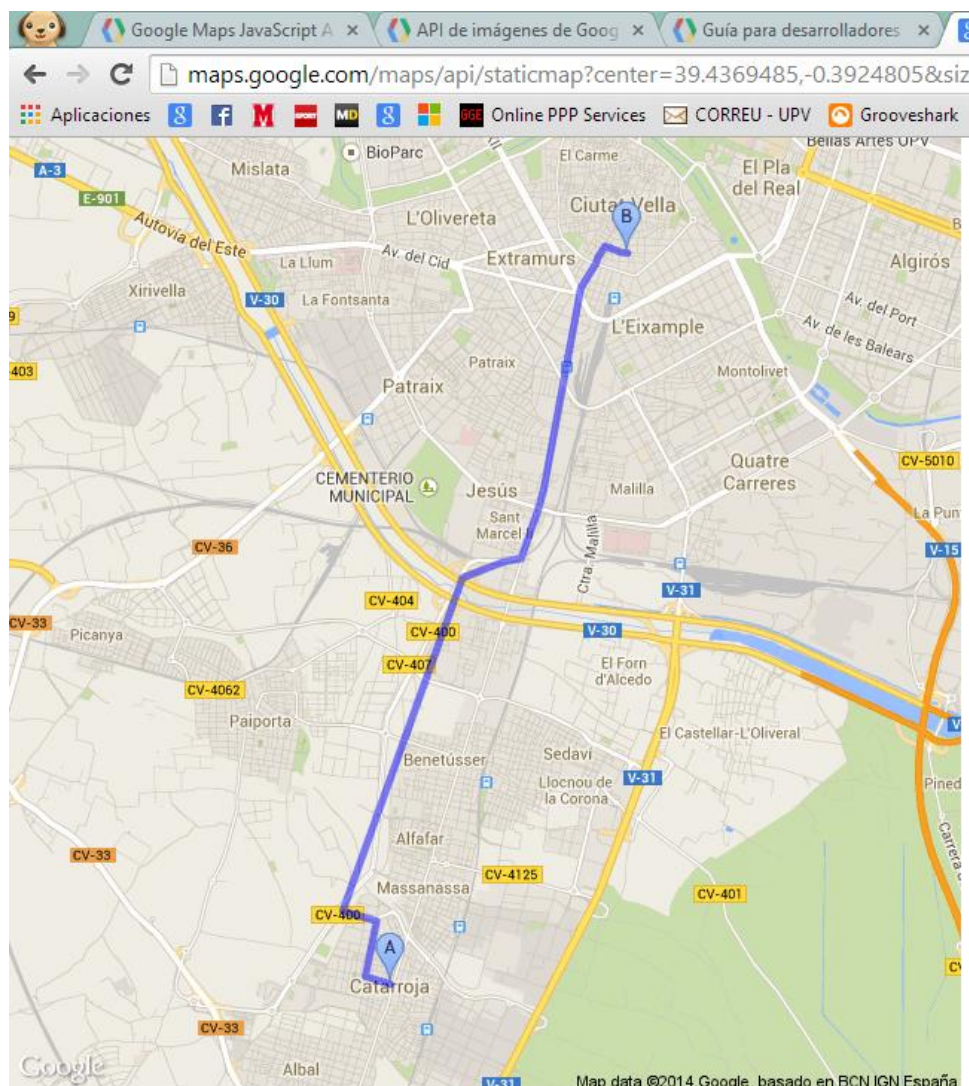


Ilustración 6. Ejemplo gráfico de la URL.

4.2.4.2. GUARDADO Y CARGA DE LA IMAGEN

Una vez obtenida la URL del mapa deseado de Google Maps deberemos descargarla, guardarla y cargarla en nuestra aplicación. Para el paso de descarga y guardado de la imagen empleamos la librería "urllib" junto con el método "urlretrieve".

Una vez obtenida la imagen procederemos a cargarla en nuestra aplicación, Para poder cargar la imagen empleamos la librería "Image" junto con el método "Photoimage". De modo, que en un primer paso se asocia la imagen al objeto "Photoimage" y a continuación, se carga la imagen asociándola al canvas de la aplicación.

4.2.5. DESPLAZAMIENTO DEL MAPA

Para poder desplazarse por el mapa se crean cuatro botones que permitirán desplazarse en las direcciones norte, sur, este y oeste. También es necesario cuantificar el desplazamiento a realizar, para ello mediante la opción "Radiobutton" elegida por el usuario se cuantificara este desplazamiento.

Para obtener una imagen desplazada en el sentido elegido por el usuario, se volverá a obtener una nueva URL pero con un nuevo centro que estará desplazado con respecto a la imagen inicial. Finalmente, se vuelve a realizar el apartado de guardado y carga de imagen (4.2.4.2).

4.2.6. ZOOM DEL MAPA

El zoom del mapa queda implementado mediante dos botones, donde el usuario podrá elegir el zoom que considera más apropiado. La primera vez que se muestra el mapa el zoom se genera por defecto, gracias a Google Maps. El rango del zoom oscila entre 0 y 22. Por lo que, cuando el usuario elige el zoom que desea este parámetro se incluye en la URL de generación de la imagen.

4.3. OPEN STREET MAP

Prosiguiendo con la segunda vertiente del trabajo, la cual es el empleo de mapas procedentes de Open Street Map. En esta ocasión como se verá más adelante el objetivo es realizar un visor de rutas pero en este caso empleando Open Street Map. A diferencia que en el caso anterior (Google Maps), en este caso se va a realizar una cache de los tiles de Open Street Map empleados.

4.3.1. CALCULO DEL TILE CENTRAL

El primer paso para la generación de mapas de Open Street Map consiste en conocer los Tiles que van a formar parte de nuestro mapa. Para nuestra aplicación vamos a utilizar 9 Tiles, es decir el tile central y los que lo rodean.

Como conocemos las coordenadas del punto central del mapa (extraído del apartado de Google Maps) podemos calcular el Tile central del mapa, para ello empleamos la formulación siguiente:

$$n = 2^{\text{zoom}}$$

$$x = \left(\frac{\lambda + 180}{360} \right) \cdot n$$

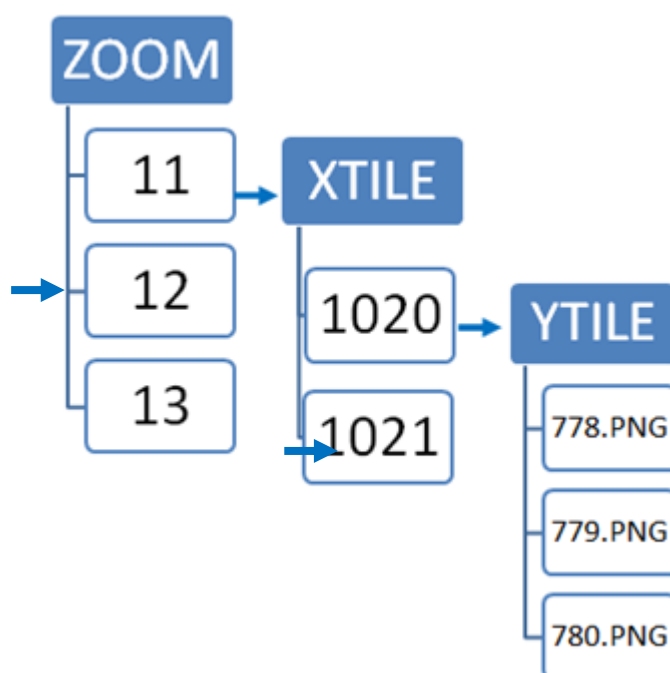
$$y = \left[\frac{1 - \frac{\log(\tan \phi + \frac{1}{\cos \phi})}{\pi}}{2} \right] \cdot n$$

Ilustración 7. Formulación Tiles.

4.3.2. CREACIÓN DE LA CACHE

En esta parte del desarrollo se va a emplear la librería os, librería la cual nos permite manipular la estructura de directorios.

El primer paso para la creación de la cache consiste en la creación de las carpetas de ubicación de los tiles. El formato de la cache es el siguiente:



La primera carpeta que se crea, si no existe, es la carpeta relacionada con el zoom,. Después dentro de la carpeta zoom correspondiente se deben ubicar las carpetas que nos indica el XTILE, de igual modo que en el caso del zoom, se creara en caso de no existir tales carpetas.

Finalmente, dentro de la carpeta XTILE se ubicaran los tiles con el nombre YTILE que le corresponde.

4.3.3. GUARDADO Y CARGA DEL MAPA

Para poder obtener los tiles de Open Street Map, debemos de generar la URL que nos proporciona el Tile deseado.

El formato de la URL es el siguiente:

<http://a.tile.openstreetmap.org/Zoom/xTILE/yTILE.png>

El proceso consiste en generar las nueve URL de los Tiles. Se conoce el Xtile e Ytile del centro del mapa (4.3.1). Para conocer los tiles de alrededor hay que sumar 1 o restar, según la zona. En cuanto al zoom, es el usuario el que elige el valor en la aplicación.

4.3.4. DESPLAZAMIENTO DEL MAPA

Para poder desplazar el mapa se deberán cargar los distintos tiles según la dirección indicada por el usuario. Para ello, se ha implementado un contador para cada una de las cuatro direcciones (norte, sur, este y oeste) con la finalidad de acceder a los diferentes tiles necesarios. Estos contadores se inicializaran a cero, cada vez que se calcule una nueva ruta.

5. COMPARATIVA

CARACTERÍSTICAS	-STATIC MAP API URL- GOOGLE MAPS	TILES OSM
Parámetros URL	SI - Limite tamaño URL	NO
TILES	NO	SI
CACHE	IMPOSIBLE	SI
LIMITE USO	25.000 solicitudes	SIN LIMITE
FORMATOS	GIF, PNG O JPEG	PNG Y JPEG
TIPOS DE MAPAS	Roadmap, Satellite, Hybrid y Terrain	Multitud de mapas en diversos servidores

6. RESULTADOS

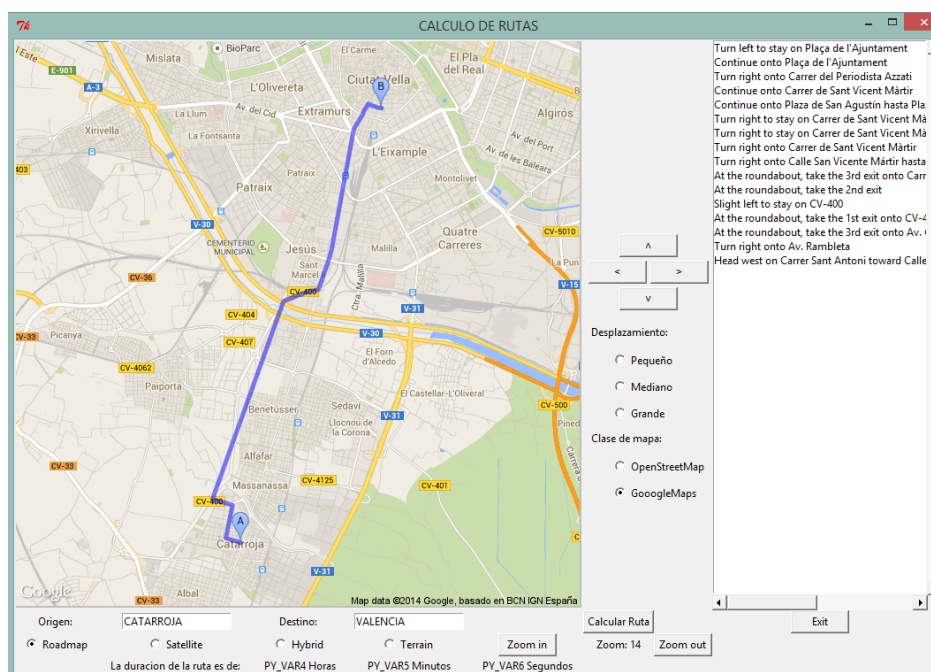


Ilustración 8. Ejemplo resultado Google Maps - Roadmap.

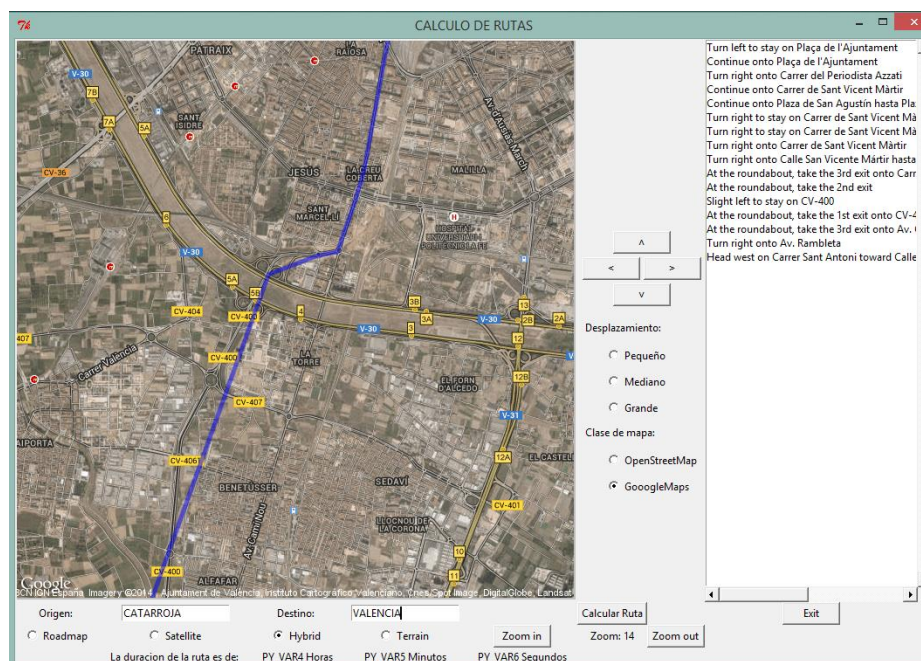
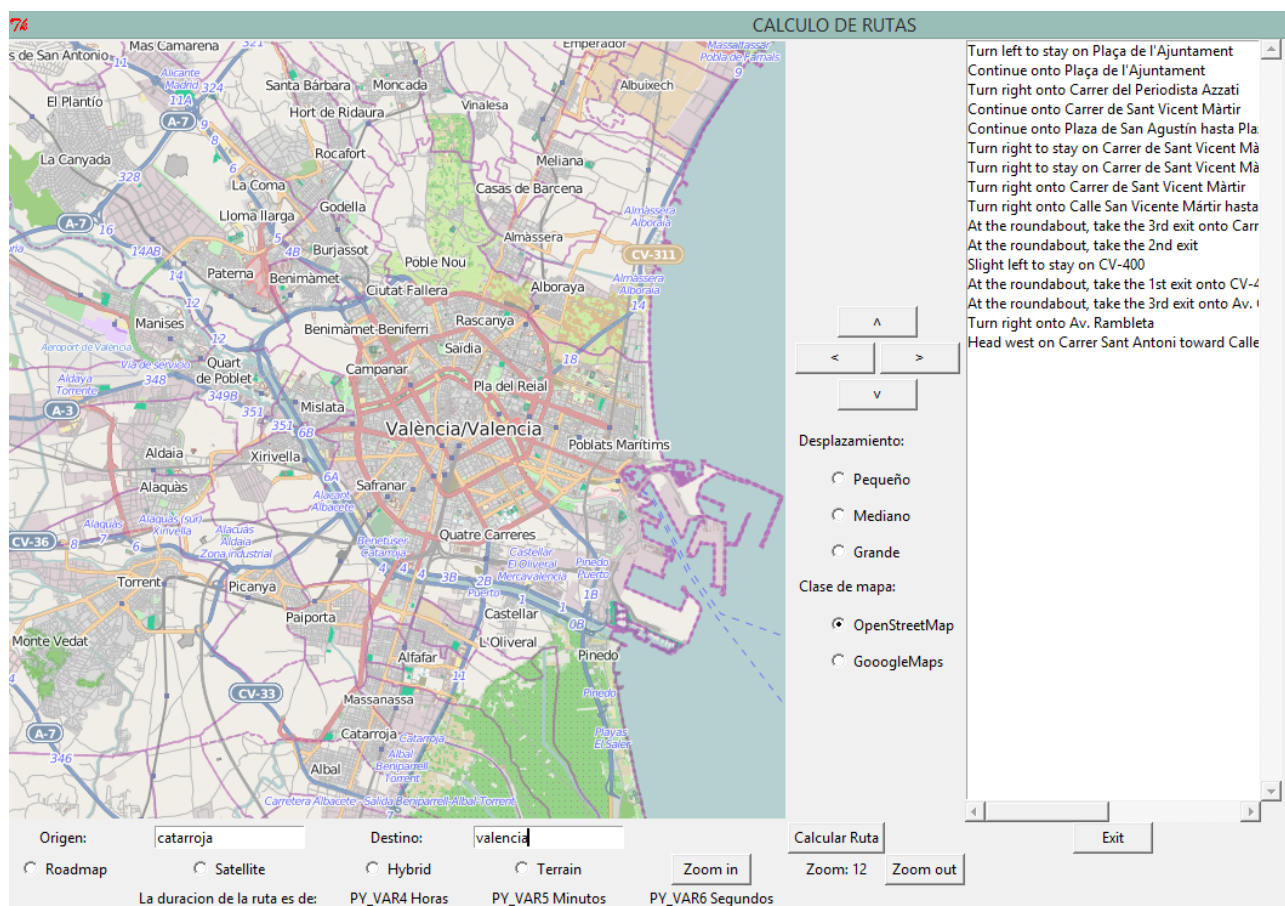


Ilustración 9. Ejemplo resultado Google Maps - Hybrid

CALCULO DE RUTAS



Origien: Destino:

☐ Roadmap
 ☐ Satellite
 ☐ Hybrid
 ☐ Terrain

La duracion de la ruta es de: PY_VAR4 Horas PY_VAR5 Minutos PY_VAR6 Segundos

Desplazamiento:

- ☐ Pequeño
- ☐ Mediano
- ☐ Grande

Clase de mapa:

- ☒ OpenStreetMap
- ☐ GoogleMaps

Turn left to stay on Plaça de l'Ajuntament
Continue onto Plaça de l'Ajuntament
Turn right onto Carrer del Periodista Azzati
Continue onto Carrer de Sant Vicent Màrtir
Continue onto Plaza de San Agustín hasta Pla
Turn right to stay on Carrer de Sant Vicent Mà
Turn right to stay on Carrer de Sant Vicent Mà
Turn right onto Carrer de Sant Vicent Màrtir
Turn right onto Calle San Vicente Mártir hasta
At the roundabout, take the 2nd exit
Slight left to stay on CV-400
At the roundabout, take the 1st exit onto CV-4
At the roundabout, take the 3rd exit onto Carr
Turn right onto Av. Rambleta
Head west on Carrer Sant Antoni toward Calle

Ilustración 10. Ejemplo resultado Open Street Map.



7. CONCLUSIONES

A la vista de los resultados se puede comprobar que el objetivo inicial se ha completado en casi toda la totalidad. Por una parte, en cuanto al empleo del mapa de Google Maps se ha implementado la totalidad de la aplicación, no siendo así para el caso de Open Street Map donde el objetivo de planteado de la ruta no se ha logrado.

Los dos tipos de mapas obtienen el mismo resultado aunque tal y como se ha visto Google Maps presenta limitaciones en su licencia, al contrario que Open Street Map con una licencia completamente abierta. Una ventaja muy importante de Open Street Map reside en la posibilidad de realizar cache.

Por último, es importante resaltar que al contrario de lo que se pudiera pensar la carga de los mapas mediante Google Maps se obtiene en un tiempo muy corto.

8. BIBLIOGRAFÍA

<https://pypi.python.org/pypi/googlemaps/1.0.2>

<http://py-googlemaps.sourceforge.net/>

http://wiki.openstreetmap.org/wiki/Slippy_map_tilenames

<http://sysmagazine.com/posts/133337/>

http://www.perforce.com/perforce/doc.current/manuals/p4script/03_python.html#1128343



9. ANEXOS

9.1. CODIGO

```
# -*- coding: UTF-8 -*-
import Tkinter
import tkFileDialog
import os
import shapefile
import tkMessageBox

from googlemaps import GoogleMaps
import math
import urllib
import cStringIO
import Image
import Image, ImageTk

directorioImagen =
"C://Users/Enric/Desktop/5Enric/Tecnicas_Graficas/Trab
ajo/"

def bbox (pol):
    """Returns the bounding rectangle or bounding box
    of polygon 'pol'."""

    xmin = pol[0][0]
    xmax = pol[0][0]
    ymin = pol[0][1]
    ymax = pol[0][1]

    for pnt in pol:
        if pnt[0] < xmin:
            xmin = pnt[0]
        elif pnt[0] > xmax:
            xmax = pnt[0]

        if pnt[1] < ymin:
            ymin = pnt[1]
        elif pnt[1] > ymax:
            ymax = pnt[1]

        return [xmin,ymin,xmax,ymax]

#Globals...
scr_w = 640
scr_h = 640
descripcion = []

horas = ''
minutos = ''
segundos = ''
contador = 0
origen_viejo = ''
destino_viejo = ''

root = Tkinter.Tk()
root.title('CALCULO DE RUTAS')

def deg2num (lat_deg, lon_deg, zoom):

    lat_rad = math.radians(lat_deg)
    n = 2.0**zoom** elevado

    xtile = int((lon_deg + 180)/360.0 * n)
    ytile = int((1.0 - math.log(math.tan(lat_rad) +
(1.0/math.cos(lat_rad)))) / math.pi)/2.0*n)

    return (xtile,ytile)
```



```
def Calculo_Rutas(e):
    global Latmedia
    global Lonmedia
    global pol
    global horas
    global minutos
    global segundos
    global contador

    try:

        origen_get = origen_text.get()
        destino_get = destino_text.get()
        origen_viejo

        if origen_get == '':
            tkinter.messagebox.showinfo("ERROR", "No has
            introducido el origen")
            if destino_get == '':
                tkinter.messagebox.showinfo("ERROR", "No has
                introducido el destino")
                if origen_get != origen_viejo and origen_viejo
                != '':
                    contador = 0;
                if destino_get != destino_viejo and destino_viejo
                != '':
                    contador = 0;

        #Creamos el objeto de GoogleMaps
        mapService = GoogleMaps()

        #Obtenemos las direcciones que da google
        directions = mapService.directions(origen_get,
        destino_get)
        print directions

        Lat = []
        Lon = []
        pol = []
        duration = []

        ##BORRADO DEL LISTBOX PARA QUE ESTE VACIO CADA VEZ
        Listbox.delete(0, Tkinter.END)

        for step in
        directions['Directions']['Routes'][0]['Steps']:
            b = str(step['Duration'])
            duracion = b[13:17]
            dur = duracion.split(',')
            duration.append(dur)
            suma = 0

            a = str(step['Point'])
            borrado = a[18:len(a)]

            borrado = borrado.replace(',
            0}}', '').replace(' ', '')
            coor = borrado.split(',')

            pol.append(coor)

            des = step['descriptionHtml']

            #LIMPIAMOS LA DESCRIPCION DE HTML

            des =
            des.replace('/', '').replace('<b>', '').replace('<wbr>',
            ' hasta ').replace('<div>', '').replace('<div
            class="google_note">', ' ')

            ##### AÑADIMOS LOS CAMPOS A LA VENTANA DE
            INDICACIONES

            i = 0
            Listbox.insert(i, des)
```




```
i = i + 1

#Calculo de la duracion del viaje
for tiempo in range(0,len(duration)):
    suma = int(duration[tiempo][0]) + suma

horas = int(suma)/3600
minutos = int(suma)/60
segundos = suma%60
print "El tiempo del trayecto es de:"
print horas,minutos,segundos

except:
    pass

# CALCULO DEL BOUNDING BOX DE LA RUTA
Lonmin,Latmin,Lonmax,Latmax = bbox(pol)
print Lonmin,Latmin,Lonmax,Latmax

# CALCULO DEL CENTRO DEL BOUNDING BOX - SERA EL
CENTRO DEL MAPA

Lonmedia = (float(Lonmin) + float(Lonmax))/2
Latmedia = (float(Latmin) + float(Latmax))/2
print Latmedia, Lonmedia

mapa = mapa_google_map(Latmedia, Lonmedia)

## CREACION DEL MAPA

def mapa_google_map(Latmedia, Lonmedia):

    global pol
    global image1
    global src_w
    global src_h
```

```
global desplazamiento
global contador
global xtile
global ytile
global zoom
global ContIzq
global ContDere
global ContArrib
global ContAbaj

ContIzq = 0
ContDere = 0
ContArrib = 0
ContAbaj = 0
```

```
##MAPA DE GOOGLE MAPS
if var3.get() == 1:
```

```
## RUTA EJEMPLO:
```

```
###http://maps.google.com/maps/api/staticmap?center=Emp
ire+State+Building&zoom=18&size=500x500&format=jpeg&ma
ptype=hybrid&sensor=false&
```

```
###http://maps.google.com/maps/api/staticmap?center=42.
950827,-
122.108974&zoom=12&size=500x500&format=jpg&maptype=ter
rain&sensor=false&
```

```
#base URL faltaran añadir mas campos a la ruta
```

```
ruta =
"http://maps.google.com/maps/api/staticmap?"
```

```
#Sumamos el centro a la ruta
ruta += "center="
ruta += str(Latmedia)
ruta += ","
```



```
ruta += str(Lonmedia)
ruta += "&"

if contador>1:
    # Sumamos el ZOOM
    ruta += "zoom="
    ruta += str(zoom)
    ruta += "&"

contador = contador+1
#Sumamos el tamaño y formato de la imagen
ruta += "size=640x640&format=jpeg&"

#Sumamos el tipo de mapa escogido y sensor en
FALSE
if var.get() == 0:
    print "Radiobutton: None selected"
elif var.get() == 1:
    ruta += "maptype=terrain&sensor=false&"
    print "Terrain"
elif var.get() == 2:
    ruta += "maptype=hybrid&sensor=false&"
    print "Hybrid"
elif var.get() == 3:
    ruta += "maptype=satellite&sensor=false&"
    print "Satellite"
elif var.get() == 4:
    ruta += "maptype=roadmap&sensor=false&"
    print "Roadmap"

# Añadimos los Marker
#Ejemplo

#http://maps.google.com/maps/api/staticmap?center=39.4
369485,-
0.3924805&zoom=12&size=500x500&format=jpeg&maptype=roa
dmap&sensor=false&

#markers=color:blue|label:A|39.404089,-
0.403403&

#markers=color:blue|label:B|39.469174,-
0.376037

ruta += "markers=color:blue|label:A|"
ruta += str(pol[0][1])
ruta += ","
ruta += str(pol[0][0])
ruta += "&markers=color:blue|label:B|"
ruta += str(pol[-1][1])
ruta += ","
ruta += str(pol[-1][0])

#Añadimos la ruta pintada
#Ejemplo de ruta con la ruta

#http://maps.google.com/maps/api/staticmap?center=39.4
369485,-
0.3924805&zoom=12&size=500x500&format=jpeg&maptype=roa
dmap&sensor=false&

#path=color:0x0000ff|weight:5|39.404089,-
0.403403|39.469174,-0.376037

ruta += "&path=color:0x0000ff|weight:5|"

for ss in range(0,len(pol)):

    ruta += str(pol[ss][1])
    ruta += ","
    ruta += str(pol[ss][0])
    if ss+1 != len(pol):
        ruta += "|"

##GUARDADO DE LA IMAGEN
urllib.urlretrieve(ruta,
"mapa_google"+"."+ "jpg")
##CARGA DE LA IMAGEN
```



```
imageFile = "mapa_google.jpg"
image1 =
ImageTk.PhotoImage(Image.open(imageFile))
#Obtencion de las dimensiones de la imagen
scr_w = image1.width()
scr_h = image1.height()
print scr_w,scr_h
##
x_ini = 0.0
y_ini = 0.0
#img = Tkinter.PhotoImage(imageFile)

cnvs_rutas.create_image(x_ini,y_ini,image=image1,
anchor='nw')

##MAPA DE OPEN_STREET_MAP
elif var3.get() == 2:
    #ruta =
    http://tile.openstreetmap.org/11/1021/779.png

    ##OBTENCION DE LAS LOS TILES CENTRALES DE
    OPENSTREETMAP
    xtile, ytile = deg2num(Latmedia, Lonmedia,
zoom)
    print xtile, ytile
    print 'Bueno'
    osm = mapa_OSM(xtile, ytile)

    return xtile, ytile
    return ContIzq, ContDere, ContArrib, ContAbaj

def mapa_OSM(xtile, ytile):
    global imageCC
    global imageCR
    global imageCL
    global imageUL
    global imageUC
    global imageUR
```

```
global imageDL
global imageDC
global imageDR
cache = 'C://Cache'
if not os.path.isdir(cache):
    directoryPath = 'C://Cache'
    os.mkdir(directoryPath)

carp_zoom = 'C://Cache/' + str(zoom)
if not os.path.isdir(carp_zoom):
    directoryPath = 'C://Cache/' + str(zoom)
    os.mkdir(directoryPath)

carp_Xtile = str(carp_zoom)+ '/' + str(xtile)
if not os.path.isdir(carp_Xtile):
    directoryPath = str(carp_zoom)+ '/' +
str(xtile)
    os.mkdir(directoryPath)

carp_Xtile_S1 = str(carp_zoom)+ '/' + str(xtile +
1)
if not os.path.isdir(carp_Xtile_S1):
    directoryPath = str(carp_zoom)+ '/' +
str(xtile + 1)
    os.mkdir(directoryPath)

carp_Xtile_R1 = str(carp_zoom)+ '/' + str(xtile -
1)
if not os.path.isdir(carp_Xtile_R1):
    directoryPath = str(carp_zoom)+ '/' +
str(xtile - 1)
    os.mkdir(directoryPath)

## CC (Parte central)
ruta = 'http://a.tile.openstreetmap.org/'
```

```
ruta += str(zoom)
ruta += '/'
ruta += str(xtile)
ruta += '/'
ruta += str(ytile)
ruta += '.png'

if not os.path.isfile('C://Cache/' + str(zoom) +
 '/' + str(xtile) + '/' + str(ytile) + '.png'):
    img = urllib.urlopen(ruta)
    f = open('C://Cache/' + str(zoom) + '/' +
 str(xtile) + '/' + str(ytile) + '.png', 'wb')
    f.write(img.read())
    f.close()

##CARGA DE LA IMAGEN CC
imageFileCC = 'C://Cache/' + str(zoom) + '/' +
 str(xtile) + '/' + str(ytile) + '.png'
imageCC =
ImageTk.PhotoImage(Image.open(imageFileCC))
cnvs_rutas.create_image(192.0,
192.0, image=imageCC, anchor='nw')

## CR (Parte central derecha)
ruta = 'http://a.tile.openstreetmap.org/'
ruta += str(zoom)
ruta += '/'
ruta += str(xtile + 1)
ruta += '/'
ruta += str(ytile)
ruta += '.png'

if not os.path.isfile('C://Cache/' + str(zoom) +
 '/' + str(xtile + 1) + '/' + str(ytile) + '.png'):
    img = urllib.urlopen(ruta)
    f = open('C://Cache/' + str(zoom) + '/' +
 str(xtile + 1) + '/' + str(ytile) + '.png', 'wb')
    f.write(img.read())
    f.close()

##CARGA DE LA IMAGEN CR
```

```
imageFileCR = 'C://Cache/' + str(zoom) + '/' +
 str(xtile + 1) + '/' + str(ytile) + '.png'
imageCR =
ImageTk.PhotoImage(Image.open(imageFileCR))
cnvs_rutas.create_image(448.0,
192.0, image=imageCR, anchor='nw')

## CL (Parte central izquierda)
ruta = 'http://a.tile.openstreetmap.org/'
ruta += str(zoom)
ruta += '/'
ruta += str(xtile - 1)
ruta += '/'
ruta += str(ytile)
ruta += '.png'

if not os.path.isfile('C://Cache/' + str(zoom) +
 '/' + str(xtile - 1) + '/' + str(ytile) + '.png'):
    img = urllib.urlopen(ruta)
    f = open('C://Cache/' + str(zoom) + '/' +
 str(xtile - 1) + '/' + str(ytile) + '.png', 'wb')
    f.write(img.read())
    f.close()

##CARGA DE LA IMAGEN CL
imageFileCL = 'C://Cache/' + str(zoom) + '/' +
 str(xtile - 1) + '/' + str(ytile) + '.png'
imageCL =
ImageTk.PhotoImage(Image.open(imageFileCL))
cnvs_rutas.create_image(-64.0,
192.0, image=imageCL, anchor='nw')

## UL (Parte superior izquierda)
ruta = 'http://a.tile.openstreetmap.org/'
ruta += str(zoom)
ruta += '/'
ruta += str(xtile - 1)
ruta += '/'
ruta += str(ytile - 1)
ruta += '.png'
```



```
if not os.path.isfile('C://Cache/' + str(zoom) +
 '/' + str(xtile - 1) + '/' + str(ytile - 1) +
 '.png'):
    img = urllib.urlopen(ruta)
    f = open('C://Cache/' + str(zoom) + '/' +
 str(xtile - 1) + '/' + str(ytile - 1) + '.png', 'wb')
    f.write(img.read())
    f.close()
##CARGA DE LA IMAGEN UL
imageFileUL = 'C://Cache/' + str(zoom) + '/' +
str(xtile - 1) + '/' + str(ytile - 1) + '.png'
imageUL =
ImageTk.PhotoImage(Image.open(imageFileUL))
cnvs_rutas.create_image(-64.0, -
64.0, image=imageUL, anchor='nw')

## UC (Parte superior central)
ruta = 'http://a.tile.openstreetmap.org/'
ruta += str(zoom)
ruta += '/'
ruta += str(xtile)
ruta += '/'
ruta += str(ytile - 1)
ruta += '.png'

if not os.path.isfile('C://Cache/' + str(zoom) +
 '/' + str(xtile) + '/' + str(ytile - 1) + '.png'):
    img = urllib.urlopen(ruta)
    f = open('C://Cache/' + str(zoom) + '/' +
str(xtile) + '/' + str(ytile - 1) + '.png', 'wb')
    f.write(img.read())
    f.close()
##CARGA DE LA IMAGEN UC
imageFileUC = 'C://Cache/' + str(zoom) + '/' +
str(xtile) + '/' + str(ytile - 1) + '.png'
imageUC =
ImageTk.PhotoImage(Image.open(imageFileUC))
cnvs_rutas.create_image(192.0, -
64.0, image=imageUC, anchor='nw')
```

```
## UR (Parte superior derecha)
ruta = 'http://a.tile.openstreetmap.org/'
ruta += str(zoom)
ruta += '/'
ruta += str(xtile + 1)
ruta += '/'
ruta += str(ytile - 1)
ruta += '.png'

if not os.path.isfile('C://Cache/' + str(zoom) +
 '/' + str(xtile + 1) + '/' + str(ytile - 1) +
 '.png'):
    img = urllib.urlopen(ruta)
    f = open('C://Cache/' + str(zoom) + '/' +
str(xtile + 1) + '/' + str(ytile - 1) + '.png', 'wb')
    f.write(img.read())
    f.close()
##CARGA DE LA IMAGEN UR
imageFileUR = 'C://Cache/' + str(zoom) + '/' +
str(xtile + 1) + '/' + str(ytile - 1) + '.png'
imageUR =
ImageTk.PhotoImage(Image.open(imageFileUR))
cnvs_rutas.create_image(448.0, -
64.0, image=imageUR, anchor='nw')

## DL (Parte inferior izquierda)
ruta = 'http://a.tile.openstreetmap.org/'
ruta += str(zoom)
ruta += '/'
ruta += str(xtile - 1)
ruta += '/'
ruta += str(ytile + 1)
ruta += '.png'

if not os.path.isfile('C://Cache/' + str(zoom) +
 '/' + str(xtile - 1) + '/' + str(ytile + 1) +
 '.png'):
    img = urllib.urlopen(ruta)
```



```
f = open('C://Cache/' + str(zoom) + '/' +  
str(xtile - 1) + '/' + str(ytile + 1) + '.png', 'wb')  
f.write(img.read())  
f.close()
```

```
##CARGA DE LA IMAGEN DL
```

```
imageFileDL = 'C://Cache/' + str(zoom) + '/' +  
str(xtile - 1) + '/' + str(ytile + 1) + '.png'
```

```
imageDL =  
ImageTk.PhotoImage(Image.open(imageFileDL))
```

```
cnvs_rutas.create_image(-64.0,  
448.0, image=imageDL, anchor='nw')
```

```
## ## DC (Parte inferior central)
```

```
ruta = 'http://a.tile.openstreetmap.org/'
```

```
ruta += str(zoom)
```

```
ruta += '/'
```

```
ruta += str(xtile)
```

```
ruta += '/'
```

```
ruta += str(ytile + 1)
```

```
ruta += '.png'
```

```
if not os.path.isfile('C://Cache/' + str(zoom) +  
 '/' + str(xtile) + '/' + str(ytile + 1) + '.png'):
```

```
img = urllib.urlopen(ruta)
```

```
f = open('C://Cache/' + str(zoom) + '/' +  
str(xtile) + '/' + str(ytile + 1) + '.png', 'wb')
```

```
f.write(img.read())
```

```
f.close()
```

```
##CARGA DE LA IMAGEN DC
```

```
imageFileDC = 'C://Cache/' + str(zoom) + '/' +  
str(xtile) + '/' + str(ytile + 1) + '.png'
```

```
imageDC =  
ImageTk.PhotoImage(Image.open(imageFileDC))
```

```
cnvs_rutas.create_image(192.0,  
448.0, image=imageDC, anchor='nw')
```

```
## DR (Parte inferior derecha)
```

```
ruta = 'http://a.tile.openstreetmap.org/'
```

```
ruta += str(zoom)
```

```
ruta += '/'
```

```
ruta += str(xtile + 1)
```

```
ruta += '/'
```

```
ruta += str(ytile + 1)
```

```
ruta += '.png'
```

```
if not os.path.isfile('C://Cache/' + str(zoom) +  
 '/' + str(xtile + 1) + '/' + str(ytile + 1) +  
 '.png'):
```

```
img = urllib.urlopen(ruta)
```

```
f = open('C://Cache/' + str(zoom) + '/' +  
str(xtile + 1) + '/' + str(ytile + 1) + '.png', 'wb')
```

```
f.write(img.read())
```

```
f.close()
```

```
##CARGA DE LA IMAGEN DR
```

```
imageFileDR = 'C://Cache/' + str(zoom) + '/' +  
str(xtile + 1) + '/' + str(ytile + 1) + '.png'
```

```
imageDR =  
ImageTk.PhotoImage(Image.open(imageFileDR))
```

```
cnvs_rutas.create_image(448.0,  
448.0, image=imageDR, anchor='nw')
```

```
def mov_izquierda(e):
```

```
global Latmedia
```

```
global Lonmedia
```

```
global ContIzq
```

```
global xtile
```

```
global ytile
```

```
if var3.get() == 1:
```

```
if var2.get() == 0:
```

```
print "Radiobutton: None selected"
```

```
elif var2.get() == 3:
```

```
Lonmedia = Lonmedia - 0.005
```

```
elif var2.get() == 2:
```

```
Lonmedia = Lonmedia - 0.015
```

```
elif var2.get() == 1:
```



```
Lonmedia = Lonmedia - 0.03

izquierda = mapa_google_map(Latmedia,
Lonmedia)

if var3.get() == 2:

    ContIzq = ContIzq + 1
    #xtile, ytile = deg2num(Latmedia, Lonmedia,
zoom)
    xtile = xtile - ContIzq
    ytile = ytile
    osm = mapa_OSM(xtile, ytile)
    return xtile, ytile

def mov_derecha(e):
    global Latmedia
    global Lonmedia
    global ContDere
    global xtile
    global ytile
    global ContDere

    if var3.get() == 1:
        if var2.get() == 0:
            print "Radiobutton: None selected"
        elif var2.get() == 3:
            Lonmedia = Lonmedia + 0.005
        elif var2.get() == 2:
            Lonmedia = Lonmedia + 0.015
        elif var2.get() == 1:
            Lonmedia = Lonmedia + 0.03

        derecha = mapa_google_map(Latmedia, Lonmedia)

if var3.get() == 2:

    ContDere = ContDere + 1
    #xtile, ytile = deg2num(Latmedia, Lonmedia,
zoom)
    xtile = xtile + ContDere
    ytile = ytile
    osm = mapa_OSM(xtile, ytile)
    return xtile, ytile

def mov_arriba(e):
    global Latmedia
    global Lonmedia
    global ContArrib
    global xtile
    global ytile
    global ContArrib

    if var3.get() == 1:
        if var2.get() == 0:
            print "Radiobutton: None selected"
        elif var2.get() == 3:
            Latmedia = Latmedia + 0.005
        elif var2.get() == 2:
            Latmedia = Latmedia + 0.015
        elif var2.get() == 1:
            Latmedia = Latmedia + 0.03

        arriba = mapa_google_map(Latmedia, Lonmedia)

    if var3.get() == 2:

        ContArrib = ContArrib - 1
        #xtile, ytile = deg2num(Latmedia, Lonmedia,
zoom)
        xtile = xtile
```



```
        ytile = ytile + ContArrib
        osm = mapa_OSM(xtile, ytile)
        return xtile, ytile

def mov_abajo(e):
    global Latmedia
    global Lonmedia
    global ContAbaj
    global xtile
    global ytile
    global ContAbaj

    if var3.get() == 1:
        if var2.get() == 0:
            print "Radiobutton: None selected"
        elif var2.get() == 3:
            Latmedia = Latmedia - 0.005
        elif var2.get() == 2:
            Latmedia = Latmedia - 0.015
        elif var2.get() == 1:
            Latmedia = Latmedia - 0.03

        abajo = mapa_google_map(Latmedia, Lonmedia)

    if var3.get() == 2:

        ContAbaj = ContAbaj + 1
        #xtile, ytile = deg2num(Latmedia, Lonmedia,
zoom)
        xtile = xtile
        ytile = ytile + ContAbaj
        osm = mapa_OSM(xtile, ytile)
        return xtile, ytile

def zoom_in (e):
    global zoom

    global zoom_str

    if zoom < 22:
        zoom = zoom + 1

    zoom_str.set('Zoom: ' + str(zoom))

def zoom_out (e):
    global zoom
    global zoom_str

    if zoom > 0:
        zoom = zoom - 1

    zoom_str.set('Zoom: ' + str(zoom))

def stop_prog(e):
    root.destroy()

##ZOOM..inicializamos las cosas

zoom = 14
zoom_str = Tkinter.StringVar()
zoom_str.set('Zoom: ' + str(zoom))

####CREACION DEL LISTBOX - VENTANA DE INDICACIONES
Listbox = Tkinter.Listbox(root,height=39,width=40)
Listbox.grid(row=2,column=8)

Scrollbar1 = Tkinter.Scrollbar(root)
Scrollbar1.grid(row=2,column=9,sticky='nsew')

Scrollbar1.config(command=Listbox.yview)
Listbox.config(yscrollcommand=Scrollbar1.set)
```




```
Scrollbar2 =  
Tkinter.Scrollbar(root,orient=Tkinter.HORIZONTAL)  
Scrollbar2.grid(row=3,column=8,sticky='nsew')  
  
Scrollbar2.config(command=Listbox.xview)  
Listbox.config(xscrollcommand=Scrollbar2.set)  
  
#LABEL ORIGEN Y DESTINO + DURACION + MOVIMIENTO  
origen = Tkinter.StringVar()  
origen.set('Origen: ')  
  
destino = Tkinter.StringVar()  
destino.set('Destino: ')  
  
duracion = Tkinter.StringVar()  
duracion.set('La duracion de la ruta es de: ')  
  
horas = Tkinter.StringVar()  
horas.set(str(horas) + ' Horas')  
  
minutos = Tkinter.StringVar()  
minutos.set(str(minutos) + ' Minutos')  
  
segundos = Tkinter.StringVar()  
segundos.set(str(segundos) + ' Segundos')  
  
Desplazamiento = Tkinter.StringVar()  
Desplazamiento.set('Desplazamiento: ')  
  
Tipo_mapa = Tkinter.StringVar()  
Tipo_mapa.set('Clase de mapa: ')  
  
#GUI...  
  
cnvs_rutas = Tkinter.Canvas(root, width=scr_w, height=  
scr_h, bg='grey')
```

```
cnvs_rutas.pack()  
  
#CREAMOS LOS BOTONES  
  
btn_ruta = Tkinter.Button(root, text = 'Calcular  
Ruta',width=10)  
btn_exit = Tkinter.Button(root, text = 'Exit',width=8)  
  
btn_zoomin = Tkinter.Button(root, text = 'Zoom  
in',width=8)  
btn_zoomout = Tkinter.Button(root, text = 'Zoom  
out',width=8)  
  
btn_izquierda= Tkinter.Button(root, text =  
'<',width=8)  
btn_izquierda.place(x = 650, y = 250)  
btn_derecha = Tkinter.Button(root, text = '>',width=8)  
btn_derecha.place(x = 720, y = 250)  
btn_arriba = Tkinter.Button(root, text = '^',width=8)  
btn_arriba.place(x = 685, y = 220)  
btn_abajo = Tkinter.Button(root, text = 'v',width=8)  
btn_abajo.place(x = 685, y = 280)  
  
# Label De Origen y Destino y de la duracion de la  
ruta  
  
lbl_origen = Tkinter.Label(root, textvariable =  
origen)  
lbl_destino = Tkinter.Label(root, textvariable =  
destino)  
  
lbl_duracion = Tkinter.Label(root, textvariable =  
duracion)  
lbl_horas = Tkinter.Label(root, textvariable = horas)  
lbl_minutos = Tkinter.Label(root, textvariable =  
minutos)
```



```
lbl_segundos = Tkinter.Label(root, textvariable =  
segundos)
```

```
lbl_desplazamiento = Tkinter.Label(root, textvariable  
= Desplazamiento)
```

```
lbl_mapa = Tkinter.Label(root, textvariable =  
Tipo_mapa)
```

```
# Label del ZOOM - Arriba esta inicializado el zoom
```

```
lbl_zoom = Tkinter.Label(root, textvariable =  
zoom_str)
```

```
## Creamos el texto de entrada de origen y destino
```

```
var_str = str()  
origen_text = Tkinter.Entry(root, textvariable=var_str)  
origen_text.grid(row=6, column=1)
```

```
var_str2 = str()  
destino_text =  
Tkinter.Entry(root, textvariable=var_str2)  
destino_text.grid(row=6, column=3)
```

```
##CREACION DE LOS CHECKBUTTON DEL TIPO DE MAPA
```

```
var = Tkinter.IntVar()
```

```
roadmap = Tkinter.Radiobutton(root, text="Roadmap",  
variable=var, value=4)  
roadmap.grid(row=7, column=0)
```

```
Satellite = Tkinter.Radiobutton(root,  
text="Satellite", variable=var, value=3)  
Satellite.grid(row=7, column=1)
```

```
Hybrid = Tkinter.Radiobutton(root, text="Hybrid",  
variable=var, value=2)
```

```
Hybrid.grid(row=7, column=2)
```

```
Terrain = Tkinter.Radiobutton(root, text="Terrain",  
variable=var, value=1)
```

```
Terrain.grid(row=7, column=3)
```

```
##CREACION DE LOS CHECKBUTTON DEL TIPO DE  
DESPLAZAMIENTO
```

```
var2 = Tkinter.IntVar()
```

```
desp1 = Tkinter.Radiobutton(root, text="Pequeño",  
variable=var2, value=3)
```

```
desp1.place(x = 675, y = 350)
```

```
desp2 = Tkinter.Radiobutton(root, text="Mediano",  
variable=var2, value=2)
```

```
desp2.place(x = 675, y = 380)
```

```
desp3 = Tkinter.Radiobutton(root, text="Grande",  
variable=var2, value=1)
```

```
desp3.place(x = 675, y = 410)
```

```
##CREACION DE LOS CHECKBUTTON DEL TIPO DE MAPA A  
EMPLEAR
```

```
var3 = Tkinter.IntVar()
```

```
OSM = Tkinter.Radiobutton(root, text="OpenStreetMap",  
variable=var3, value=2)
```

```
OSM.place(x = 675, y = 470)
```

```
GM = Tkinter.Radiobutton(root, text="GoogleMaps",  
variable=var3, value=1)
```



```
GM.place(x = 675, y = 500)
```

```
### Packaging
```

```
cnvs_rutas.grid(row = 0, columnspan= 5, rowspan = 5)
```

```
btn_ruta.grid(row=6,column=5)
```

```
lbl_origen.grid(row=6,column=0)
```

```
lbl_destino.grid(row=6,column=2)
```

```
btn_exit.grid(row=6,column=8)
```

```
lbl_duracion.grid(row=8,column=1)
```

```
lbl_horas.grid(row=8,column=2)
```

```
lbl_minutos.grid(row=8,column=3)
```

```
lbl_segundos.grid(row=8,column=4)
```

```
btn_zoomin.grid(row=7,column=4)
```

```
btn_zoomout.grid(row=7,column=6)
```

```
lbl_zoom.grid(row=7,column=5)
```

```
lbl_desplazamiento.place(x = 650, y = 320)
```

```
lbl_mapa.place(x = 650, y = 440)
```

```
## FUNCION DE LOS BOTONES
```

```
btn_zoomin.bind('<Button-1>',zoom_in)
```

```
btn_zoomout.bind('<Button-1>',zoom_out)
```

```
btn_exit.bind('<Button-1>',stop_prog)
```

```
btn_ruta.bind('<Button-1>',Calculo_Rutas)
```

```
btn_izquierda.bind('<Button-1>',mov_izquierda)
```

```
btn_derecha.bind('<Button-1>',mov_derecha)
```

```
btn_arriba.bind('<Button-1>',mov_arriba)
```

```
btn_abajo.bind('<Button-1>',mov_abajo)
```

```
root.mainloop()
```