



ISIS-1221

# INTRODUCCIÓN A LA PROGRAMACIÓN

## Proyecto de Nivel 4

### Personas desmovilizadas en proceso de reintegración

#### Objetivo general

El objetivo de este proyecto es crear una aplicación para analizar las estadísticas de las personas desmovilizadas que han ingresado al proceso de reintegración. En el desarrollo de esta aplicación pondrá en práctica los conceptos del nivel 4.

#### Objetivos específicos

1. Implementar algoritmos para construir y recorrer matrices.
2. Utilizar las librerías pandas y matplotlib, así como consultar los sitios web oficiales donde se encuentra la documentación.
3. Descomponer un problema en subproblemas e implementar las funciones que los resuelven.

#### Instrucciones generales

La sección descripción de la aplicación le permitirá conocer el alcance, las funcionalidades esperadas y lo que debe realizar en este proyecto. Tenga en cuenta que, a lo largo de dicha sección encontrará el título "**ATENCIÓN**" con indicaciones para conseguir que el resultado de su proyecto corresponda con lo esperado. Es importante que las siga cuidadosamente. Antes de empezar, le sugerimos leer con atención todo el proyecto. Mientras lo lee, trate de reconocer los conceptos del curso que tendrá que poner en práctica. Recuerde que este proyecto debe realizarse de forma **completamente individual**.

#### Descripción de la aplicación

A lo largo de la historia de Colombia, el país se ha visto inmerso en un entorno de violencia rodeado por múltiples grupos armados. Como mecanismo para combatir de manera pacífica a esos colectivos, se ha planteado la desmovilización de sus integrantes para reintegrarlos a la vida civil. Para el año 2019, gracias a múltiples programas nacionales, se habían desmovilizado 60.193 personas. En miras de entender la condición socioeconómica y las actividades desarrolladas por este grupo de personas, el Estado ha llevado control de los desmovilizados durante todo su proceso de reintegración. Para ello, ha recolectado datos de los individuos desde 2001 hasta el día de hoy con información tal como su año de ingreso al proceso, el grupo al cual pertenecía, su departamento y municipio de residencia e información económica y familiar. Estos datos se han puesto a disposición de la ciudadanía de manera libre y abierta, protegiendo la información personal de los desmovilizados.

En este proyecto, usted trabajará con datos anonimizados de personas en el proceso de reintegración civil desde 2001 hasta 2021. Estos datos se encuentran registrados en el portal de datos del gobierno colombiano (<https://www.datos.gov.co/Inclusi-n-Social-y-Reconciliaci-n/ESTAD-STICAS-DE-LAS-PERSONAS-DESMOVLIZADAS-QUE-HA/39pj-dba6>). En el archivo "desmovilizados.csv" usted encuentra una versión simplificada y filtrada de los datos originales: suprimimos algunas entradas incompletas y columnas que no vamos a utilizar en este proyecto. Con este conjunto de datos usted podrá analizar información esencial para entender cómo se han comportado los mecanismos de inserción y su impacto tanto en los desmovilizados como en el país.

**¡Le deseamos la mejor de las suertes explorando estos datos!**

Su aplicación debe tener las siguientes partes:

## Parte 1: Leer la información del archivo

### Requerimiento 0: Cargar datos

Lo primero que debe hacer es permitir que se carguen los datos de un archivo csv a un DataFrame. Le debe preguntar al usuario el nombre del archivo, es decir que, **la función que implemente este requerimiento debe recibir como parámetro el nombre del archivo y debe retornar un DataFrame.**

Cada registro representa los datos de un desmovilizado. Las columnas del archivo y sus significados son los siguientes:

- TipoDeDesmovilización: Forma de desmovilización del individuo. Individual o Colectiva
- ExGrupo: Grupo del cual era miembro el individuo antes de su desmovilización: AUC: Autodefensas Unidas de Colombia, ELN: Ejército de Liberación Nacional, EPL: Ejército Popular de Liberación, ERG: Ejército Revolucionario Guevarista, ERP: Ejército Revolucionario del Pueblo, o FARC: Fuerzas Armadas Revolucionarias de Colombia.
- AñoDesmovilización: Año en el cual el individuo se desmovilizó.
- Sexo: Sexo del individuo.
- SituacionFinalFrenteAlProceso: Describe el estado del proceso en el que se encuentra la persona desmovilizada. Puede ser cualquiera entre 'Fuera del Proceso', 'Culminado', 'Ausente del proceso', 'En Proceso' o 'No ha ingresado'.
- DepartamentoDeResidencia: Departamento de residencia del individuo según registros de ubicación a la fecha de corte.
- MunicipioDeResidencia: Municipio de residencia del individuo según registros de ubicación a la fecha de corte.
- BeneficioTRV: Define si el individuo recibió beneficio en Actividades Transversales en el año anterior.
- BeneficioFA: Define si el individuo recibió beneficio en Formación Académica en el año anterior.
- BeneficioFPT: Define si el individuo recibió beneficio en Formación Para el Trabajo en el año anterior.
- BeneficioPDT: Define si el individuo recibió beneficio en Plan De Trabajo en el año anterior.
- OcupacionEconomica: Ocupación económica del individuo para la fecha de corte.
- DesembolsoBIE: Define si el individuo recibió desembolso de Beneficio de Inserción Económica a la fecha de corte.
- NumDeHijos: Número de hijos y/o hijastros registrados como integrantes del grupo familiar del individuo según el censo de familia. Los valores negativos corresponden a aquellas personas a quienes no se le ha aplicado el censo de familia.
- TotalIntegrantesGrupoFamiliar: Número total de integrantes del grupo familiar registrados según el censo de familia. Los valores negativos corresponden a aquellas personas a quienes no se le ha aplicado el censo de familia.

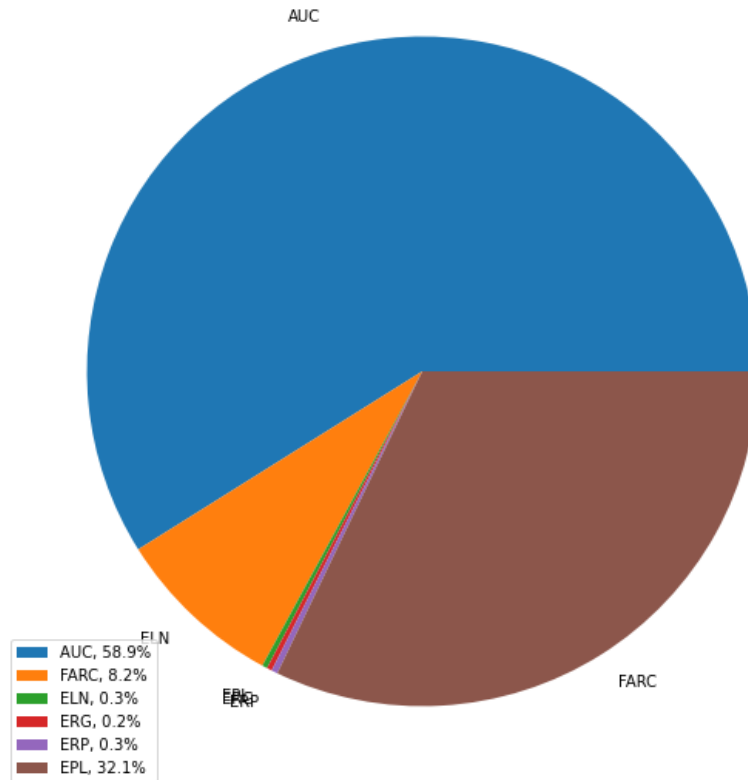
**ATENCIÓN:** cuando esté estudiando el problema y el archivo, recuerde que las funciones describe(), unique() y filter() pueden serle de utilidad. La función describe() aplicada sobre un DataFrame retorna información estadística de todas las columnas numéricas. La función unique() aplicada sobre una columna retorna una lista con los valores únicos que aparezcan en esa columna.

## Parte 2: Análisis de la distribución de los datos en los desmovilizados

### Requerimiento 1: Distribución de los desmovilizados según grupo armado

Para este requerimiento se quiere conocer la distribución de los desmovilizados de acuerdo con el grupo armado al que pertenecían. La siguiente figura muestra la apariencia de la gráfica esperada.

Diagrama de torta según ex grupo armado



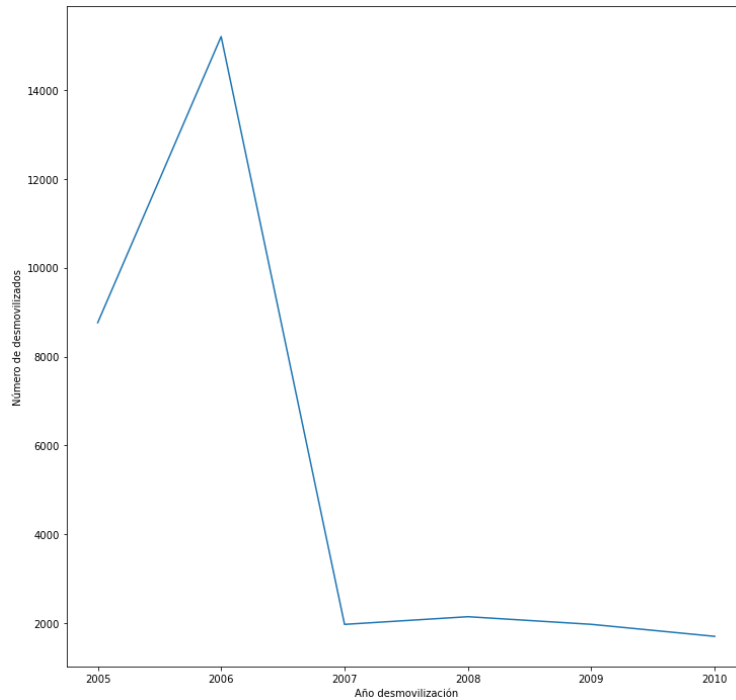
**ATENCIÓN:** Para poder añadir los porcentajes de cada grupo al lado de la leyenda puede calcular los porcentajes de los grupos y crear una lista de Strings con las etiquetas y los porcentajes con el formato '{l},{p}'. Donde *l* es el grupo y *p* el porcentaje de dicho grupo. Para añadir las leyendas al gráfico puede utilizar la siguiente línea de código.

```
plt.legend( loc='lower left', labels=labels)
```

Donde “labels” se la lista de las leyendas.

## Requerimiento 2: Tendencia del número de desmovilizados por un rango de años

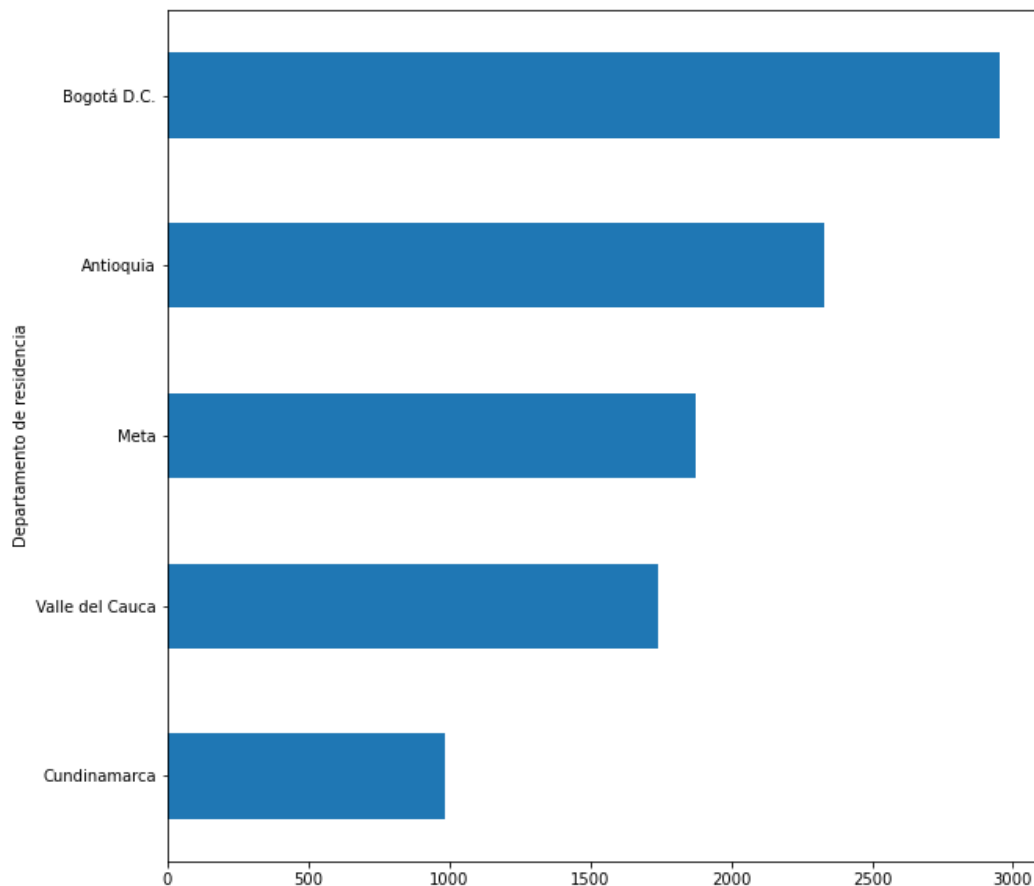
En este requerimiento se quiere conocer el número de desmovilizados en un rango de años. El usuario debe poder ingresar un límite inferior y un límite superior para determinar el intervalo de años que desea visualizar. Se debe mostrar un diagrama de línea con el número de personas desmovilizadas en el rango como se muestra a continuación:



**ATENCIÓN:** Tenga en cuenta que para el ejemplo se usó el rango de 2005 a 2010.

### Requerimiento 3: Top 5 departamentos por tipo de desmovilización

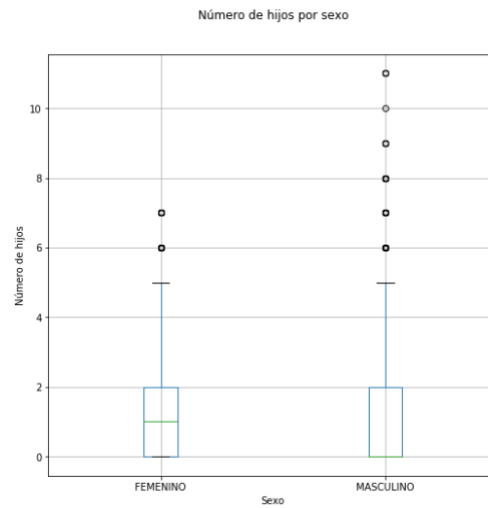
En este requerimiento se quieren conocer los 5 departamentos con mayor número de desmovilizados según el tipo de desmovilización ingresado por el usuario. Se debe mostrar un diagrama de barras horizontal como el que se muestra a continuación:



**ATENCIÓN:** Tenga en cuenta que para el ejemplo se ingresó la desmovilización de tipo Individual.

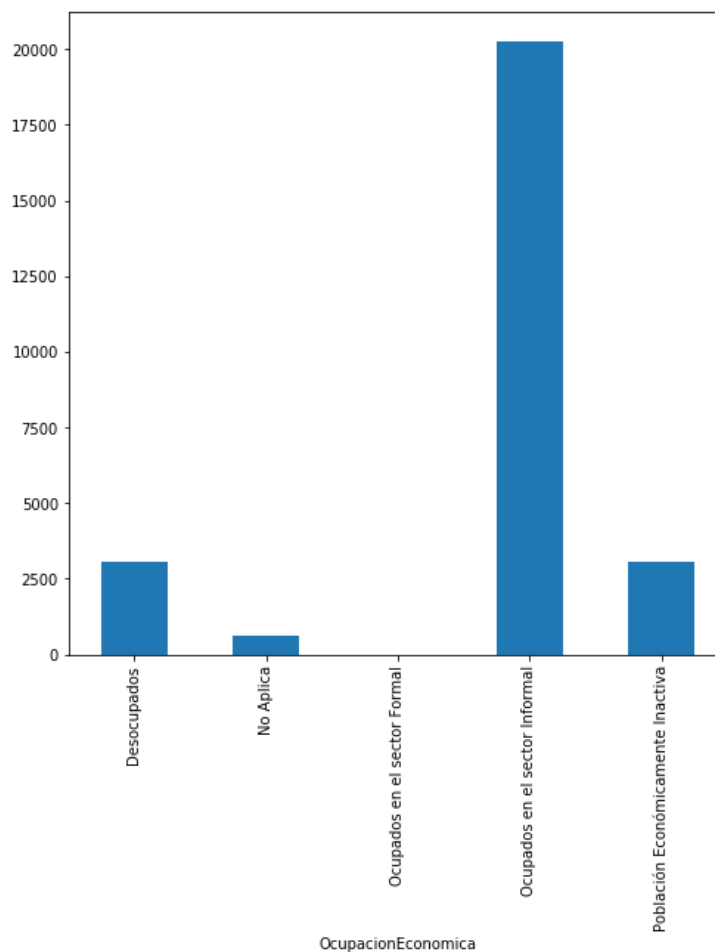
#### Requerimiento 4: Diagramas de caja y bigotes según número de hijos y sexo

En este requerimiento se quiere mostrar la distribución del número de hijos que tienen los desmovilizados según su sexo. El diagrama debe verse como el que se muestra a continuación:



#### Requerimiento 5: Ocupación de los individuos que hayan recibido algún beneficio o desembolso

Dentro de los datos se encuentran columnas que incluyen las palabras Beneficio y Desembolso, las cuales representan beneficios económicos y sociales a los cuales han tenido acceso algunos desmovilizados. El requerimiento debe mostrar un diagrama de barras con la distribución de las ocupaciones de los individuos que hayan tenido acceso a cualquier tipo de beneficio o desembolso. La gráfica se debe ver como la que se muestra a continuación:



**ATENCIÓN:** Para este requerimiento puede pensar en la negación para el filtrado. Seleccionar las filas en las que se cumpla al menos un beneficio, es igual a no tener en cuenta aquellas en donde todos los posibles beneficios son “No”. Para esto, puede hacer uso de los operadores & (and) y | (or) al momento de realizar el filtrado. Por ejemplo, si usted desea hacer un filtro con múltiples condiciones sobre las columnas de un Dataframe, puede hacer lo siguiente:

```
df_filtered = df[(df['columna1']== 'valor1') & (df['columna2']>=100)]
```

El anterior filtro selecciona las filas que tienen en la columna1 el valor1 y en la columna2 un valor mayor o igual a 100.

### Parte 3: Estudiar la cantidad de desmovilizados por departamento y ex grupo

En esta última parte no se trabajará directamente sobre el DataFrame original sino que se creará una matriz a partir de este. La matriz debe relacionar la cantidad de desmovilizados según su departamento y el grupo armado al que pertenecían.

#### Requerimiento 6: Construcción de la matriz de Departamento vs Ex Grupo

Este requerimiento consiste en construir una matriz que cruce el departamento en el que residen los desmovilizados y el grupo armado al que pertenecían. La matriz tiene la siguiente estructura:

785	78852	25151	....	5452
8765	645	8746	...	41231
...	...	...	...	...
754	87211	57571	...	54252

**NOTA:** La matriz mostrada anteriormente es un ejemplo y **NO** hace referencia a los valores que usted obtendrá al construir la matriz. Cada posición (f,c) en la matriz contiene el número de desmovilizados que pertenecen al departamento f y que pertenecían al grupo c. Por ejemplo, si Amazonas está en la fila 1 y ELN en la columna 3, el número de desmovilizados de Amazonas que pertenecían al ELN sería el valor en la casilla (1,3).

Así mismo, se propone la construcción de 2 diccionarios para la referencia de las filas y las columnas. La llave de los diccionarios hace referencia al índice de la fila o columna de la matriz y el valor hace referencia al departamento o ex grupo correspondiente. A continuación, se muestran los diccionarios para las filas y las columnas.

Llave	Valor
0	“Amazonas”
1	“Antioquia”
2	“Arauca”
...	...
4	“Vaupés”
5	“Vichada”

Llave	Valor
0	“AUC”
1	“ELN”
...	...
3	“ERP”
4	“FARC”

**NOTA:** El código para la creación de los dos diccionarios ya se encuentra en el esqueleto proporcionado.

### ATENCIÓN:

- Recuerde que tiene a su disposición la función `unique()` para obtener todos los valores únicos de una columna en un DataFrame.
- Para comprobar que los valores de la matriz creada son los correctos puede utilizar filtros, agrupaciones y sumas sobre el DataFrame original y verificar que estos arrojen el mismo resultado que el almacenado en su matriz.
- Puede ser de mucha utilidad pensar en descomponer el problema en varios problemas más pequeños.

**Nota importante:** Dado que el tamaño del DataFrame es bastante grande, la construcción de la matriz podría tardar más tiempo del que suelen tardar otras funciones. Si el tiempo supera los 2 minutos es muy probable que haya un problema en el código. En ese caso, le recomendamos que detenga el programa y revise si existe algún error en sus funciones.

La función para crear la matriz debe retornar una tupla que contenga la matriz creada y los diccionarios de filas y columnas en el siguiente formato (matriz, dict\_filas, dict\_columnas). Tenga en cuenta que usted deberá implementar los siguientes requerimientos utilizando como base la tupla anterior:

#### Requerimiento 7: Grupo con más desmovilizados por departamento dado

En este requerimiento se indicará el grupo con más desmovilizados en un departamento específico. Para esto, el usuario debe indicar el departamento para el cual desea obtener el grupo. La función que implemente esta opción debe retornar una cadena de texto con el nombre del grupo con más desmovilizados en ese departamento.

#### Requerimiento 8: Contar personas por grupo

En este requerimiento se debe calcular el número total de desmovilizados según su ex grupo armado. Para esto, el usuario debe indicar el nombre del ex grupo para el cual desea obtener el número total de desmovilizados. La función que implemente esta opción debe retornar un entero que represente el número de desmovilizados que pertenecían al grupo armado indicado.

#### Requerimiento 9: Departamento y grupo armado con mayor cantidad de desmovilizados

En este requerimiento se debe determinar el departamento y el grupo armado en el que hubo mayor número de desmovilizados. La función que implemente esta opción debe calcular la fila y la columna en donde se registró el mayor número de desmovilizados dentro de toda la matriz, y retornar una tupla de la forma  $(d, g)$ , donde  $d$  es el nombre del departamento y  $g$  es el nombre del ex grupo armado.

#### Requerimiento 10: Ex grupo con mayor desmovilización por departamento

Este último requerimiento de la aplicación consiste en determinar qué grupo armado fue el que más tuvo desmovilizados en cada departamento. Para ello, usted debe mostrar la información **sobre un mapa de Colombia**. Para esta visualización utilizaremos el archivo **mapa.png** (que se encuentra adjunto a este enunciado), el cual es una imagen del mapa de Colombia en blanco y negro con resolución de 800x800 píxeles. La idea es indicar dentro de cada departamento del mapa el ex grupo con mayor número de desmovilizados en ese departamento. Para cada uno de estos departamentos, se pintará un pequeño cuadrado de 13x13 píxeles centrado en las coordenadas (en píxeles) del Departamento dentro del mapa. Las coordenadas de los 33 departamentos se encuentran en el archivo **coordenadas.txt** (que se encuentra adjunto a este enunciado).

Para cargar el mapa como una matriz de píxeles y luego visualizarlo puede guiarse por el siguiente fragmento de código:

```
import matplotlib.image as mpimg
mapa = mpimg.imread("mapa.png").tolist()
plt.imshow(mapa)
plt.show()
```

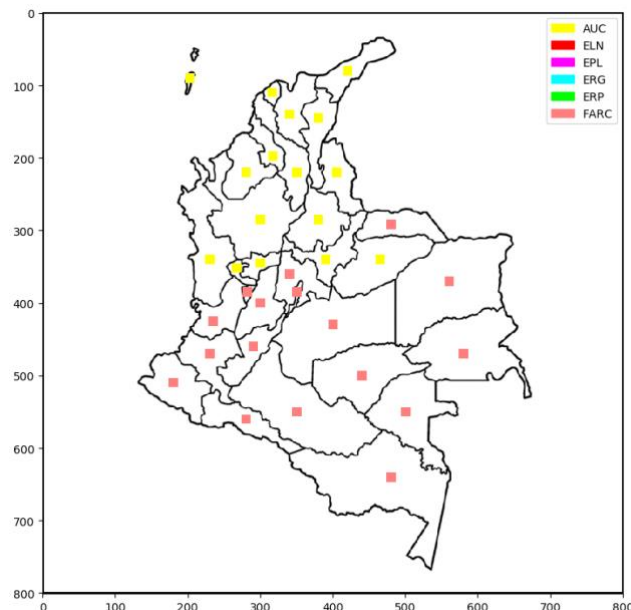
La siguiente función le permitirá cargar el archivo de coordenadas y retornar un diccionario, cuyas llaves son los nombres de los departamentos y los valores son tuplas con las coordenadas (x,y) de cada departamento. El nombre del departamento en este archivo es idéntico al que se encuentra en el DataFrame y en la matriz.

```
def cargar_coordenadas(nombre_archivo:str)->dict:
    deptos = {}
    archivo = open(nombre_archivo, encoding="utf8")
    titulos = archivo.readline()
    linea = archivo.readline()
    while len(linea) > 0:
        linea = linea.strip()
        datos = linea.split(";")
        deptos[datos[0]] = (int(datos[1]),int(datos[2]))
        linea = archivo.readline()
    return deptos
```

Una vez cargado el mapa y construido el diccionario de coordenadas, el siguiente paso es utilizar dichas coordenadas para pintar en el mapa varios puntos con un color que represente a cada grupo armado. Para ello, se han asignado los siguientes colores a cada grupo como se muestra a continuación:

Grupo armado	Componente RGB
AUC	1.0,1.0,0.0
ELN	1.0,0.0,0.0
EPL	1.0,0.0,1.0
ERG	0.0,1.0,1.0
ERP	0.0,1.0,0.0
FARC	1.0,0.5,0.5

El mapa resultante debe verse como el que se muestra a continuación:



### ATENCIÓN:

Para poder crear las leyendas de la gráfica anterior se debe utilizar el paquete mpatches de la librería de matplotlib. Para poder añadir las leyendas, haga uso del siguiente código:



- a) Importe la librería patches con el siguiente código:  

```
import matplotlib.patches as mpatches
```
- b) Cree un diccionario con los grupos y colores con la siguiente instrucción:  

```
colores = {"AUC": [1.0, 1.0, 0.0], "ELN": [1.0, 0.0, 0.0], "EPL": [1.0, 0.0, 1.0], "ERG": [0.0, 1.0, 1.0], "ERP": [0.0, 1.0, 0.0], "FARC": [1.0, 0.5, 0.5]}
```
- c) Arme una lista para las leyendas recorriendo el diccionario con el siguiente fragmento de código (teniendo en cuenta que `matriz` es la matriz de Departamento vs Ex grupo, y `dict_columns` el diccionario que contiene los nombres de las columnas de la matriz):  

```
legends = []
for i in range(1, len(matriz[0])):
    legends.append(mpatches.Patch(color = colores[dict_columns.keys()[i]],
    label=dict_columns.keys()[i]))
```
- d) Agregue las leyendas a la gráfica con la siguiente línea:  

```
plt.legend(handles = legends)
```

## Actividad 1: Preparación del ambiente de trabajo

1. Cree una carpeta para trabajar, poniéndole su nombre o login.
2. Descargue de Bloque Neón el archivo “desmovilizados.zip” que contiene el archivo “.csv” con los datos a procesar y un archivo .py que debe usar como esqueleto para realizar el proyecto. Este archivo contiene el código para crear los diccionarios de las filas y columnas de la matriz que usted debe construir.
3. Abra Spyder y cambie la carpeta de trabajo para que sea la carpeta donde descargó el archivo con los datos.

## Actividad 2: Construir el módulo de funciones

4. Usando Spyder, cree en su carpeta de trabajo un nuevo archivo con el nombre “desmovilizados.py”. En este archivo usted va a construir el módulo en el que va a implementar las funciones que responden a los requerimientos de la aplicación. **Defina, documente e implemente** las funciones en su nuevo archivo. Usted puede crear cuántas funciones considere necesarias dentro de su librería o módulo. Mínimo debe haber una función por cada uno de los requerimientos del programa.

## Actividad 3: Construir la interfaz de usuario basada en consola

5. Implemente la interfaz de la aplicación en un nuevo archivo llamado “consola\_desmovilizados.py”. Esta interfaz debe seguir la misma estructura de las consolas que hemos implementado en laboratorios y proyectos anteriores. Es decir, debe existir una función llamada `iniciar_aplicacion()` para que muestre el menú usando la función `mostrar_menu()` y permita al usuario seleccionar una opción. El menú que se despliegue debe permitir al usuario ejecutar todas las acciones de la descripción de la aplicación, así como salir (terminar) del programa. Por cada una de las funciones principales de su programa debe existir una función en la consola que la ejecute, pidiendo previamente los datos necesarios al usuario (si aplica) e imprimiendo por pantalla el resultado de la función. Se sugiere nombrar estas funciones como `ejecutar_XX`, donde XX es la respectiva función de su módulo. Cuando haya implementado la función `iniciar_aplicacion()` corra su programa y verifique que: se comporta adecuadamente, le permite al usuario seleccionar las opciones que quiere ejecutar, y termina el programa cuando se le indica.

## Actividad 4: Probar el correcto funcionamiento de su programa

6. **Ejecute el programa** y **pruebe** cada una de las funciones para asegurarse que esté funcionando. Puede probar el correcto funcionamiento de su programa cargando la información que se encuentra en el archivo “desmovilizados.csv” o creando su propio archivo de prueba de menor tamaño (respetando el mismo formato) que le permita corroborar que los resultados arrojados por su programa son correctos.

## Entrega

7. Comprima los dos archivos: `desmovilizados.py` y `consola_desmovilizados.py` en un solo archivo .zip. El archivo comprimido debe llamarse “N4-PROY-login.zip”, donde login es su nombre de usuario de Uniandes (omite el punto del login para evitar posibles problemas con la extensión de los archivos. Por ejemplo, si su login fuese “p.perez123”, nombre el archivo como “N4-PROY-pperez123.zip”).

8. Entregue el archivo comprimido a través de Bloque Neón en la actividad designada como **“Proyecto del Nivel 4”**.