

# ANEXO I: CÓDIGO FUENTE DE LA APLICACIÓN.





1.	ALMACENA_RUTA_ARCHIVO.M .....	5
2.	ApSEGMMPP.M.....	6
3.	BIN2GRAY.M .....	15
4.	BINARIZA.M.....	16
5.	CALCULA_RIESGO.M .....	17
6.	CALCULA_RUTA.M .....	19
7.	CALCULA_SIGUIENTE_POSICION_ESTRICTO.M .....	24
8.	CARGA_RUTAS.M .....	26
9.	DISTANCIA.M .....	27
10.	EXPANDIR_NODO.M .....	28
11.	GENERA_CAMION.M.....	31
12.	GENERA_FICHEROS_RUTAS.M .....	33
13.	GENERA_MAPA_RIESGO.M .....	36
14.	GENERA_NUEVO_DESTINO.M.....	42
15.	GENERA_RUTA_ORIGENES_DESTINOS.M.....	44
16.	GENERA_VARIOS_CAMIONES.M.....	48
17.	GRAY2RGB.M.....	51
18.	INSERTAR_EN_LISTA.M .....	52
19.	LEE_COORDENADAS.M.....	54
20.	LEE_RUTA_FICHERO.M.....	55
21.	MOSTRAR_MAPAS.M.....	56
22.	MUESTRA_CIUDADES.M .....	67
23.	MUESTRA_RUTA.M .....	70
24.	PRUEBA_META_MAP.M .....	71
25.	SELECCIÓN_VELOCIDAD_TRAMO.M .....	72
26.	SELECCIONA_COLOR.M .....	74
27.	SELECCIONA_NOMBRE.M.....	76



## 1. almacena\_ruta\_archivo.m

```

function []=almacena_ruta_archivo(nfichero,ruta,profundidad)

%-----
%-----
% Función para almacenar una ruta en un fichero
%-----
% SINTAXIS: []=almacena_ruta_archivo(nfichero, ruta,profundidad)
%-----
% INPUT: nfichero: cadena de caracteres con el nombre del fichero a
%         generar
%         ruta: sucesión de puntos [fila,columna] entre dos ciudades
%         profundidad: número de puntos que contiene la ruta a
%                 almacenar
%-----
% OUTPUT:
%-----
%
%                                     Trabajo de Fin de Grado
%         "Sistema de seguimiento y gestión de mercancías peligrosas"
%                                     Diego Cuervo Fernández (16/04/2014)
%                                     Matlab R2013a
%-----
%-----

%Creación del fichero con permiso de lectura y escritura
fileid=fopen(nfichero,'w+');
if(fileid==-1)
    disp(sprintf('Error en la apertura del archivo'));
    return
end

%Se almacena la profundidad en la primera línea del fichero
fprintf(fileid,'%u ',profundidad);
fprintf(fileid,'\n');
i=1;

%Bucle para recorrer la ruta almacenando secuencialmente cada punto
while(i<=profundidad)
    fprintf(fileid,'%u \n%u \n',ruta(i).fila,ruta(i).columna);
    i=i+1;
end

%Se cierra el fichero
fclose(fileid);

```

## 2. ApSegMMPP.m

```
function varargout = ApSegMMPP(varargin)

%-----
%-----
% Interfaz de usuario para arrancar la simulación de camiones en
% tiempo real
%-----
% SINTAXIS: [vargout]=ApSegMMPP(varargin)
%-----
% INPUT:
%-----
% OUTPUT:
%-----
%
%
%                                     Trabajo de Fin de Grado
%      "Sistema de seguimiento y gestión de mercancías peligrosas"
%                                     Diego Cuervo Fernández (20/06/2014)
%                                     Matlab R2013a
%-----
%-----

% Código de inicialización - NO MODIFICAR
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @ApSegMMPP_OpeningFcn, ...
                  'gui_OutputFcn',  @ApSegMMPP_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% Fin del código de inicialización - NO MODIFICAR

% Se ejecuta previamente a mostrar la interfaz gráfica de usuario
function ApSegMMPP_OpeningFcn(hObject, eventdata, handles, varargin)

% Selecciona la línea de comandos como salida por defecto
handles.output = hObject;
```

```

% Actualiza la estructura de elementos de la GUI
guidata(hObject, handles);

% Muestra las imagenes de la GUI
axes(handles.axes3)
imshow('Imagen1.jpg');

% Muestra las señales para cada grupo de mercancías peligrosas
axes(handles.axes5)
imshow('explosivos.png');
axes(handles.axes6)
imshow('gases toxicos.png');
axes(handles.axes7)
imshow('liquidos_inflamables.png');

% Muestra los dibujos para cada grupo de mercancías peligrosas
axes(handles.axes8)
imshow('dibujo_camion_g2.png');
axes(handles.axes9)
imshow('dibujo_camion_g6.png');
axes(handles.axes10)
imshow('dibujo_camion_g16.png');

% Valor inicial del porcentaje de vehículos del grupo 2
pgrupo2=0.2;
set(handles.edit_grupo2, 'String', pgrupo2*100);
set(handles.slider_grupo2, 'Value', pgrupo2);

% Valor inicial del porcentaje de vehículos del grupo 6
pgrupo6=0.2;
set(handles.edit_grupo6, 'String', pgrupo6*100);
set(handles.slider_grupo6, 'Value', pgrupo6);

% Valor inicial del porcentaje de vehículos del grupo 16
pgrupo16=0.6;
set(handles.edit_grupo16, 'String', pgrupo16*100);
set(handles.slider_grupo16, 'Value', pgrupo16);

% Valor inicial del número de camiones de la simulación
num_camiones=150;
set(handles.edit_num_camiones, 'String', num_camiones);
set(handles.slider_num_camiones, 'Value', num_camiones-1);

% Valor inicial del tiempo de actualización
tactualizacion=30;

```

```

set(handles.edit_tiempo, 'String', tactualizacion);
set(handles.slider_tiempo, 'Value', tactualizacion-10);

function varargout = ApSegMMPP_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

% Se ejecuta al mover el deslizador de número de camiones
function slider_num_camiones_Callback(hObject, eventdata, handles)
num_camiones=round(get(hObject, 'Value'))+1;
set(handles.edit_num_camiones, 'String', num_camiones);

% Se ejecuta al crear el deslizador de número de camiones
% (NO MODIFICAR)
function slider_num_camiones_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject, 'BackgroundColor'), ...
...get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', [.9 .9 .9]);
end

% Se ejecuta al mover el deslizador del porcentaje del grupo 2
function slider_grupo2_Callback(hObject, eventdata, handles)
pgrupo2=(round(get(hObject, 'Value')*100))/100;
pgrupo6=get(handles.slider_grupo6, 'Value');
pgrupo16=get(handles.slider_grupo16, 'Value');

%Si pgrupo2 aumenta, disminuye pgrupo16 primero y después pgrupo6
if(pgrupo2+pgrupo6+pgrupo16>1)
    if(pgrupo2+pgrupo6<=1)
        pgrupo16=1-(pgrupo2+pgrupo6);
    else
        pgrupo16=0;
        pgrupo6=1-pgrupo2;
    end
%Si pgrupo2 disminuye, aumenta pgrupo16
elseif(pgrupo2+pgrupo6+pgrupo16<1)
    pgrupo16=1-(pgrupo2+pgrupo6);
end
set(handles.edit_grupo2, 'String', round(pgrupo2*100));
set(handles.edit_grupo6, 'String', round(pgrupo6*100));
set(handles.edit_grupo16, 'String', round(pgrupo16*100));
set(handles.slider_grupo2, 'Value', pgrupo2);
set(handles.slider_grupo6, 'Value', pgrupo6);
set(handles.slider_grupo16, 'Value', pgrupo16);

% Se ejecuta al crear el deslizador del porcentaje del grupo 2
% (NO MODIFICAR)

```



```

function slider_grupo2_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject,'BackgroundColor'), ...
...get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% Se ejecuta al mover el deslizador del porcentaje del grupo 6
function slider_grupo6_Callback(hObject, eventdata, handles)
pgrupo6=(round(get(hObject,'Value')*100))/100;
pgrupo2=get(handles.slider_grupo2,'Value');
pgrupo16=get(handles.slider_grupo16,'Value');

%Si pgrupo6 aumenta, disminuye pgrupo16 primero y después pgrupo2
if(pgrupo2+pgrupo6+pgrupo16>1)
    if(pgrupo2+pgrupo6<=1)
        pgrupo16=1-(pgrupo2+pgrupo6);
    else
        pgrupo16=0;
        pgrupo2=1-pgrupo6;
    end
%Si pgrupo6 disminuye, aumenta pgrupo16
elseif(pgrupo2+pgrupo6+pgrupo16<1)
    pgrupo16=1-(pgrupo6+pgrupo2);
end
set(handles.edit_grupo2,'String',round(pgrupo2*100));
set(handles.edit_grupo6,'String',round(pgrupo6*100));
set(handles.edit_grupo16,'String',round(pgrupo16*100));
set(handles.slider_grupo2,'Value',pgrupo2);
set(handles.slider_grupo6,'Value',pgrupo6);
set(handles.slider_grupo16,'Value',pgrupo16);

% Se ejecuta al crear el deslizador del porcentaje del grupo 6
% (NO MODIFICAR)
function slider_grupo6_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject,'BackgroundColor'), ...
...get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% Se ejecuta al mover el deslizador del porcentaje de vehículos del
grupo 16
function slider_grupo16_Callback(hObject, eventdata, handles)
pgrupo16=(round(get(hObject,'Value')*100))/100;
pgrupo2=get(handles.slider_grupo2,'Value');
pgrupo6=get(handles.slider_grupo6,'Value');

```

```

%Si pgrupo16 aumenta, disminuye pgrupo6 primero y después pgrupo2
if(pgrupo2+pggrupo6+pggrupo16>1)
    if(pgrupo2+pggrupo16<=1)
        pgrupo6=1-(pggrupo2+pggrupo16);
    else
        pgrupo6=0;
        pgrupo2=1-pgrupo16;
    end
%Si pgrupo16 disminuye, aumenta pgrupo6
elseif(pgrupo2+pggrupo6+pggrupo16<1)
    pgrupo6=1-(pggrupo2+pggrupo16);
end
set(handles.edit_grupo2, 'String', round(pgrupo2*100));
set(handles.edit_grupo6, 'String', round(pgrupo6*100));
set(handles.edit_grupo16, 'String', round(pgrupo16*100));
set(handles.slider_grupo2, 'Value', pgrupo2);
set(handles.slider_grupo6, 'Value', pgrupo6);
set(handles.slider_grupo16, 'Value', pgrupo16);

% Se ejecuta al crear el deslizador del porcentaje del grupo 16
% (NO MODIFICAR)
function slider_grupo16_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject, 'BackgroundColor'), ...
...get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', [.9 .9 .9]);
end

% Se ejecuta al pulsar el botón de inicio
function boton_inicio_Callback(hObject, eventdata, handles)
num_camiones=round(get(handles.slider_num_camiones, 'Value'))+1;
pggrupo2=(round(get(handles.slider_grupo2, 'Value')*100))/100;
pggrupo6=(round(get(handles.slider_grupo6, 'Value')*100))/100;
tactualizacion=round(get(handles.slider_tiempo, 'Value'))+10;
genera_varios_camiones(num_camiones,pggrupo2,pggrupo6,tactualizacion);

% Se ejecuta al modificar el cuadro de texto de vehículos del grupo 2
function edit_grupo2_Callback(hObject, eventdata, handles)
pggrupo2=round(str2double(get(hObject, 'String')))/100;
if(pgrupo2<0)
    pgrupo2=0;
elseif(pgrupo2>1)
    pgrupo2=1;
end

pggrupo6=get(handles.slider_grupo6, 'Value');
pggrupo16=get(handles.slider_grupo16, 'Value');

```

```

%Si pgrupo2 aumenta, disminuye pgrupo16 primero y después pgrupo6
if(pgrupo2+pggrupo6+pggrupo16>1)
    if(pgrupo2+pggrupo6<=1)
        pgrupo16=1-(pggrupo2+pggrupo6);
    else
        pgrupo16=0;
        pgrupo6=1-pgrupo2;
    end
%Si pgrupo2 disminuye, aumenta pgrupo16
elseif(pgrupo2+pggrupo6+pggrupo16<1)
    pgrupo16=1-(pggrupo2+pggrupo6);
end
%Movimiento de los deslizadores y corrección de los cuadros de texto
set(hObject,'String',round(pgrupo2*100));
set(handles.edit_grupo6,'String',round(pgrupo6*100));
set(handles.edit_grupo16,'String',round(pgrupo16*100));
set(handles.slider_grupo2,'Value',pggrupo2);
set(handles.slider_grupo6,'Value',pggrupo6);
set(handles.slider_grupo16,'Value',pggrupo16);

% Se ejecuta al crear el cuadro de texto de vehículos del grupo 2
% (NO MODIFICAR)
function edit_grupo2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
...get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% Se ejecuta al modificar el cuadro de texto de vehículos del grupo 6
function edit_grupo6_Callback(hObject, eventdata, handles)
pggrupo6=round(str2double(get(hObject,'String')))/100;
if(pgrupo6<0)
    pgrupo6=0;
elseif(pgrupo6>1)
    pgrupo6=1;
end

pggrupo2=get(handles.slider_grupo2,'Value');
pggrupo16=get(handles.slider_grupo16,'Value');

%Si pgrupo6 aumenta, disminuye pgrupo16 primero y después pgrupo2
if(pgrupo2+pggrupo6+pggrupo16>1)
    if(pgrupo2+pggrupo6<=1)
        pgrupo16=1-(pggrupo2+pggrupo6);
    else

```

```

        pgrupo16=0;
        pgrupo2=1-pgrupo6;
    end
    %Si pgrupo6 disminuye, aumenta pgrupo16
    elseif(pgrupo2+pggrupo6+pggrupo16<1)
        pgrupo16=1-(pggrupo6+pggrupo2);
    end
    set(handles.edit_grupo2, 'String', round(pgrupo2*100));
    set(hObject, 'String', round(pgrupo6*100));
    set(handles.edit_grupo16, 'String', round(pgrupo16*100));
    set(handles.slider_grupo2, 'Value', pgrupo2);
    set(handles.slider_grupo6, 'Value', pgrupo6);
    set(handles.slider_grupo16, 'Value', pgrupo16);

    % Se ejecuta al crear el cuadro de texto de vehículos del grupo 6
    % (NO MODIFICAR)
    function edit_grupo6_CreateFcn(hObject, eventdata, handles)
    if ispc && isequal(get(hObject, 'BackgroundColor'), ...
        ...get(0, 'defaultUicontrolBackgroundColor'))
        set(hObject, 'BackgroundColor', 'white');
    end

    % Se ejecuta al modificar el cuadro de texto del grupo 16
    function edit_grupo16_Callback(hObject, eventdata, handles)
    pgrupo16=round(str2double(get(hObject, 'String')))/100;
    if(pgrupo16<0)
        pgrupo16=0;
    elseif(pgrupo16>1)
        pgrupo16=1;
    end
    pgrupo2=get(handles.slider_grupo2, 'Value');
    pgrupo6=get(handles.slider_grupo6, 'Value');

    %Si pgrupo16 aumenta, disminuye pgrupo6 primero y después pgrupo2
    if(pgrupo2+pggrupo6+pggrupo16>1)
        if(pgrupo2+pggrupo16<=1)
            pgrupo6=1-(pggrupo2+pggrupo16);
        else
            pgrupo6=0;
            pgrupo2=1-pgrupo16;
        end
    %Si pgrupo16 disminuye, aumenta pgrupo6
    elseif(pgrupo2+pggrupo6+pggrupo16<1)
        pgrupo6=1-(pggrupo2+pggrupo16);
    end
end

```

```

set(handles.edit_grupo2, 'String', round(pgrupo2*100));
set(handles.edit_grupo6, 'String', round(pgrupo6*100));
set(hObject, 'String', pgrupo16*100);
set(handles.slider_grupo2, 'Value', pgrupo2);
set(handles.slider_grupo6, 'Value', pgrupo6);
set(handles.slider_grupo16, 'Value', pgrupo16);

% Se ejecuta al crear el cuadro de texto de vehículos del grupo 16
% (NO MODIFICAR)
function edit_grupo16_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'), ...
...get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% Se ejecuta al modificar el cuadro de texto del número de camiones
function edit_num_camiones_Callback(hObject, eventdata, handles)
num_camiones=round(str2double(get(hObject, 'String')));
if(num_camiones>(get(handles.slider_num_camiones, 'Max')+1))
    num_camiones=get(handles.slider_num_camiones, 'Max')+1;
elseif(num_camiones<(get(handles.slider_num_camiones, 'Min')+1))
    num_camiones=get(handles.slider_num_camiones, 'Min')+1;
end
set(hObject, 'String', num_camiones);
set(handles.slider_num_camiones, 'Value', num_camiones-1);

% Se ejecuta al crear el cuadro de texto del número de camiones
% (NO MODIFICAR)
function edit_num_camiones_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'), ...
...get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% Se ejecuta al mover el deslizador del tiempo de actualización
function slider_tiempo_Callback(hObject, eventdata, handles)
tactualizacion=round(get(hObject, 'Value'))+10;
set(handles.edit_tiempo, 'String', tactualizacion);

% Se ejecuta al crear el deslizador del tiempo de actualización
% (NO MODIFICAR)
function slider_tiempo_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject, 'BackgroundColor'), ...
...get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', [.9 .9 .9]);

```

```
end

% Se ejecuta al modificar el cuadro de texto de tiempo de
% actualización
function edit_tiempo_Callback(hObject, eventdata, handles)
tactualizacion=round(str2double(get(hObject,'String')));
if(tactualizacion>(get(handles.slider_tiempo,'Max')+10))
    tactualizacion=get(handles.slider_tiempo,'Max')+10;
elseif(tactualizacion<(get(handles.slider_tiempo,'Min')+10))
    tactualizacion=get(handles.slider_tiempo,'Min')+10;
end
set(hObject,'String',tactualizacion);
set(handles.slider_tiempo,'Value',tactualizacion-10);

% Se ejecuta al crear el cuadro de texto de tiempo de actualización
% (NO MODIFICAR)
function edit_tiempo_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
...get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

### 3. bin2gray.m

```
function[T] = bin2gray (I)

%-----
%-----
% Función para generar una imagen en tonalidades de gris a partir de
% una imagen binaria
%-----
% SINTAXIS: [T]=bin2gray(I)
%-----
% INPUT: I: Matriz correspondiente a la imagen binaria
%-----
% OUTPUT: T: Matriz correspondiente a la imagen en tonalidades de
% gris
%-----
%
%
%                               Trabajo de Fin de Grado
%       "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (11/04/2014)
%                               Matlab R2013a
%-----
%-----

%Almacenamiento del tamaño de la matriz de la imagen de entrada
[fil,col]=size(I);

%Creación de una matriz de enteros de 8 bits del tamaño de la imagen
binaria
T=uint8(zeros(fil,col));

%Se recorre la matriz I y se actualiza la matriz T
for i=1:fil
    for j=1:col

        %Si en la imagen binaria hay un 0 en T habrá color negro
        if(I(i,j)==0)
            T(i,j)=0;
        %Si en la imagen binaria hay un 1 en T habrá color blanco
        else
            T(i,j)=255;
        end
    end
end
end
```

#### 4. binariza.m

```
function[T] = binariza (I)

%-----
%-----
% Función para generar una imagen binaria a partir de una imagen en
% tonalidades de gris
%-----
% SINTAXIS: [T]=binariza(I)
%-----
% INPUT: I: Matriz correspondiente a la imagen en tonalidades de gris
%-----
% OUTPUT: T: Matriz correspondiente a la imagen binaria
%-----
%
%
%                               Trabajo de Fin de Grado
%       "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (05/04/2014)
%                               Matlab R2013a
%-----
%-----

%Almacenamiento del tamaño de la matriz de la imagen de entrada
[fil,col] = size(I);

%Creación de una matriz booleana del tamaño de la imagen binaria
T=logical(zeros(fil,col));

%Se recorre la matriz I y se actualiza la matriz T
for i=1:fil
    for j=1:col
        %Si en I hay un pixel negro, en T le corresponderá un 0
        if(I(i,j)<=0)
            T(i,j)=0;

            %Si el pixel no es negro, en T le corresponde un 1
        else
            T(i,j)=1;
        end
    end
end
end
```



## 5. calcula\_riesgo.m

```
function [riesgo]=calcula_riesgo(camion,mapg2,mapg6,mapg16)

%-----
%-----
% Función para asignar el riesgo que representa una mercancía
% peligrosa de
% un tipo determinado en una posición
%-----
% SINTAXIS: [riesgo]=calcula_riesgo(camion,mapg2,mapg6,mapg16)
%-----
% INPUT: camion: Estructura con los datos de un vehiculo determinado
%         mapg2: Matriz que contiene el riesgo que supone una
%               mercancía del grupo 2 en todos los puntos del mapa
%         mapg6: Matriz que contiene el riesgo que supone una
%               mercancía del grupo 6 en todos los puntos del mapa
%         mapg16: Matriz que contiene el riesgo que supone una
%                mercancía del grupo 16 en todos los puntos del mapa
%-----
% OUTPUT: riesgo: Valor asociado al riesgo que supone el camion, en
%             función de su posición y tipo de mercancía
%-----
%
%                                     Trabajo de Fin de Grado
%             "Sistema de seguimiento y gestión de mercancías peligrosas"
%                                     Diego Cuervo Fernández (23/04/2014)
%                                     Matlab R2013a
%-----
%-----

%En función del tipo de mercancía, se consulta uno u otro mapa
switch camion.grupo
    case 2
        %Asignación del riesgo en función de la posición
        riesgo=mapg2(camion.ruta(camion.indice).fila,...
        ...camion.ruta(camion.indice).columna);
    case 6
        %Asignación del riesgo en función de la posición
        riesgo=mapg6(camion.ruta(camion.indice).fila,...
        ...camion.ruta(camion.indice).columna);
    case 16
        %Asignación del riesgo en función de la posición
        riesgo=mapg16(camion.ruta(camion.indice).fila,...
        ...camion.ruta(camion.indice).columna);
    otherwise
```

```
end          disp(sprintf('Grupo inválido'));
```

## 6. calcula\_ruta.m

```

function [ruta]=calcula_ruta(origen,destino,ficheroTIF,modo)

%-----
%-----
% Función para calcular la ruta más corta entre dos puntos de un
% mapa, empleando un algoritmo de búsqueda A*, y representar la ruta
% calculada en un mapa
%-----
% SINTAXIS: [ruta]=calcula_ruta(origen,destino,ficheroTIF,modo)
%-----
% INPUT: origen: Ciudad de la que parte la ruta
%         destino: Ciudad a la que llega la ruta
%         ficheroTIF: Fichero imagen que contiene el mapa sobre el que
%                   se calculará la ruta
%         modo: Criterio de optimalidad de la ruta
%-----
% OUTPUT: ruta: Sucesión de puntos [fila,columna] óptima entre dos
%          puntos a través de un mapa
%-----
%
%
%                               Trabajo de Fin de Grado
%          "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (08/04/2014)
%                               Matlab R2013a
%-----
%-----

%CONSTANTE FACTOR DE ESCALA CARACTERÍSTICO DEL MAPA
factor_escala=81.709626;

%CONSTANTES CON LAS VELOCIDADES DE CADA TIPO DE CARRETERA (en m/s) Y
FICHERO CARR_TIPO
v_autovia=80/3.6;
v_nacional=60/3.6;
v_comarcal=50/3.6;
v_local=40/3.6;
carr_tipo='Carreteras_tipo.tif';

%Obtencion de coordenadas origen y destino
[fil_ini,col_ini]=lee_coordenadas(origen);
if(fil_ini == -1 || col_ini == -1)
    return
end

[fil_obj,col_obj]=lee_coordenadas(destino);

```

```
if(fil_obj == -1 || col_obj == -1)
    return
end

%Obtencion del parametro a optimizar
if(strcmp(modo,'distancia_minima')==1)
    param=1;
else
    if(strcmp(modo,'tiempo_minimo')==1)
        param=2;
    else
        disp(sprintf('Error en el cuarto argumento de la...
...funcion.\n Debe ser "tiempo_minimo" o...
..."distancia_minima"'));
        return
    end
end

%Se carga el mapa y se transforma en una imagen binaria
mapaRGB=imread(ficheroTIF);
mapa=binariza(mapaRGB);

%Se corrige el desfase por el uso del '0' como índice de la matriz
mapa(1:4981,1:6324)=mapa(2:4982,2:6325);

%Se carga el mapa con los tipos de carreteras
mapa_tipo=imread(carr_tipo);

%Se corrige el desfase por el uso del '0' como índice de la matriz
mapa_tipo(1:4981,1:6324)=mapa_tipo(2:4982,2:6325);

%Almacenamiento del tamaño del mapa
[fila_max,columna_max]=size(mapa);

%Creación de la estructura de datos de los nodos para el algoritmo
nodo(1).profundidad=0;
nodo(1).fila=fil_ini;
nodo(1).columna=col_ini;
nodo(1).padre=0; %Índice del padre del nodo
nodo(1).dist_recorrida=0; %Distancia recorrida en el nodo
nodo(1).tiempo_employado=0;
nodo(1).siguiente=0;

nodos_expandidos=0;

if(param==1)
```

```

        %El coste de un nodo es la suma de la distancia recorrida más
        % la distancia hasta la posición objetivo
        nodo(1).coste=0+distancia(nodo(1).fila,nodo(1).columna,...
        ...fil_obj, col_obj, factor_escal);
    else
        % El coste del nodo será ahora el tiempo empleado más el menor
        % tiempo posible hasta posición objetivo
        nodo(1).coste=0+(distancia(nodo(1).fila,nodo(1).columna,...
        ...fil_obj, col_obj, factor_escal))/v_autovia;
    end

    %Puntero al inicio de la lista
    inicio=1;

    %Indice para recorrer la lista de nodos
    indice=1;

    %Numero de nodos en la lista
    num_nodos=1;

    %Indicador de haber alcanzado el objetivo
    exito=0;

    tic
    %Bucle principal del programa, que se acaba al alcanzar el éxito o al
    %quedarse sin nodos en la lista
    while(exito==0 && indice ~= 0)

        %Se comprueba si el nodo actual es el objetivo

        exito=prueba_meta_map(nodo(indice).fila,nodo(indice).columna...
        ...,fil_obj,col_obj);

        %Si no es el objetivo
        if(exito==0)

            %Llamada a la función de expansión del nodo
            [hijo,num_hijos]=expandir_nodo(nodo(indice),indice, ...
            ...mapa,fila_max,columna_max,factor_escal,mapa_tipo, ...
            ...v_autovia,v_nacional,v_comarcal,v_local);
            nodos_expandidos=nodos_expandidos+1;
            %Bucle para recorrer el vector de hijos
            for j=1:num_hijos

                if(param==1)

```

```

        %Cálculo del coste para cada uno de los hijos
        %(criterio de distancia mínima)
        nuevo_coste=hijo(j).dist_recorrida+...
        ...distancia(hijo(j).fila, ...
        ...hijo(j).columna,fil_obj, ...
        ... col_obj, factor_escala);
    else
        %Cálculo del coste para cada uno de los hijos
        (criterio de tiempo mínimo)

        nuevo_coste=hijo(j).dist_recorrida+...
        ...distancia(hijo(j).fila, ...
        ...hijo(j).columna,fil_obj, ...
        ... col_obj, factor_escala)/v_autovia;
    end

    %Cada uno de los hijos se asigna a un nuevo
    %elemento de la lista de nodos
    nodo(num_nodos+1)=hijo(j);

    %Llamada a la función para insertar en lista el
    %nuevo nodo por orden de coste

    [nodo,inicio,num_nodos]=insertar_en_lista...
    ...(nuevo_coste,nodo,inicio,num_nodos);

    %Se marca con un '0' en el mapa el hijo para
no
    %volver a generarlo
    mapa(hijo(j).fila,hijo(j).columna)=0;
end

    %Variación de índice y de inicio para la siguiente
    %iteracion
    indice=nodo(inicio).siguiente;
    inicio=indice;
end
end
t=toc;

%Una vez acabado el bucle, si se ha alcanzado la posición objetivo
if(exito==1)

    %Cálculo de la distancia total en km
    distancia_total=nodo(indice).dist_recorrida;
    horas_totales=floor(nodo(indice).tiempo_empleado/3600);

```

```

minutos_totales=floor((nodo(indice).tiempo_empleado/...
...3600-horas_totales)*60);
segundos_totales=round(((nodo(indice).tiempo_empleado/...
...3600-horas_totales)*60-minutos_totales)*60);

    %Se muestran los reultados obtenidos por pantalla
    disp(sprintf('Éxito alcanzado'));
    disp(sprintf('Distancia recorrida: %.2f...
... kilometros',distancia_total/1000));
    disp(sprintf('Tiempo empleado: %d horas %d minutos ...
...%.d segundos', horas_totales, minutos_totales, ...
... segundos_totales));
    disp(sprintf('Profundidad de la solucion: %d', ...
...nodo(indice).profundidad));
    disp(sprintf('Nodos expandidos: %d', nodos_expandidos));
    disp(sprintf('Tiempo de ejecución del algoritmo: %.2f ...
...segundos', t));

%Se inicializa el puntero i para almacenar el vector ruta
i=nodo(indice).profundidad+1;
tamano_ruta=i;

%Bucle para almacenar la ruta
while(i>0)

    %Se almacena la ruta optima
    ruta(i).fila=nodo(indice).fila;
    ruta(i).columna=nodo(indice).columna;

    %Variacion del indice para la siguiente iteracion
    indice=nodo(indice).padre;

    %Actualización del puntero
    i=i-1;
end

%Llamada a la función para representar la ruta sobre el mapa
muestra_ruta(ficheroTIF,ruta,tamano_ruta);
else

    %Si no se alcanza el objetivo, se advierte al usuario
    disp(sprintf('Exito no alcanzado :('));
end

```

## 7. calcula\_siguiente\_posicion\_estricto.m

```
function [indice,recorrido] = calcula_siguiente_posicion_estricto...
...(velocidad_kph,tiempo,ruta,indice,recorrido,profundidad)

%-----
%-----
% Función para calcular la siguiente posición de un camión dado una
% velocidad, un tiempo y una posición dentro de una ruta
%-----
% SINTAXIS: [indice,recorrido]=calcula_siguiente_posicion_estricto...
%           ...(velocidad_kph,tiempo,ruta,indice,recorrido,profundidad)
%-----
% INPUT: velocidad_kph: Velocidad del camión en kilómetros por hora
%        tiempo: Tiempo para el cual se calcula la siguiente posición
%        ruta: Ruta a seguir por el camión
%        indice: Posición a cada momento del camión dentro de su ruta
%        recorrido: Distancia que el camión lleva recorrida dentro
%                  del píxel actual
%        profundidad: Número de puntos en la ruta del camión
%-----
% OUTPUT: indice: Posición calculada del camión dentro de su ruta
%         recorrido: Distancia que el camión lleva recorrida dentro
%                  del píxel actual
%-----
%
%                                     Trabajo de Fin de Grado
%      "Sistema de seguimiento y gestión de mercancías peligrosas"
%                                     Diego Cuervo Fernández (19/04/2014)
%                                     Matlab R2013a
%-----
%-----

%Se convierte la velocidad a metros por segundo
velocidad_mps=velocidad_kph/3.6;

%Escala del ráster del mapa (81.709626 metros por fila o columna)
factor_escal=81.709626;

%Se comprueba si el camión avanza en diagonal o vertical/horizontal
if(ruta(indice).fila==ruta(indice+1).fila ||...
...ruta(indice).columna==ruta(indice+1).columna)

    % La distancia al próximo píxel en horizontal o vertical es el
    % factor de escala menos lo ya recorrido.
    restante=1*factor_escal-recorrido;
```



```

else

    % La distancia al próximo píxel en diagonal es el factor de
    % escala por raíz de 2 menos lo que ya llevamos recorrido
    restante=sqrt(2)*factor_escala-recorrido;
end

%Tiempo consumido en avanzar hasta el siguiente píxel
consumido=restante/velocidad_mps;

% Mientras el consumido no supere el tiempo total y no se llegue al
% final de la ruta
while(consumido<=tiempo && indice<profundidad-1);

    %Se avanza al píxel siguiente dentro de la ruta
    indice=indice+1;

    %Se calcula la distancia restante del pixel
    if(ruta(indice).fila==ruta(indice+1).fila ||...
    ...ruta(indice).columna==ruta(indice+1).columna)
        restante=1*factor_escala;
    else
        restante=sqrt(2)*factor_escala;
    end

    %Se actualiza el tiempo consumido
    consumido=consumido+restante/velocidad_mps;
end

%Si se llega al final de la ruta, se devuelve un -1 para indicarlo
if(indice==profundidad-1)
    recorrido=-1;

%Si se acaba el tiempo, se calcula el recorrido restante del píxel
actual
else
    recorrido=restante-(consumido-tiempo)*velocidad_mps;
end

```

## 8. carga\_rutas.m

```
function [coleccion_rutas]=carga_rutas(nfichero,num_ciudades)

%-----
%-----
% Función para cargar los nombres de las rutas desde un fichero para
% emplearlo como una tabla para indexar ficheros de rutas
%-----
% SINTAXIS: [coleccion_rutas]=carga_rutas(nfichero,num_ciudades)
%-----
% INPUT: nfichero: Cadena de caracteres con el nombre del fichero con
%          los nombres de los ficheros de rutas
%          num_ciudades: Número de ciudades orígenes y destinos de las
%          rutas
%-----
% OUTPUT: coleccion_rutas: Colección de nombres de los ficheros que
%          contienen todas las rutas posibles entre
%          las ciudades
%-----
%
%
%                               Trabajo de Fin de Grado
%          "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (17/04/2014)
%                               Matlab R2013a
%-----
%-----

%Apertura del fichero con permiso de sólo lectura
fileid=fopen(nfichero,'r');
if(fileid==-1)
    disp(sprintf('Error en la apertura del archivo'));
    return
end

% Se establece el tamaño del vector en función del número de ciudades
tamano_vector=num_ciudades*(num_ciudades-1);

% Se recorre el fichero y se van almacenando los nombres de los
% ficheros de rutas en la tabla de índice
for i=1:tamano_vector
    coleccion_rutas{i}=fgetl(fileid);
end

%Se cierra el fichero
fclose(fileid);
```

## 9. distancia.m

```

function [dist] = distancia(x1,y1,x2,y2,factor_escal)

%-----
%-----
% Función para calcular la distancia entre dos coordenadas
% cartesianas
%-----
% SINTAXIS: [dist] = distancia(x1,y1,x2,y2,factor_escal)
%-----
% INPUT: x1: Fila 1
%         y1: Columna 1
%         x2: Fila 2
%         y2: Columna 2
%         factor_escal: Factor de escala del mapa
%-----
% OUTPUT: dist: Distancia entre los puntos [x1,y1] y [x2,y2]
%-----
%
%                                     Trabajo de Fin de Grado
%         "Sistema de seguimiento y gestión de mercancías peligrosas"
%                                     Diego Cuervo Fernández (08/04/2014)
%                                     Matlab R2013a
%-----
%-----

% Distancia entre dos puntos mediante el teorema de Pitágoras
dist=sqrt((x1-x2)^2 + (y1-y2)^2)*factor_escal;

```

## 10. expandir\_nodo.m

```
function [hijo,num_hijos] = expandir_nodo (nodo, indice, mapa,...
...fila_max, columna_max, factor_escal, mapa_tipo, v_autovia,...
...v_nacional, v_comarcal, v_local)

%-----
%-----
% Función para expandir un nodo buscando nodos contiguos válidos (un
% '1' en el mapa)
%-----
% SINTAXIS: [hijo,num_hijos] = expandir_nodo(nodo, indice, mapa,...
%         ...fila_max, columna_max, factor_escal, mapa_tipo,...
%         ...v_autovia, v_nacional, v_comarcal, v_local)
%-----
% INPUT: nodo: Nodo a expandir
%         indice: Valor del índice correspondiente al nodo a expandir
%                dentro de la lista de nodos
%         mapa: Mapa binario, con '1' en las carreteras (posiciones
%                válidas) y '0' en el resto (posiciones no válidas)
%         fila_max: Fila máxima del mapa
%         columna_max: Columna máxima del mapa
%         factor_escal: Factor de escala del mapa
%         mapa_tipo: Matriz generada a partir de un mapa raster de la
%                capa tipos de carreteras
%         v_autovia: Velocidad a la que el vehículo circula en una
%                autovía
%         v_nacional: Velocidad a la que el vehículo circula en una
%                carretera nacional
%         v_comarcal: Velocidad a la que el vehículo circula en una
%                carretera comarcal
%         v_local: Velocidad a la que el vehículo circula en una
%                carretera local
%
%-----
% OUTPUT: hijo: Vector que contiene los hijos del nodo expandido
%          num_hijos: Variable con el número de hijos generados
%-----
%
%
%                                     Trabajo de Fin de Grado
%          "Sistema de seguimiento y gestión de mercancías peligrosas"
%                                     Diego Cuervo Fernández (09/04/2014)
%                                     Matlab R2013a
%-----
%-----

%Inicializacion para que devuelva algo en caso de que num_hijos=0
```

```

hijo(1)=nodo;
num_hijos=0;

%Bucle de expansión del nodo
%k y j se emplean para buscar en las 8 posiciones contiguas al nodo
for k= 1:-1:-1
    for j= 1:-1:-1

        %El nodo en sí mismo no es un sucesor
        if(j~=k || k~=0)

            %Si la posición del hijo a estudiar está dentro de las
            %dimensiones del mapa
            if(nodo.columna+k>0 && nodo.fila+j>0 && nodo.columna+k...
            ... <= columna_max && nodo.fila <= fila_max)
                %Si la posición corresponde a un '1' en el mapa, se crea
                %el hijo
                if(mapa(nodo.fila+j,nodo.columna+k)==1)

                    %Incremento del número de hijos
                    num_hijos=num_hijos+1;
                    %Creación del hijo
                    hijo(num_hijos)=nodo;
                    hijo(num_hijos).columna=nodo.columna+k;
                    hijo(num_hijos).fila=nodo.fila+j;
                    hijo(num_hijos).profundidad=nodo.profundidad+1;

                    %El padre del hijo será el índice del nodo expandido
                    hijo(num_hijos).padre=indice;

                    %Se calcula la nueva distancia recorrida en metros
                    dist_nva_iteracion = distancia(...
                    ...hijo(num_hijos).fila, hijo(num_hijos).columna, ...
                    ... nodo.fila, nodo.columna, factor_escalas);

                    hijo(num_hijos).dist_recorrida =...
                    ...nodo.dist_recorrida+dist_nva_iteracion;

                    %Se calcula el tiempo empleado en segundos

                    hijo(num_hijos).tiempo_empleado = ...
                    ...nodo.tiempo_empleado + dist_nva_iteracion ...
                    .../seleccion_velocidad_tramo(nodo.fila, ...
                    ...nodo.columna, mapa_tipo, v_autovia, v_nacional, ...
                    ...v_comarcal, v_local);
                end
            end
        end
    end
end

```

```
        end  
    end  
end
```

11. `genera_camion.m`

```

function [camion] = genera_camion(coleccion_rutas, num_ciudades,...
...pgrupo2, pgrupo6, mapg2, mapg6, mapg16)

%-----
%-----
% Función para generar un camión, al que se asigna un origen y
% destino aleatorios, un grupo de mercancía peligrosa, y una posición
% inicial aleatoria dentro de su ruta
%-----
% SINTAXIS: [camion] = genera_camion(coleccion_rutas, num_ciudades...
%      ..., pgrupo2, pgrupo6,mapg2,mapg6,mapg16)
%-----
% INPUT: coleccion_rutas: Tabla para indexar los ficheros que
%      contienen cada una de las rutas
%      num_ciudades: Número de ciudades posibles como orígenes y
%      destinos
%      pgrupo2: Porcentaje de camiones del grupo 2 que circularán
%      pgrupo6: Porcentaje de camiones del grupo 6 que circularán
%      mapg2: Matriz que contiene el riesgo que supone una
%      mercancía del grupo 2 en todos los puntos del mapa
%      mapg6: Matriz que contiene el riesgo que supone una
%      mercancía del grupo 6 en todos los puntos del mapa
%      mapg16: Matriz que contiene el riesgo que supone una
%      mercancía del grupo 16 en todos los puntos del mapa
%-----
% OUTPUT: camion: Estructura de datos que contiene la ruta del
%      camión, su posición actual, mercancía y riesgo
%-----
%
%      Trabajo de Fin de Grado
%      "Sistema de seguimiento y gestión de mercancías peligrosas"
%      Diego Cuervo Fernández (17/04/2014)
%      Matlab R2013a
%-----
%-----

%En primer lugar se crea el origen y el destino
origen=0;
destino=0;

%Se seleccionan aleatoriamente un origen y un destino diferentes
while(destino==origen)
    origen=randi(num_ciudades);
    destino=randi(num_ciudades);
end

```

```
%Se almacena el número correspondiente a origen y destino
camion.origen=origen;
camion.destino=destino;

%Se genera el índice para obtener el nombre del fichero adecuado de
%la colección de rutas
if(origen<destino)
    indice=(origen-1)*(num_ciudades-1)+destino-1;
else
    indice=(origen-1)*(num_ciudades-1)+destino;
end

%Se muestra el nombre del fichero de la ruta del camión
display(coleccion_rutas{indice});

%Llamada a la función para leer la ruta del fichero correspondiente
[camion.ruta,camion.profundidad] = ...
...lee_ruta_fichero(coleccion_rutas{indice});

%La posición inicial será un punto aleatorio dentro de la ruta
camion.indice=randi(camion.profundidad-1);
camion.recorrido=0;

%Seleccion_grupo es un número real entre 0 y 1
seleccion_grupo=rand;

%En función de seleccion_grupo, se selecciona la mercancía y se
%calcula el riesgo en el mapa adecuado
if(seleccion_grupo<=pgrupo2)
    camion.grupo=2;
    camion.riesgo=mapg2(camion.ruta(camion.indice).fila, ...
    ...camion.ruta(camion.indice).columna);
else
    if(seleccion_grupo<=pgrupo2+pgrupo6)
        camion.grupo=6;
        camion.riesgo=mapg6(camion.ruta(camion.indice).fila, ...
        ...camion.ruta(camion.indice).columna);
    else
        camion.grupo=16;
        camion.riesgo=mapg16(camion.ruta(camion.indice).fila, ...
        ...camion.ruta(camion.indice).columna);
    end
end
end
```



12. `genera_ficheros_rutas.m`

```

function [] = genera_ficheros_rutas(fcol, ffil, fnrutas,...
...ficheroTIF, modo)

%-----
%-----
% Función para generar todas las combinaciones de rutas entre un
% número determinado de ciudades, a partir de un fichero con las
% coordenadas columna de las ciudades, otro con las coordenadas fila,
% un fichero con los nombres de todas las rutas a generar y un
% fichero mapa
%-----
% SINTAXIS: [] = genera_ficheros_rutas(fcol, ffil, fnrutas, ...
%          ...ficheroTIF, modo)
%-----
% INPUT: fcol: Cadena de caracteres con el nombre del fichero que
%          contiene las coordenadas columna de las ciudades
%          ffil: Cadena de caracteres con el nombre del fichero que
%          contiene las coordenadas fila de las ciudades
%          fnrutas: Cadena de caracteres con el nombre del fichero que
%          contiene los nombres de ficheros de ruta a generar
%          ficheroTIF: Cadena de caracteres con el nombre del mapa en
%          formato TIFF
%          modo: Cadena de caracteres con el parámetro a optimizar en
%          el cálculo de la ruta
%-----
% OUTPUT:
%-----
%
%          Trabajo de Fin de Grado
%          "Sistema de seguimiento y gestión de mercancías peligrosas"
%          Diego Cuervo Fernández (16/04/2014)
%          Matlab R2013a
%-----
%-----

%NOMBRE DEL FICHERO CON LOS TIPOS DE CARRETERAS
carr_tipo='Carreteras_tipo.tif';

%Obtencion del parametro a optimizar
if(strcmp(modo,'distancia_minima')==1)
    param=1;
else
    if(strcmp(modo,'tiempo_minimo')==1)
        param=2;
    else
        disp(sprintf('Error en el cuarto argumento de la ...

```

```
        ... funcion.\n Debe ser "tiempo_minimo" o ...
        ..."distancia_minima"'));
    return
end
end

%Se carga la matriz correspondiente al mapa y se binariza
mapaRGB=imread(ficheroTIF);
mapa=binariza(mapaRGB);

%Corrección del mapa por el uso del 0 como índice de la matriz válido
mapa(1:4981,1:6324)=mapa(2:4982,2:6325);

%Se carga el mapa con los tipos de carreteras
mapa_tipo=imread(carr_tipo);

%Se corrige el desfase por el uso del '0' como índice válido de la
matriz
mapa_tipo(1:4981,1:6324)=mapa_tipo(2:4982,2:6325);

%Apertura del fichero de filas
filasID=fopen(ffil,'r');
if(filasID==-1)
    disp(sprintf('Error en la apertura del fichero de filas'));
    return
end

%Lectura del número de ciudades y preparación del vector de filas
i=1;
num_ciudades=str2double(fgetl(filasID));
filas=zeros(num_ciudades);

%Se cargan los datos de las filas en el vector de filas
while ~feof(filasID)
    filas(i)=str2double(fgetl(filasID));
    i=i+1;
end

%Se cierra el fichero de filas
fclose(filasID);

%Apertura del fichero de columnas
columnasID=fopen(fcol,'r');
if(columnasID==-1)
    disp(sprintf('Error en la apertura del fichero de filas'));
    return
end
```

```

end

%Lectura del número de ciudades y preapración del vector de columnas
i=1;
num_ciudades=str2double(fgetl(columnasID));
columnas=zeros(num_ciudades);

%Se cargan los datos de las filas en el vector de columnas
while ~feof(columnasID)
    columnas(i)=str2double(fgetl(columnasID));
    i=i+1;
end

%Se cierra el fichero de columnas
fclose(columnasID);

%Apertura del fichero con los nombres de las rutas
rutasID=fopen(fnrutas,'r');
if(rutasID==-1)
    disp(sprintf('Error en la apertura del fichero de nombres de...
...ficheros de ruta'));
    return
end

%Bucle de servicio, generando la ruta con origen i y destino j
for i=1:num_ciudades
    for j=1:num_ciudades

        %Origen y destino deben ser diferentes
        if j~=i

            %Obtención el nombre del fichero a generar
            nfichero=fgetl(rutasID);

            %Llamada a la función para crear el fichero
            genera_ruta_origenes_destinos(filas(i), ...
            ...columnas(i),filas(j),columnas(j),mapa, ...
            ... mapa_tipo,nfichero,param);

            %Ruta desde i hasta j completada
            disp(sprintf('Ruta desde %d hasta %d completada...
            ..., con nombre %s',i,j,nfichero));
        end
    end
end
end

```

### 13. `genera_mapa_riesgo.m`

```
function[] = genera_mapa_riesgo(camion, ncamiones, ...
...mapa_riesgo_g2, mapa_riesgo_g6, mapa_riesgo_g16, mapaRGB)

%-----
%-----
% Función para generar y mostrar el mapa del riesgo asociado a la
% posición instantánea de un grupo de camiones, en función del tipo
% de mercancía que transporten, así como listas de conflictos de alto
% riesgo
%-----
% SINTAXIS: []=genera_mapa_riesgo(camion,ncamiones,mapa_riesgo_g2,...
% ...mapa_riesgo_g6,mapa_riesgo_g16,mapaRGB)
%-----
% INPUT: camion: Vector que contiene los datos de todos los camiones
%           presentes en la red de carreteras en el momento
%         ncamiones: Número de camiones presentes en la red de
%                   carreteras en el momento
%         mapa_riesgo_g2: Matriz que contiene el riesgo que supone una
%                         mercancía del grupo 2 en cada una de los
%                         puntos del mapa
%         mapa_riesgo_g6: Matriz que contiene el riesgo que supone una
%                         mercancía del grupo 6 en cada una de los
%                         puntos del mapa
%         mapa_riesgo_g16: Matriz que contiene el riesgo que supone
%                          una mercancía del grupo 16 en cada una de
%                          los puntos del mapa
%         mapaRGB: Matriz con el mapa RGB inicialmente en blanco
%-----
% OUTPUT:
%-----
%
%
%                                     Trabajo de Fin de Grado
%         "Sistema de seguimiento y gestión de mercancías peligrosas"
%                                     Diego Cuervo Fernández (24/04/2014)
%                                     Matlab R2013a
%-----
%-----

%CONSTANTE CON EL RADIO DE AFECCIÓN DEL RIESGO Y DE REPRESENTACION
RADIO_RIESGO = 10; %800 metros
RADIO_REPRESENTACION = 35; %2.8 kilómetros
cont_bucle=round(RADIO_REPRESENTACION/sqrt(2));
umbral1=24; %Hasta umbral 1 color azul
umbral2=100; %Hasta umbral 2 color verde
umbral3=150; %Hasta umbral 3 color amarillo
```

```

umbral4=200; %Hasta umbral 4 color naranja
umbral5=288; %A partir de umbral 5 color violeta

%Matrices para almacenar los índices de los conflictos
alerta_naranja=-ones(ncamiones+1,2);
alerta_roja=-ones(ncamiones+1,2);
alerta_violeta=-ones(ncamiones+1,2);

%Contadores de número de conflictos de cada tipo
num_naranja=0;
num_rojo=0;
num_violeta=0;

%En la matriz de riesgo se almacenará el riesgo de cada punto
matrizriesgo=zeros(size(mapa_riesgo_g2));

%En la matriz de conflictos se almacenará el índice del camión que ha
%generado el valor del elemento actual de la matriz de riesgo
matrizconflictos=-ones(size(mapa_riesgo_g2));

%Se recorren cada uno de los camiones actualizando el mapa
for i=1:ncamiones
    %Cada camión sólo puede participar en un conflicto de cada tipo
    conflicto_naranja=0;
    conflicto_rojo=0;
    conflicto_violeta=0;

    %Llamada a la función para calcular el riesgo del camión
    camion(i).riesgo = calcula_riesgo(camion(i), mapa_riesgo_g2, ...
    ...mapa_riesgo_g6, mapa_riesgo_g16);

    %Bucle para actualizar la matriz de riesgo y el mapa RGB
    for k=-cont_bucle:cont_bucle
        for j=-cont_bucle:cont_bucle

            %Si la posición corresponde a carretera (color blanco)
            if(mapaRGB(camion(i).ruta(camion(i).indice).fila+j, ...
            ...camion(i).ruta(camion(i).indice).columna+k,1)~=255 || ...
            ...mapaRGB(camion(i).ruta(camion(i).indice).fila+j, ...
            ...camion(i).ruta(camion(i).indice).columna+k,2)~=255 || ...
            ...mapaRGB(camion(i).ruta(camion(i).indice).fila+j, ...
            ...camion(i).ruta(camion(i).indice).columna+k,3)~=255 )

                %Se actualizan todos los puntos correspondientes a
                %carretera en un radio determinado
                if((abs(j)+abs(k)<=RADIO_RIESGO))

```

```

%Actualización de la matriz de riesgo, añadimos al
%riesgo hasta el momento el riesgo del camión actual

matrizriesgo(camion(i).ruta(camion(i).indice).fila...
...+j, camion(i).ruta(camion(i).indice).columna+k)=...
...matrizriesgo(camion(i).ruta(camion(i).indice). ...
...fila+j, camion(i).ruta(camion(i).indice).columna...
...+k) + camion(i).riesgo;

%Conflictos formados por un solo camión
if(matrizconflictos(camion(i).ruta ...
...(camion(i).indice).fila+j, camion(i).ruta ...
...(camion(i).indice).columna+k) == -1)

%Actualización de matriz de conflictos
matrizconflictos(camion(i).ruta ...
...(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k)=i;

%Comprobación lista naranja
    if(conflicto_naranja==0 && matrizriesgo ...
...(camion(i).ruta(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k) ...
...>umbral3 && matrizriesgo(camion(i).ruta ...
...(camion(i).indice).fila+j, camion(i).ruta ...
...(camion(i).indice).columna+k) <umbral4)
        conflicto_naranja=1;
        num_naranja=num_naranja+1;
        alerta_naranja(num_naranja,1)=i;
        alerta_naranja(num_naranja,3) = matrizriesgo ...
...(camion(i).ruta(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k);

%Comprobación lista roja
elseif(conflicto_rojo==0 && matrizriesgo ...
...(camion(i).ruta(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k) ...
...>umbral4 && matrizriesgo(camion(i).ruta ...
...(camion(i).indice).fila+j, camion(i).ruta ...
...(camion(i).indice).columna+k) <umbral5)
        conflicto_rojo=1;
        num_rojo=num_rojo+1;
        alerta_roja(num_rojo,1)=i;
        alerta_roja(num_rojo,3) = matrizriesgo ...
...(camion(i).ruta(camion(i).indice).fila+j, ...

```

```

...camion(i).ruta(camion(i).indice).columna+k);

%Comprobación lista violeta
elseif(conflicto_violeta==0 && matrizriesgo ...
...(camion(i).ruta(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k) ...
...>umbral5)
    conflicto_violeta=1;
    num_violeta=num_violeta+1;
    alerta_violeta(num_violeta,1)=i;
    alerta_violeta(num_violeta,3) = matrizriesgo ...
...(camion(i).ruta(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k);
end

%Conflictos en los que participan dos camiones
else

    %Comprobación lista naranja
    if(conflicto_naranja==0 && matrizriesgo ...
...(camion(i).ruta(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k) ...
...>umbral3 && matrizriesgo(camion(i).ruta ...
...(camion(i).indice).fila+j, camion(i).ruta ...
...(camion(i).indice).columna+k) <umbral4)
        conflicto_naranja=1;
        num_naranja=num_naranja+1;
        alerta_naranja(num_naranja,1)=i;
        alerta_naranja(num_naranja,2)= ...
...matrizconflictos(camion(i).ruta ...
...(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k);
        alerta_naranja(num_naranja,3) = matrizriesgo ...
...(camion(i).ruta(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k);

    %Comprobación lista roja
elseif(conflicto_rojo==0 && matrizriesgo ...
...(camion(i).ruta(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k) ...
...>umbral4 && matrizriesgo(camion(i).ruta ...
...(camion(i).indice).fila+j, camion(i).ruta ...
...(camion(i).indice).columna+k) <umbral5)
    conflicto_rojo=1;
    num_rojo=num_rojo+1;
    alerta_roja(num_rojo,1)=i;
    alerta_roja(num_rojo,2)= ...

```

```

...matrizconflictos(camion(i).ruta ...
...(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k);
alerta_roja(num_rojo,3) = matrizriesgo ...
...(camion(i).ruta(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k);

%Comprobación lista violeta
elseif(conflicto_violeta==0 && matrizriesgo ...
...(camion(i).ruta(camion(i).indice).fila+j, ...
...camion(i).ruta(camion(i).indice).columna+k) ...
...>umbral5)
    conflicto_violeta=1;
    num_violeta=num_violeta+1;
    alerta_violeta(num_violeta,1)=i;
    alerta_violeta(num_violeta,2)= ...
    ...matrizconflictos(camion(i).ruta ...
    ...(camion(i).indice).fila+j, ...
    ...camion(i).ruta(camion(i).indice).columna+k);
    alerta_violeta(num_violeta,3) = matrizriesgo ...
    ...(camion(i).ruta(camion(i).indice).fila+j, ...
    ...camion(i).ruta(camion(i).indice).columna+k);
end
end

%Se colorea de negro la posición exacta del camión
if(abs(j)+abs(k)<=1)
    mapaRGB(camion(i).ruta(camion(i).indice).fila+j,...
    ...camion(i).ruta(camion(i).indice).columna+k,1)=0;
    mapaRGB(camion(i).ruta(camion(i).indice).fila+j,...
    ...camion(i).ruta(camion(i).indice).columna+k,2)=0;
    mapaRGB(camion(i).ruta(camion(i).indice).fila+j,...
    ...camion(i).ruta(camion(i).indice).columna+k,3)=0;
%Para el resto de puntos, se selecciona un color en
%función del riesgo
else

    if(mapaRGB(camion(i).ruta(camion(i).indice).fila...
    ...+j, camion(i).ruta(camion(i).indice).columna...
    ...+k,1)~=0 || mapaRGB(camion(i).ruta...
    ...(camion(i).indice).fila+j, camion(i).ruta...
    ...(camion(i).indice).columna+k,2)~=0 || ...
    ...mapaRGB(camion(i).ruta(camion(i).indice).fila...
    ...+j, camion(i).ruta (camion(i).indice).columna...
    ...+k,3)~=0 )

```



```

        [mapaRGB(camion(i).ruta...
        ...(camion(i).indice).fila+j, ...
        ...camion(i).ruta(camion(i).indice).columna+k,...
        ...:)] = selecciona_color (matrizriesgo...
        ...(camion(i).ruta(camion(i).indice).fila+j, ...
        ...camion(i).ruta(camion(i).indice).columna+k));
    end
end
%Si no estamos dentro del radio de influencia del camión
else
    if(abs(j)+abs(k)<=RADIO_REPRESENTACION && ...
    ...matrizriesgo(camion(i).ruta...
    ...(camion(i).indice).fila +j, camion(i).ruta...
    ...(camion(i).indice).columna+k)==0)

        mapaRGB(camion(i).ruta(camion(i).indice).fila+j,...
        ...camion(i).ruta(camion(i).indice).columna+k,1)=139;

        mapaRGB(camion(i).ruta(camion(i).indice).fila+j,...
        ...camion(i).ruta(camion(i).indice).columna+k,2)=0;

        mapaRGB(camion(i).ruta(camion(i).indice).fila+j,...
        ...camion(i).ruta(camion(i).indice).columna+k,3)=0;
    end
end
end
end
end
end

%Se muestra el mapa RGB
mostrar_mapas(mapaRGB, camion, num_naranja, alerta_naranja, ...
...num_rojo, alerta_roja, num_violeta, alerta_violeta);

```

#### 14. genera\_nuevo\_destino.m

```
function [nvo_camion] = genera_nuevo_destino(coleccion_rutas, ...
... num_ciudades, camion)

%-----
%-----
% Función para generar un nuevo destino aleatorio para un camión que
% ya ha terminado su ruta anterior, comenzando desde la ciudad en la
% que se encuentra actualmente
%-----
% SINTAXIS: [nvo_camion]= genera_nuevo_destino(coleccion_rutas,...
%          ...num_ciudades,camion)
%-----
% INPUT: coleccion_rutas: Tabla para indexar los ficheros que
%          contienen cada una de las rutas
%          num_ciudades: Número de ciudades posibles como orígenes y
%          destinos
%          camion: Estructura que contiene los datos del camión para el
%          que se va a generar un nuevo destino
%-----
% OUTPUT: nvo_camion: Estructura de contiene los datos de la ruta
%          generada para el nuevo destino del camión
%-----
%
%                               Trabajo de Fin de Grado
%          "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (28/04/2014)
%                               Matlab R2013a
%-----
%-----

%El origen del camión ahora es el destino de la ruta anterior
nvo_camion.origen=camion.destino;
nvo_camion.destino=nvo_camion.origen;

%Se genera nuevo destino aleatoriamente (debe ser distinto que el
origen)
while(nvo_camion.destino==nvo_camion.origen)
    nvo_camion.destino=randi(num_ciudades);
end

%Se genera el índice adecuado para encontrar el nombre del fichero en
%la colección de rutas
if(nvo_camion.origen<nvo_camion.destino)
    indice=(nvo_camion.origen-1)*(num_ciudades-1) +...
        ...nvo_camion.destino-1;
else
```

```

        indice=(nvo_camion.origen-1)*(num_ciudades-1) + ...
        ...nvo_camion.destino;
end

%Se muestra la nueva ruta
display(coleccion_rutas{indice});

%Apertura del fichero que contiene la nueva ruta
fileid=fopen(coleccion_rutas{indice},'r');
if(fileid==-1)
    disp(sprintf('Error en la apertura del archivo'));
    return
end

%Se carga la información de la nueva ruta en la estructura nvo_camion
profundidad=str2double(fgetl(fileid));
for i=1:profundidad
    nvo_camion.ruta(i).fila=str2double(fgetl(fileid));
    nvo_camion.ruta(i).columna=str2double(fgetl(fileid));
end

%Se cierra el fichero
fclose(fileid);

%Se completan el resto de campos de la estructura nvo_camion
nvo_camion.profundidad=profundidad;
nvo_camion.indice=1;
nvo_camion.recorrido=0;
nvo_camion.grupo=camion.grupo;
nvo_camion.riesgo=camion.riesgo;

```

## 15. genera\_ruta\_origenes\_destinos.m

```
function [ruta] = genera_ruta_origenes_destinos(fil_ini, col_ini,...
...fil_obj, col_obj, mapa, mapa_tipo, nfichero, param)

%-----
%-----
% Función para calcular la ruta más corta entre dos puntos de un
% mapa, empleando un algoritmo de búsqueda A*, y guardar la ruta
% calculada en un fichero de datos
%-----
%
% SINTAXIS: [ruta]=genera_ruta_origenes_destinos(fil_ini,col_ini,...
%          ...fil_obj,col_obj,mapa,mapa_tipo,nfichero,param)
%-----
% INPUT: fil_ini: Fila del mapa correspondiente a la posición inicial
%        col_ini: Columna del mapa correspondiente a la posición
%              inicial
%        fil_obj: Fila del mapa correspondiente a la posición
%              objetivo
%        col_obj: Columna del mapa correspondiente a la posición
%              objetivo
%        mapa: Matriz binaria con los datos de las carreteras
%        mapa_tipo: Matriz con los tipos de carreteras
%        nfichero: Nombre del fichero en el que se almacenará la ruta
%        param: Variable de selección del parámetro a optimizar (si
%              vale 1 optimiza distancia, si vale 2 optimiza tiempo)
%-----
% OUTPUT: ruta: Sucesión de puntos [fila,columna] óptima entre dos
%           puntos a través de un mapa
%-----
%
%
%                               Trabajo de Fin de Grado
%           "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (16/04/2014)
%                               Matlab R2013a
%-----
%-----

%CONSTANTE FACTOR DE ESCALA CARACTERÍSTICO DEL MAPA
factor_escala=81.709626;

%CONSTANTES CON LAS VELOCIDADES DE CADA TIPO DE CARRETERA (en m/s) Y
FICHERO CARR_TIPO
v_autovia=80/3.6;
v_nacional=60/3.6;
v_comarcal=50/3.6;
```

```

v_local=40/3.6;

%Almacenamiento del tamaño del mapa
[fil_max,columna_max]=size(mapa);

%Creación de la estructura de datos de los nodos para el algoritmo
nodo(1).profundidad=0;
nodo(1).fila=fil_ini;
nodo(1).columna=col_ini;
nodo(1).padre=0; %Índice del padre del nodo
nodo(1).dist_recorrida=0; %Distancia recorrida en el nodo
nodo(1).tiempo_employado=0;
nodo(1).siguiente=0;

if(param==1)
    %El coste de un nodo es la suma de la distancia recorrida más
    % la distancia hasta la posición objetivo
    nodo(1).coste=0+distancia(nodo(1).fila,nodo(1).columna,...
    ...fil_obj, col_obj, factor_escal);
else
    % El coste del nodo será ahora el tiempo empleado más el menor
    % tiempo posible hasta posición objetivo
    nodo(1).coste=0+(distancia(nodo(1).fila,nodo(1).columna,...
    ...fil_obj, col_obj, factor_escal))/v_autovia;
end

%Puntero al inicio de la lista
inicio=1;

%Índice para recorrer la lista de nodos
indice=1;

%Número de nodos en la lista
num_nodos=1;

%Indicador de haber alcanzado el objetivo
exito=0;

%Bucle principal del programa, que se acaba al alcanzar el éxito o al
%quedarse sin nodos en la lista
while(exito==0 && indice ~= 0)

    %Se comprueba si el nodo actual es el objetivo

    exito=prueba_meta_map(nodo(indice).fila,nodo(indice).columna...
    ...,fil_obj,col_obj);

```

```

%Si no es el objetivo
if(exito==0)

    %Llamada a la función de expansión del nodo
    [hijo,num_hijos]=expandir_nodo(nodo(indice),indice, ...
    ...mapa,fila_max,columna_max,factor_escalas,mapa_tipo, ...
    ...v_autovia,v_nacional,v_comarcal,v_local);
    nodos_expandidos=nodos_expandidos+1;
    %Bucle para recorrer el vector de hijos
    for j=1:num_hijos

        if(param==1)
            %Cálculo del coste para cada uno de los hijos
            %(criterio de distancia mínima)
            nuevo_coste=hijo(j).dist_recorrida+...
            ...distancia(hijo(j).fila, ...
            ...hijo(j).columna,fil_obj, ...
            ... col_obj, factor_escalas);
        else
            %Cálculo del coste para cada uno de los hijos
            (criterio de tiempo mínimo)

            nuevo_coste=hijo(j).dist_recorrida+...
            ...distancia(hijo(j).fila, ...
            ...hijo(j).columna,fil_obj, ...
            ... col_obj, factor_escalas)/v_autovia;
        end

        %Cada uno de los hijos se asigna a un nuevo
        %elemento de la lista de nodos
        nodo(num_nodos+1)=hijo(j);

        %Llamada a la función para insertar en lista el
        %nuevo nodo por orden de coste

        [nodo,inicio,num_nodos]=insertar_en_lista...
        ...(nuevo_coste,nodo,inicio,num_nodos);

        %Se marca con un '0' en el mapa el hijo para
no
        %volver a generarlo
        mapa(hijo(j).fila,hijo(j).columna)=0;
    end
end

```

```

        %Variación de índice y de inicio para la siguiente
        %iteracion
        indice=nodo(inicio).siguiente;
        inicio=indice;
    end
end

%Una vez acabado el bucle, si se ha alcanzado la posición objetivo
if(exito==1)

    %Se inicializa el puntero i para almacenar el vector ruta
    i=nodo(indice).profundidad+1;
    tamano_ruta=i;

    %Bucle para almacenar la ruta
    while(i>0)

        %Se almacena ruta optima
        ruta(i).fila=nodo(indice).fila;
        ruta(i).columna=nodo(indice).columna;

        %Variacion del indice para la siguiente iteracion
        indice=nodo(indice).padre;

        %Actualización del puntero
        i=i-1;
    end

    %Llamada a la función que almacena la ruta en un fichero
    almacena_ruta_archivo(nfichero,ruta,tamano_ruta);
else
    %Si no se alcanza el objetivo, se advierte al usuario
    disp(sprintf('Exito no alcanzado :('));
end

```

16. `genera_varios_camiones.m`

```
function [] = genera_varios_camiones(ncamiones, pgrupo2, pgrupo6, ...  
...tactualizacion)
```

```
%-----  
% Función para generar un número de camiones determinados,  
% actualizando periódicamente su posición mediante la función de  
% seguimiento de rutas, y creando los mapas de riesgo  
%-----  
% SINTAXIS: []=genera_varios_camiones(n_camiones,pgrupo2,pgrupo6,...  
%      ...tactualizacion)  
%-----  
% INPUT: num_camiones: Número de vehículos que se van a generar  
%        pgrupo2: Porcentaje de camiones del grupo 2 que circularán  
%        pgrupo6: Porcentaje de camiones del grupo 6 que circularán  
%        tactualizacion: Periodo de actualización de la posición de  
%                       los vehículos en tiempo real  
%-----  
% OUTPUT:  
%-----  
  
%                               Trabajo de Fin de Grado  
% "Sistema de seguimiento y gestión de mercancías peligrosas"  
%                               Diego Cuervo Fernández (23/04/2014)  
%                               Matlab R2013a  
%-----
```

```
%CONSTANTES CON LAS VELOCIDADES DE CADA TIPO DE CARRETERA (en km/h)
v_autovia=80;
v_nacional=60;
v_comarcal=50;
v_local=40;
```

```
%Semillas para números aleatorios
rand('state',sum(100*clock));
randn('state',sum(100*clock));
```

```
%Se carga la colección de rutas
disp(sprintf('Cargando rutas...\n'));
coleccion_rutas=carga_rutas('Nombres_ficheros.dat',20);
```

```
%Se carga el mapa de tipos de carreteras
disp(sprintf('Cargando mapa de tipos de carreteras...\n'));
mapa_tipo=imread('Carreteras_tipo.tif');
```



```

mapa_tipo(1:4981,1:6324)=mapa_tipo(2:4982,2:6325);

%Se cargan los mapas de riesgo
disp(sprintf('Cargando mapas de riesgo...\n'));
mapa_riesgo_g2=imread('Riesgo_grupo_2.tif');
mapa_riesgo_g2(1:4981,1:6324)=mapa_riesgo_g2(2:4982,2:6325);

mapa_riesgo_g6=imread('Riesgo_grupo_6.tif');
mapa_riesgo_g6(1:4981,1:6324)=mapa_riesgo_g6(2:4982,2:6325);

mapa_riesgo_g16=imread('Riesgo_grupo_16.tif');
mapa_riesgo_g16(1:4981,1:6324)=mapa_riesgo_g16(2:4982,2:6325);

%Se carga el mapa para mostrar
disp(sprintf('Cargando mapa para mostrar...\n'));
mapa_muestra_RGB=imread('mapa_rgb_1.tif');

%Se generan tantos camiones como haya elegido el usuario
for i=1:ncamiones
    camion(i) = genera_camion(coleccion_rutas, 20, pgrupo2, ...
        ...pggrupo6, mapa_riesgo_g2, mapa_riesgo_g6, mapa_riesgo_g16);
end

%Inicializa a cero un temporizador
t=0;
while(1)
    %Arranca un temporizador
    tic
    %Para cada camión se genera la nueva posición
    for i=1:ncamiones
        %Velocidad en función del tramo
        velocidad = seleccion_velocidad_tramo(camion(i).ruta...
            ...(camion(i).indice).fila, camion(i).ruta...
            ...(camion(i).indice).columna, mapa_tipo, v_autovia, ...
            ...v_nacional, v_comarcal, v_local);

        [camion(i).indice,camion(i).recorrido] =...
            ...calcula_siguiete_posicion_estricto(velocidad, ...
            ...t+tactualizacion, camion(i).ruta,camion(i).indice, ...
            ...camion(i).recorrido,camion(i).profundidad);

        %Si se llega al final de la ruta, se genera nuevo destino
        if(camion(i).recorrido<0)
            disp(sprintf('El camion numero %d ha llegado a...
                ...su destino. Cargando nueva ruta',i));

```

```
        camion(i) = genera_nuevo_destino(...
        ...coleccion_rutas, 20, camion(i));
    end
end
disp(sprintf('Nueva posicion calculada.\n'));

%Se llama a la función para generar el mapa de riesgo
genera_mapa_riesgo(camion, ncamiones, mapa_riesgo_g2, ...
...mapa_riesgo_g6, mapa_riesgo_g16, mapa_muestra_RGB);

%Se detiene el temporizador
t=toc;

%Pausa un periodo de actualización
pause(tactualizacion);
end
```

17. `gray2rgb.m`

```
function[T] = gray2rgb (I)
```

```
%-----
%-----
% Función para generar una imagen RGB a partir de una imagen en
% tonalidades de gris
%-----
% SINTAXIS: [T]=gray2rgb(I)
%-----
% INPUT: I: Matriz correspondiente a la imagen en tonalidades de gris
%-----
% OUTPUT: T: Matriz correspondiente a la imagen RGB
%-----
%
%                               Trabajo de Fin de Grado
%       "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (09/04/2014)
%                               Matlab R2013a
%-----
%-----

%Se triplica la matriz gris original para que corresponda a los
%colores [R G B]
T=repmat(I,[1 1 3]);
```

## 18. insertar\_en\_lista.m

```
function[nodo, inicio, num]=insertar_en_lista(nuevo_coste, nodo, ...
... inicio, num)

%-----
%-----
% Función para actualizar una lista enlazada, insertando un nuevo
% nodo cada vez por orden de coste
%-----
% SINTAXIS: [nodo, inicio, num]=insertar_en_lista(nuevo_coste, nodo, ...
%          ... inicio, num)
%-----
% INPUT: nuevo_coste: Coste del nodo a insertar
%        nodo: Vector completo de nodos
%        inicio: Puntero al inicio de la lista enlazada
%        num: Número de nodos en la lista
%-----
% OUTPUT: nodo: Vector completo de nodos actualizado
%         inicio: Puntero al inicio de la lista enlazada
%         num: Número de nodos en la lista
%-----
%
%
%                                Trabajo de Fin de Grado
%        "Sistema de seguimiento y gestión de mercancías peligrosas"
%                                Diego Cuervo Fernández (08/04/2014)
%                                Matlab R2013a
%-----
%-----

%Actualización del número de nodos de la lista
num=num+1;

%Se le asigna al nuevo nodo su coste
nodo(num).coste=nuevo_coste;

%Si nuevo coste es menor que el primero, se inserta en el inicio
if nuevo_coste<nodo(inicio).coste
    nodo(num).siguiente=inicio;
    inicio=num;

%Si la posición de inserción no es la primera, comienza la búsqueda
%recorriendo el vector
else

    %Se comienza a recorrer por el principio
    indice=inicio;
```

```
while(1)
    anterior=indice;
    indice=nodo(indice).siguiente;

    %Si se alcanza el final de la lista sin encontrar la
    % posición de inserción, se inserta al final
    if(indice==0)
        nodo(anterior).siguiente=num;
        nodo(num).siguiente=0;
        break
    end

    %Si se alcanza la posición de inserción, se inserta
    if(nodo(indice).coste > nuevo_coste)
        nodo(anterior).siguiente=num;
        nodo(num).siguiente=indice;
        break
    end
end
end
end
```

## 19. lee\_coordenadas.m

```
function [fila,columna]= lee_coordenadas(nombre_ciudad)

%-----
%-----
% Función para leer las coordenadas de una ciudad de un fichero a
% partir de una cadena de caracteres con su nombre
%-----
% SINTAXIS: [fila,columna]=lee_coordenadas(nombre_ciudad)
%-----
% INPUT: nombre_ciudad: Cadena de caracteres con el nombre de la
%              ciudad
%-----
% OUTPUT: fila: Fila del mapa en la que se encuentra la ciudad
%          columna: Columna del mapa en la que se encuentra la ciudad
%-----
%
%
%                               Trabajo de Fin de Grado
%          "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (12/05/2014)
%                               Matlab R2013a
%-----
%-----

%Se añaden los strings con la ubicación y extensión del fichero
str_ubicacion='./ciudades/';
str_extension='.dat';
nfichero=[str_ubicacion nombre_ciudad str_extension];

%Apertura del fichero que contiene la ruta
fileid=fopen(nfichero,'r');
if(fileid==-1)
    disp(sprintf('Nombre de la ciudad inexistente'));
    fila=-1;
    columna=-1;
else
    fila=str2double(fgetl(fileid));
    columna=str2double(fgetl(fileid));
end
```

## 20. lee\_ruta\_fichero.m

```

function [ruta,profundidad]= lee_ruta_fichero(nfichero)

%-----
%-----
% Función para leer una ruta de un fichero y almacenarla en una
% variable del programa
%-----
% SINTAXIS: [ruta,profundidad]=lee_ruta_fichero(nfichero)
%-----
% INPUT: nfichero: Cadena de caracteres con el nombre del fichero con
%          la ruta a leer
%-----
% OUTPUT: ruta: Sucesión de puntos [fila,columna] óptima entre dos
%          puntos a través de un mapa
%          profundidad: Número de puntos de la ruta
%-----
%
%
%                               Trabajo de Fin de Grado
%          "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (16/04/2014)
%                               Matlab R2013a
%-----
%-----

%Apertura del fichero que contiene la ruta
fileid=fopen(nfichero,'r');
if(fileid==-1)
    disp(sprintf('Error en la apertura del archivo'));
    return
end

%El primer valor del fichero de rutas es la profundidad de ésta
profundidad=str2double(fgetl(fileid));

%Bucle para cargar la información de la ruta
for i=1:profundidad
    ruta(i).fila=str2double(fgetl(fileid));
    ruta(i).columna=str2double(fgetl(fileid));
end

%Se cierra el fichero
fclose(fileid);

```

## 21. mostrar\_mapas.m

```
function varargout = mostrar_mapas(varargin)

%-----
%-----
% Interfaz de usuario para mostrar el mapa de riesgo y las listas de
% conflictos
%-----
% SINTAXIS: [vargout]=mostrar_mapas(varargin)
%-----
% INPUT: varargin{1}: Mapa RGB para mostrar
%         varargin{2}: Vector de camiones
%         varargin{3}: Número de alertas en la lista naranja
%         varargin{4}: Lista de conflictos naranja
%         varargin{5}: Número de alertas en la lista roja
%         varargin{6}: Lista de conflictos roja
%         varargin{7}: Número de alertas en la lista violeta
%         varargin{8}: Lista de conflictos violeta
%-----
% OUTPUT:
%-----
%
%                                     Trabajo de Fin de Grado
%      "Sistema de seguimiento y gestión de mercancías peligrosas"
%                                Diego Cuervo Fernández (20/06/2014)
%                                Matlab R2013a
%-----
%-----

% Código de inicialización - NO MODIFICAR
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @mostrar_mapas_OpeningFcn, ...
                  'gui_OutputFcn',  @mostrar_mapas_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```



```

% Fin del código de inicialización - NO MODIFICAR

% Se ejecuta previamente a mostrar la interfaz gráfica de usuario
function mostrar_mapas_OpeningFcn(hObject, eventdata, handles, ...
...varargin)

% Selecciona la línea de comandos como salida por defecto
handles.output = hObject;

% Actualiza la estructura de elementos de la GUI
guidata(hObject, handles);

% Muestra el mapa RGB
imshow(varargin{1});
hold on

%Lista de conflictos violetas
if(varargin{7}>=1)
    %Primer elemento de la lista
    if(varargin{8}(1,2)>0)
        %Conflicto de dos camiones
        cell_violeta={['Camiones nº ',num2str(varargin{8}(1,1)), ...
        ...' de ', selecciona_nombre(varargin{2}(varargin{8}...
        ...(1,1)).origen), ' a ', selecciona_nombre(varargin{2}...
        ...(varargin{8}(1,1)).destino), ' (g', num2str(varargin{2}...
        ...(varargin{8}(1,1)).grupo),') y ',num2str(varargin{8}...
        ...(1,2)), ' de ', selecciona_nombre(varargin{2}(varargin{8}...
        ...(1,2)).origen), ' a ', selecciona_nombre(varargin{2}...
        ...(varargin{8}(1,2)).destino), ' (g', num2str(varargin{2}...
        ...(varargin{8}(1,2)).grupo),') riesgo ', num2str(round(...
        ...varargin{8}(1,3)))]];
    else
        %Conflicto de un camión
        cell_violeta={['Camión nº ',num2str(varargin{8}(1,1)), ...
        ...' de ', selecciona_nombre(varargin{2}(varargin{8}...
        ...(1,1)).origen), ' a ', selecciona_nombre(varargin{2}...
        ...(varargin{8}(1,1)).destino), ' (g', num2str(varargin{2}...
        ...(varargin{8}(1,1)).grupo),') riesgo ', num2str(round(...
        ...varargin{8}(1,3)))]];
    end
    cell_filav = {num2str(varargin{2} (varargin{8} (1,1)).ruta ...
    ...(varargin{2} (varargin{8} (1,1)).indice).fila)};
    cell_colv = {num2str(varargin{2} (varargin{8} (1,1)).ruta...
    ...(varargin{2} (varargin{8} (1,1)).indice).columna)};

%Se completa la lista

```

```

for i=2:varargin{7}
    if(varargin{8}(i,2)>0)
        %Conflicto de dos camiones
        cell_violeta=[cell_violeta;{'Camiones nº ',[num2str(...
            ...varargin{8}(i,1)), ' de ',selecciona_nombre(varargin{2}...
            ...(varargin{8}(i,1)).origen), ' a ', selecciona_nombre(...
            ...varargin{2}(varargin{8}(i,1)).destino), ' (g', num2str(...
            ...varargin{2}(varargin{8}(i,1)).grupo), ') y ', num2str(...
            ...varargin{8}(i,2)), ' de ',selecciona_nombre(varargin{2}...
            ...(varargin{8}(i,2)).origen), ' a ', selecciona_nombre(...
            ...varargin{2}(varargin{8}(i,2)).destino), ' (g', num2str(...
            ...varargin{2}(varargin{8}(i,2)).grupo), ') riesgo ', ...
            ...num2str(round(varargin{8}(i,3)))}]]];
    else
        %Conflicto de un camión
        cell_violeta=[cell_violeta;{'Camión nº ',[num2str(...
            varargin{8}(i,1)), ' de ', selecciona_nombre(varargin{2}...
            ...(varargin{8}(i,1)).origen), ' a ', selecciona_nombre(...
            ...varargin{2}(varargin{8}(i,1)).destino), ' (g', num2str(...
            ...varargin{2}(varargin{8}(i,1)).grupo), ') riesgo ', ...
            ...num2str(round(varargin{8}(i,3)))}]]];
    end
    cell_fila_v = [cell_fila_v; {num2str(varargin{2}(varargin{8}...
        ... (i,1)).ruta(varargin{2}(varargin{8}(i,1)).indice).fila)}];
    cell_col_v = [cell_col_v; {num2str(varargin{2}(varargin{8}...
        ... (i,1)).ruta(varargin{2}(varargin{8}(i,1)).indice).columna)}];
end
%Si no hay ninguna alerta violeta, se avisa al usuario
else
    cell_violeta={'Ninguna alerta violeta'};
    cell_fila_v={num2str(-1)};
    cell_col_v={num2str(-1)};
end
%Se muestra lista violeta
set(handles.lista_violeta,'String',cell_violeta);
set(handles.lista_fila_v,'String',cell_fila_v);
set(handles.lista_col_v,'String',cell_col_v);

%Lista de conflictos rojos
if(varargin{5}>=1)
    %Primer elemento de la lista
    if(varargin{6}(1,2)>0)
        %Conflicto de dos camiones
        cell_roja={'Camiones nº ',[num2str(varargin{6}(1,1)), ...
            ...' de ', selecciona_nombre(varargin{2}(varargin{6}...
            ...(1,1)).origen), ' a ', selecciona_nombre(varargin{2}...

```

```

... (varargin{6}(1,1)).destino), ' (g', num2str(varargin{2}...
... (varargin{6}(1,1)).grupo), ') y ', num2str(varargin{6}...
... (1,2)), ' de ', selecciona_nombre(varargin{2}(varargin{6}...
... (1,2)).origen), ' a ', selecciona_nombre(varargin{2}...
... (varargin{6}(1,2)).destino), ' (g', num2str(varargin{2}...
... (varargin{6}(1,2)).grupo), ') riesgo ', num2str(round(...
... varargin{6}(1,3)))]]];
else
    %Conflicto de un camión
    cell_roja={['Camión nº ', num2str(varargin{6}(1,1)), ...
    ... ' de ', selecciona_nombre(varargin{2}(varargin{6}...
    ... (1,1)).origen), ' a ', selecciona_nombre(varargin{2}...
    ... (varargin{6}(1,1)).destino), ' (g', num2str(varargin{2}...
    ... (varargin{6}(1,1)).grupo), ') riesgo ', num2str(round(...
    ... varargin{6}(1,3)))]]];
end

cell_fil_r = {num2str(varargin{2} (varargin{6} (1,1)).ruta ...
... (varargin{2} (varargin{6} (1,1)).indice).fila)};
cell_col_r = {num2str(varargin{2} (varargin{6} (1,1)).ruta ...
... (varargin{2} (varargin{6} (1,1)).indice).columna)};

%Se completa la lista
for i=2:varargin{5}
    if(varargin{6}(i,2)>0)
        %Conflicto de dos camiones
        cell_roja=[cell_roja;{'Camiones nº ', num2str(...
        ... varargin{6}(i,1)), ' de ', selecciona_nombre(varargin{2}...
        ... (varargin{6}(i,1)).origen), ' a ', selecciona_nombre(...
        ... varargin{2}(varargin{6}(i,1)).destino), ' (g', num2str(...
        ... varargin{2}(varargin{6}(i,1)).grupo), ') y ', num2str(...
        ... varargin{6}(i,2)), ' de ', selecciona_nombre(varargin{2}...
        ... (varargin{6}(i,2)).origen), ' a ', selecciona_nombre(...
        ... varargin{2}(varargin{6}(i,2)).destino), ' (g', num2str(...
        ... varargin{2}(varargin{6}(i,2)).grupo), ') riesgo ', ...
        ... num2str(round(varargin{6}(i,3)))}}];
    else
        %Conflicto de un camión
        cell_roja=[cell_roja;{'Camión nº ', num2str(...
        ... varargin{6}(i,1)), ' de ', selecciona_nombre(varargin{2}...
        ... (varargin{6}(i,1)).origen), ' a ', selecciona_nombre(...
        ... varargin{2}(varargin{6}(i,1)).destino), ' (g', num2str(...
        ... varargin{2}(varargin{6}(i,1)).grupo), ') riesgo ', ...
        ... num2str(round(varargin{6}(i,3)))}}];
    end

    cell_fil_r = [cell_fil_r; {num2str(varargin{2}(varargin{6}...

```

```

        ...(i,1)).ruta(varargin{2}(varargin{6}(i,1)).indice).fila)]];
        cell_col_r = [cell_col_r; {num2str(varargin{2}(varargin{6}(...
        ...i,1)).ruta(varargin{2}(varargin{6}(i,1)).indice).columna)}}];
    end
    %Si no hay ninguna alerta roja, se avisa al usuario
    else
        cell_roja={'Ninguna alerta roja'};
        cell_fila_r={num2str(-1)};
        cell_col_r={num2str(-1)};
    end
    %Se muestra la lista roja
    set(handles.lista_roja,'String',cell_roja);
    set(handles.lista_fila_r,'String',cell_fila_r);
    set(handles.lista_col_r,'String',cell_col_r);

    %Lista de conflictos naranjas
    if(varargin{3}>=1)
        %Primer elemento de la lista
        if(varargin{4}(1,2)>0)
            %Conflicto de dos camiones
            cell_naranja=[['Camiones nº ',[num2str(varargin{4}(1,1)), ...
            ...' de ', selecciona_nombre(varargin{2}(varargin{4}...
            ...(1,1)).origen), ' a ', selecciona_nombre(varargin{2}...
            ...(varargin{4}(1,1)).destino), ' (g', num2str(varargin{2}...
            ...(varargin{4}(1,1)).grupo),') y ',num2str(varargin{4}...
            ...(1,2)), ' de ', selecciona_nombre(varargin{2}(varargin{4}...
            ...(1,2)).origen), ' a ', selecciona_nombre(varargin{2}...
            ...(varargin{4}(1,2)).destino), ' (g', num2str(varargin{2}...
            ...(varargin{4}(1,2)).grupo),') riesgo ', num2str(round(...
            ...varargin{4}(1,3)))]];
        else
            %Conflicto de un camión
            cell_naranja=[['Camión nº ',[num2str(varargin{4}(1,1)), ...
            ...' de ', selecciona_nombre(varargin{2}(varargin{4}...
            ...(1,1)).origen), ' a ', selecciona_nombre(varargin{2}...
            ...(varargin{4}(1,1)).destino), ' (g', num2str(varargin{2}...
            ...(varargin{4}(1,1)).grupo),') riesgo ', num2str(round(...
            ...varargin{4}(1,3)))]];
        end

        cell_fila_n = {num2str(varargin{2} (varargin{4} (1,1)).ruta ...
        ...(varargin{2} (varargin{4} (1,1)).indice).fila)};
        cell_col_n = {num2str(varargin{2} (varargin{4} (1,1)).ruta ...
        ...(varargin{2} (varargin{4} (1,1)).indice).columna)};

        %Se completa la lista

```

```

for i=2:varargin{3}
    if(varargin{4}(i,2)>0)
        %Conflicto de dos camiones
        cell_naranja=[cell_naranja;{'Camiones nº ',[num2str(...
            ...varargin{4}(i,1)), ' de ',selecciona_nombre(varargin{2}...
            ...(varargin{4}(i,1)).origen), ' a ', selecciona_nombre(...
            ...varargin{2}(varargin{4}(i,1)).destino), ' (g', num2str(...
            ...varargin{2}(varargin{4}(i,1)).grupo),') y ', num2str(...
            ...varargin{4}(i,2)), ' de ',selecciona_nombre(varargin{2}...
            ...(varargin{4}(i,2)).origen), ' a ', selecciona_nombre(...
            ...varargin{2}(varargin{4}(i,2)).destino), ' (g', num2str(...
            ...varargin{2}(varargin{4}(i,2)).grupo),') riesgo ', ...
            ...num2str(round(varargin{4}(i,3)))}]]];
    else
        %Conflicto de un camión
        cell_naranja=[cell_naranja;{'Camión nº ', [num2str(...
            ...varargin{4}(i,1)), ' de ',selecciona_nombre(varargin{2}...
            ...(varargin{4}(i,1)).origen), ' a ', selecciona_nombre(...
            ...varargin{2}(varargin{4}(i,1)).destino), ' (g', num2str(...
            ...varargin{2}(varargin{4}(i,1)).grupo),') riesgo ', ...
            ...num2str(round(varargin{4}(i,3)))}]]];
    end

    cell_fila_n = [cell_fila_n;{num2str(varargin{2}(varargin{4}...
        ...i,1)).ruta(varargin{2}(varargin{4}(i,1)).indice).fila)}];
    cell_col_n = [cell_col_n;{num2str(varargin{2}(varargin{4}(...
        ...i,1)).ruta(varargin{2}(varargin{4}(i,1)).indice).columna)}];
end

%Si no hay ninguna alerta naranja, se avisa al usuario
else
    cell_naranja={'Ninguna alerta naranja'};
    cell_fila_n={num2str(-1)};
    cell_col_n={num2str(-1)};
end

%Se muestra la lista naranja
set(handles.lista_naranja,'String',cell_naranja);
set(handles.lista_fila_n,'String',cell_fila_n);
set(handles.lista_col_n,'String',cell_col_n);

% Se inicializa la lista auxiliar de últimos visitados
cell_ultimos_visitados=[{num2str(-1)};{num2str(-1)};{num2str(-1)}];
set(handles.lista_ultimos_visitados,'String',cell_ultimos_visitados);

function varargout = mostrar_mapas_OutputFcn(hObject, eventdata, ...
    ...handles)
varargout{1} = handles.output;

```

```

% Se ejecuta cuando cambia la selección de la lista violeta.
function lista_violeta_Callback(hObject, eventdata, handles)
%Obtenemos índice del elemento seleccionado
indice_violeta = get(hObject, 'Value');

%Obtenemos último elemento visitado
u_visitados= get(handles.lista_ultimos_visitados, 'String');
ultimo_visitado=str2double(u_visitados{1});

%Obtenemos las coordenadas del elemento seleccionado
cell_fila_v = get(handles.lista_fila_v, 'String');
cell_col_v = get(handles.lista_col_v, 'String');
y=str2double(cell_fila_v{indice_violeta});
x=str2double(cell_col_v{indice_violeta});

%Si previamente se visitó algún elemento, eliminamos su cursor
if(ultimo_visitado~= -1)
    y_ultimo=str2double(cell_fila_v{ultimo_visitado});
    x_ultimo=str2double(cell_col_v{ultimo_visitado});
    plot(x_ultimo,y_ultimo, 'ko', 'markerfacecolor', [1 1 1]);
end

%Si las coordenadas son válidas, colocamos el cursor
if(x>0)
    plot(x,y, 'ko', 'markerfacecolor', [0.8 0 0.8]);
end

%Actualizamos la lista de últimos visitados
u_visitados{1}=num2str(indice_violeta);
set(handles.lista_violeta, 'Value', 1);
set(handles.lista_ultimos_visitados, 'String', u_visitados);

% Se ejecuta al crear la lista violeta (NO MODIFICAR).
function lista_violeta_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'), ...
...get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% Se ejecuta cuando cambia la selección de la lista naranja.
function lista_naranja_Callback(hObject, eventdata, handles)
%Obtenemos índice del elemento seleccionado
indice_naranja = get(hObject, 'Value');

%Obtenemos último elemento visitado
u_visitados= get(handles.lista_ultimos_visitados, 'String');

```

```

ultimo_visitado=str2double(u_visitados{3});

%Obtenemos las coordenadas del elemento seleccionado
cell_fila_n = get(handles.lista_fila_n,'String');
cell_col_n = get(handles.lista_col_n,'String');
y=str2double(cell_fila_n{indice_naranja});
x=str2double(cell_col_n{indice_naranja});

%Si previamente se visitó algún elemento, eliminamos su cursor
if(ultimo_visitado~= -1)
    y_ultimo=str2double(cell_fila_n{ultimo_visitado});
    x_ultimo=str2double(cell_col_n{ultimo_visitado});
    plot(x_ultimo,y_ultimo,'ko','markerfacecolor',[1 1 1]);
end

%Si las coordenadas son válidas, colocamos el cursor
if(x>0)
    plot(x,y,'ko','markerfacecolor',[1 0.5 0]);
end

%Actualizamos la lista de últimos visitados
u_visitados{3}=num2str(indice_naranja);
set(handles.lista_naranja,'Value',1);
set(handles.lista_ultimos_visitados,'String',u_visitados);

% Se ejecuta al crear la lista naranja (NO MODIFICAR).
function lista_naranja_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
...get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% Se ejecuta cuando cambia la selección de la lista roja.
function lista_roja_Callback(hObject, eventdata, handles)
%Obtenemos índice del elemento seleccionado
indice_rojo = get(hObject,'Value');

%Obtenemos último elemento visitado
u_visitados= get(handles.lista_ultimos_visitados,'String');
ultimo_visitado=str2double(u_visitados{2});

%Obtenemos las coordenadas del elemento seleccionado
cell_fila_r = get(handles.lista_fila_r,'String');
cell_col_r = get(handles.lista_col_r,'String');
y=str2double(cell_fila_r{indice_rojo});
x=str2double(cell_col_r{indice_rojo});

```

```

%Si previamente se visitó algún elemento, eliminamos su cursor
if(ultimo_visitado~-1)
    y_ultimo=str2double(cell_fila_r{ultimo_visitado});
    x_ultimo=str2double(cell_col_r{ultimo_visitado});
    plot(x_ultimo,y_ultimo,'ko','markerfacecolor',[1 1 1]);
end

%Si las coordenadas son válidas, colocamos el cursor
if(x>0)
    plot(x,y,'ko','markerfacecolor',[1 0 0]);
end
u_visitados{2}=num2str(indice_rojo);
set(handles.lista_roja,'Value',1);
set(handles.lista_ultimos_visitados,'String',u_visitados);

% Se ejecuta al crear la lista roja (NO MODIFICAR).
function lista_roja_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% Se ejecuta cuando cambia la selección de una lista auxiliar (NO SE
% USA).
function lista_fila_v_Callback(hObject, eventdata, handles)

% Se ejecuta al crear la lista de coordenadas fila violeta (NO
% MODIFICAR).
function lista_fila_v_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...)
...get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% Se ejecuta cuando cambia la selección de una lista auxiliar(NO SE
% USA).
function lista_col_v_Callback(hObject, eventdata, handles)

% Se ejecuta al crear la lista de coordenadas columna violeta (NO
% MODIFICAR).
function lista_col_v_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...)
...get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```



```
% Se ejecuta cuando cambia la selección de una lista auxiliar(NO SE
% USA).
```

```
function lista_fila_r_Callback(hObject, eventdata, handles)
```

```
% Se ejecuta al crear la lista de coordenadas fila roja (NO
% MODIFICAR).
```

```
function lista_fila_r_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
...get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% Se ejecuta cuando cambia la selección de una lista auxiliar(NO SE
% USA).
```

```
function lista_col_r_Callback(hObject, eventdata, handles)
```

```
% Se ejecuta al crear la lista de coordenadas columna roja (NO
% MODIFICAR).
```

```
function lista_col_r_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
...get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% Se ejecuta cuando cambia la selección de una lista auxiliar(NO SE
% USA).
```

```
function lista_col_n_Callback(hObject, eventdata, handles)
```

```
% Se ejecuta al crear la lista de coordenadas columna naranja (NO
% MODIFICAR).
```

```
function lista_col_n_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
...get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% Se ejecuta cuando cambia la selección de una lista auxiliar(NO SE
% USA).
```

```
function lista_fila_n_Callback(hObject, eventdata, handles)
```

```
% Se ejecuta al crear la lista de coordenadas fila naranja (NO
% MODIFICAR).
```

```
function lista_fila_n_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
...get(0,'defaultUiControlBackgroundColor'))
```

```
        set(hObject, 'BackgroundColor', 'white');
    end

% Se ejecuta cuando cambia la selección de una lista auxiliar(NO SE
% USA).
function lista_ultimos_visitados_Callback(hObject, eventdata, ...
...handles)

% Se ejecuta al crear la lista de últimos visitados (NO MODIFICAR).
function lista_ultimos_visitados_CreateFcn(hObject, eventdata, ...
...handles)
if ispc && isequal(get(hObject, 'BackgroundColor'), ...
...get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

## 22. muestra\_ciudades.m

```
function []=muestra_ciudades(fcol,ffil,fmap)

%-----
%-----
% Función para mostrar en un mapa una serie de puntos contenidos en
% dos ficheros, uno para filas y otro para columnas
%-----
% SINTAXIS: [] = muestra_ciudades(fcol,ffil,fmap)
%-----
% INPUT: fcol: Cadena de caracteres con el nombre del fichero que
%         contiene las coordenadas columna de las ciudades
%        ffil: Cadena de caracteres con el nombre del fichero que
%         contiene las coordenadas fila de las ciudades
%        fmap: Cadena de caracteres con el nombre del mapa en formato
%         TIFF
%-----
% OUTPUT:
%-----
%
%                                     Trabajo de Fin de Grado
%      "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (15/04/2014)
%                                   Matlab R2013a
%-----
%-----

%Se carga la matriz correspondiente al mapa y se binariza
mapa_RGB_Importado=imread(fmap);
mapa_binario=binariza(mapa_RGB_Importado);

%Corrección del mapa por el uso del 0 como índice de la matriz válido
mapa_binario(1:4981,1:6324)=mapa_binario(2:4982,2:6325);

%Obtenemos un mapa RGB en blanco y negro a partir del mapa binario
mapa_gray=bin2gray(mapa_binario);
mapaRGB=gray2rgb(mapa_gray);

%Apertura del fichero de filas
filasID=fopen(ffil,'r');
if(filasID==-1)
    disp(sprintf('Error en la apertura del fichero de filas'));
    return
end

%Apertura del fichero de columnas
```

```

columnasID=fopen(fcol,'r');
if(columnasID==-1)
    disp(sprintf('Error en la apertura del fichero de filas'));
    return
end

%Inicialización del contador y lectura del número de ciudades a
mostrar
i=1;
num_ciudades=str2double(fgetl(filasID));

%Inicialización del vector de filas
filas=zeros(num_ciudades);

%Se cargan las filas desde el archivo
while ~feof(filasID)
    filas(i)=str2double(fgetl(filasID));
    i=i+1;
end

%Inicialización del contador y lectura del número de ciudades a
mostrar
i=1;
num_ciudades=str2double(fgetl(columnasID));

%Inicialización del vector de columnas
columnas=zeros(num_ciudades);

%Se cargan las filas desde el archivo
while ~feof(columnasID)
    columnas(i)=str2double(fgetl(columnasID));
    i=i+1;
end

%Se cierran los archivos
fclose(filasID);
fclose(columnasID);

%Bucle para mostrar las ciudades
for i=1:num_ciudades
    for j=-40:40
        for k=-40:40
            if(abs(j)+abs(k)<=56)
                mapaRGB(filas(i)+j,columnas(i)+k,1)=255;
                mapaRGB(filas(i)+j,columnas(i)+k,2)=0;
                mapaRGB(filas(i)+j,columnas(i)+k,3)=0;
            end
        end
    end
end

```

```
end
end
end
end

%Se muestra el mapa con las ciudades
imshow(mapaRGB);
```

### 23. muestra\_ruta.m

```
function[] = muestra_ruta(ficheroTIF,ruta,tamano_ruta)

%-----
%-----
% Función para marcar en un mapa una serie de puntos pertenecientes a
% una ruta en color verde
%-----
% SINTAXIS: [] = muestra_ruta(ficheroTIF,ruta,tamano_ruta)
%-----
% INPUT: ficheroTIF: Cadena de caracteres con el nombre del mapa en
%          formato GEOTIFF
%          ruta: Sucesión de puntos [fila,columna] óptima entre dos
%          puntos a través de un mapa
%          tamano_ruta: Número de puntos de la ruta
%-----
% OUTPUT:
%-----
%
%
%                               Trabajo de Fin de Grado
%          "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (10/04/2014)
%                               Matlab R2013a
%-----
%-----

%Recibe fichero mapa raster y lo binariza
mapaRGB=imread(ficheroTIF);
mapa_bin=binariza(mapaRGB);

%Corrección del desfase por uso del '0' como índice válido
mapa_bin(1:4981,1:6324)=mapa_bin(2:4982,2:6325);

%Se obtiene mapa RGB a partir de mapa binario
mapa_gray=bin2gray(mapa_bin);
mapa_RGB=gray2rgb(mapa_gray);

%Bucle para marcar la ruta en el mapa
for i=1:tamano_ruta
    mapa_RGB(ruta(i).fila,ruta(i).columna,1)=0;
    mapa_RGB(ruta(i).fila,ruta(i).columna,2)=255;
    mapa_RGB(ruta(i).fila,ruta(i).columna,3)=0;
end

%Se muestra el mapa con la ruta marcada
imshow(mapa_RGB);
```

## 24. prueba\_meta\_map.m

```

function [iguales]=prueba_meta_map(x1,y1,x2,y2)

%-----
%-----
% Función que comprueba si dos puntos dados por sus coordenadas
% cartesianas son iguales o no
%-----
% SINTAXIS: [iguales]=prueba_meta_map(x1,y1,x2,y2)
%-----
% INPUT: x1: Fila del mapa correspondiente a la posición 1
%         y1: Columna del mapa correspondiente a la posición 1
%         x2: Fila del mapa correspondiente a la posición 2
%         y2: Columna del mapa correspondiente a la posición 2
%-----
% OUTPUT: iguales: Variable binaria que vale '1' si los puntos son
% iguales o '0' si son diferentes
%-----
%
%
%                                     Trabajo de Fin de Grado
%       "Sistema de seguimiento y gestión de mercancías peligrosas"
%                                     Diego Cuervo Fernández (08/04/2014)
%                                     Matlab R2013a
%-----
%-----

if(x1==x2 && y1==y2)
    iguales=1;
else
    iguales=0;
end

```

## 25. selección\_velocidad\_tramo.m

```
function [velocidad]=seleccion_velocidad_tramo(fil, col, mapa_tipo...
..., v_autovia, v_nacional, v_comarcal, v_local)
```

```
%-----
%-----
% Función que consulta en un mapa el tipo de carretera en el que se
% encuentra un vehículo y devuelve su velocidad media
%-----
% SINTAXIS: [velocidad] = seleccion_velocidad_tramo(fil, col, ...
%           ...mapa_tipo, v_autovia,v_nacional,v_comarcal,v_local)
%-----
% INPUT: fil: Fila del mapa correspondiente a la posición actual
%        col: Columna del mapa correspondiente a la posición actual
%        mapa_tipo: Matriz generada a partir de un mapa raster de la
%                  capa tipos de carreteras
%        v_autovia: Velocidad a la que el camión circula en una
%                  autovía
%        v_nacional: Velocidad a la que el camión circula en una
%                  carretera nacional
%        v_comarcal: Velocidad a la que el camión circula en una
%                  carretera comarcal
%        v_local: Velocidad a la que el camión circula en una
%                  carretera local
%-----
% OUTPUT: velocidad: Velocidad del camión en el tramo
%-----
%
%
%                               Trabajo de Fin de Grado
%       "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (23/04/2014)
%                               Matlab R2013a
%-----
%-----
```

```
%Se almacena el tipo de carretera por la que circula el camión
seleccion=mapa_tipo(fil,col);
```

```
%En función del tipo de carretera, se selecciona una velocidad u otra
```

```
switch seleccion
    case 1
        velocidad=v_autovia;
    case 2
        velocidad=v_autovia;
    case 3
        velocidad=v_nacional;
```



```
case 4
    velocidad=v_nacional;
case 5
    velocidad=v_local;
case 6
    velocidad=v_comarcal;
otherwise
    disp(sprintf('Error en la seleccion de velocidad. La ...
...variable seleccion vale %d',seleccion));
    velocidad=v_local;
end
```

## 26. selecciona\_color.m

```
function [RGB]=selecciona_color(riesgo)

%-----
%-----
% Función que selecciona un color para un determinado riesgo en
% función de unos umbrales definidos como constantes
%-----
% SINTAXIS: [RGB]=selecciona_color(riesgo)
%-----
% INPUT: riesgo: Riesgo actual de un determinado punto del mapa
%-----
% OUTPUT: RGB: Vector de 3 componentes correspondientes a los colores
%           rojo, verde y azul
%-----
%
%
%                               Trabajo de Fin de Grado
%       "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (24/04/2014)
%                               Matlab R2013a
%-----
%-----

umbral1=24; %Hasta umbral 1 color azul cyan
umbral2=100; %Hasta umbral 2 color verde
umbral3=150; %Hasta umbral 3 color amarillo
umbral4=200; %Hasta umbral 4 color naranja
umbral5=288; %Hasta umbral 5 color rojo
%A partir de umbral 5 color violeta

%Si el riesgo es menor que el umbral 1, color azul cyan
if (riesgo<=umbral1)
    RGB(1)=0;
    RGB(2)=255;
    RGB(3)=255;
else
    %Si el riesgo es menor que el umbral 2, color verde
    if(riesgo<=umbral2)
        RGB(1)=0;
        RGB(2)=255;
        RGB(3)=0;
    else
        %Si el riesgo es menor que el umbral 3, color amarillo
        if(riesgo<=umbral3)
            RGB(1)=255;
            RGB(2)=255;
```

```
    RGB(3)=0;
else
    %Si el riesgo es menor que el umbral 4, color naranja
    if(riesgo<=umbral4)
        RGB(1)=255;
        RGB(2)=127;
        RGB(3)=0;
    %Si el riesgo es mayor que el umbral 4, color rojo
    else
        if(riesgo<=umbral5)
            RGB(1)=255;
            RGB(2)=0;
            RGB(3)=0;
        else
            RGB(1)=204;
            RGB(2)=0;
            RGB(3)=204;
        end
    end
end
end
end
end
```

## 27. selecciona\_nombre.m

```
function [nombre]=selecciona_nombre(numero)

%-----
%-----
% Función que selecciona el nombre del municipio de la red de
% comunicación a partir de su número en la lista
%-----
% SINTAXIS: [nombre]=selecciona_nombre(numero)
%-----
% INPUT: numero: Número del municipio en la lista de nodos
%-----
% OUTPUT: nombre: Cadena de caracteres que contiene el nombre del
nodo
%-----
%
%                                     Trabajo de Fin de Grado
% "Sistema de seguimiento y gestión de mercancías peligrosas"
%                               Diego Cuervo Fernández (23/06/2014)
%                               Matlab R2013a
%-----
%-----

%En función del número se selecciona el nombre del municipio
switch numero
    case 1
        nombre='Agreda';
    case 2
        nombre='Aguilar';
    case 3
        nombre='Aranda';
    case 4
        nombre='Avila';
    case 5
        nombre='Bejar';
    case 6
        nombre='Benavente';
    case 7
        nombre='Burgos';
    case 8
        nombre='Caldas';
    case 9
        nombre='CRodrigo';
    case 10
        nombre='Leon';
    case 11
```

```
        nombre='Miranda';
case 12
    nombre='Palencia';
case 13
    nombre='Ponferrada';
case 14
    nombre='Puebla';
case 15
    nombre='Salamanca';
case 16
    nombre='SanRafael';
case 17
    nombre='Segovia';
case 18
    nombre='Soria';
case 19
    nombre='Valladolid';
case 20
    nombre='Zamora';
otherwise
    nombre='ERROR';
end
```