



ISIS-1221

# INTRODUCCIÓN A LA PROGRAMACIÓN

## Nivel 4 - Proyecto Rendimiento de cultivos en Colombia

### Objetivo general

El objetivo de este proyecto es crear una aplicación para analizar el rendimiento de los cultivos en Colombia en el año 2017. 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 repetidas veces 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

La agricultura en Colombia es una pieza fundamental de la economía nacional. Con una diversidad climática y geográfica excepcional, el país cultiva una amplia variedad de productos agrícolas que van desde café y flores hasta cereales y frutas tropicales. Esta actividad no solo garantiza el abastecimiento de alimentos para la población, sino que también es un importante motor de desarrollo económico en las áreas rurales, generando empleo y contribuyendo al crecimiento del país. Los agricultores colombianos, con su dedicación y experiencia, han logrado posicionar a Colombia como uno de los principales productores y exportadores de café, flores, banano, aguacate, entre otros productos agrícolas. La agricultura se ha convertido en un sector estratégico para la economía nacional, contribuyendo significativamente a la generación de divisas y fortaleciendo la presencia de Colombia en el mercado internacional.

En este proyecto usted trabajará con datos del rendimiento de cultivos en Colombia en el año 2017. Estos datos se encuentran registrados en el portal de datos del gobierno colombiano

(<https://www.datos.gov.co/Agricultura-y-Desarrollo-Rural/RENDIMIENTO-DE-CULTIVOS-EN-COLOMBIA-POR-A-O/fgh5-rjkd>). En el archivo “cultivos.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 de los cultivos en Colombia.

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.**

Las columnas del archivo y sus significados son los siguientes:

- **Departamento:** Nombre del departamento productor.
- **Municipio:** Nombre del municipio productor.
- **Tipo\_Cultivo:** Tipo de cultivo que se produce en el territorio.
- **Cultivo:** Cultivo específico que se produce en el territorio.
- **Hectareas\_semradas:** Cantidad de hectáreas sembradas en un cultivo en específico.
- **Hectareas\_cosechadas:** Cantidad de hectáreas cosechadas en un cultivo en específico.
- **Toneladas:** Toneladas obtenidas al momento de cosechar un cultivo.

**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

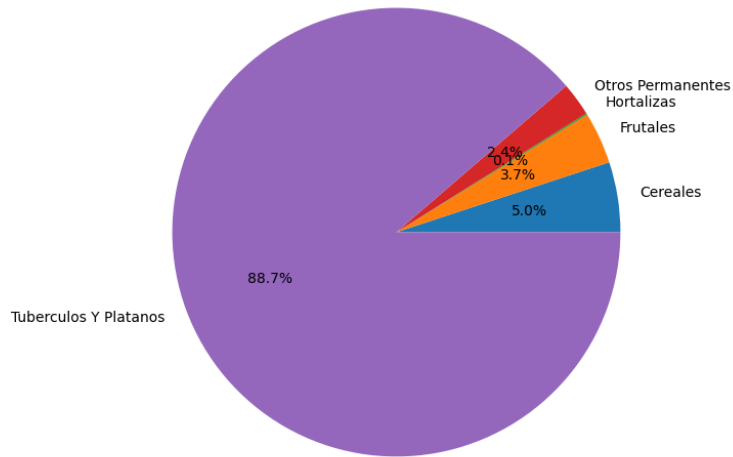
### Requerimiento 1: Distribución de tipos de cultivo en un departamento

Para este requerimiento se quiere conocer la distribución de la producción por tipo de cultivo en un departamento. Para esto, el usuario debe indicar el departamento al cual se le desea consultar la información y el programa deberá generar un diagrama de tipo pastel con la información. La siguiente figura muestra la apariencia de la gráfica esperada.

**Nota:** La gráfica muestra la información del departamento del Vaupés (Va sin tilde en el dataset).

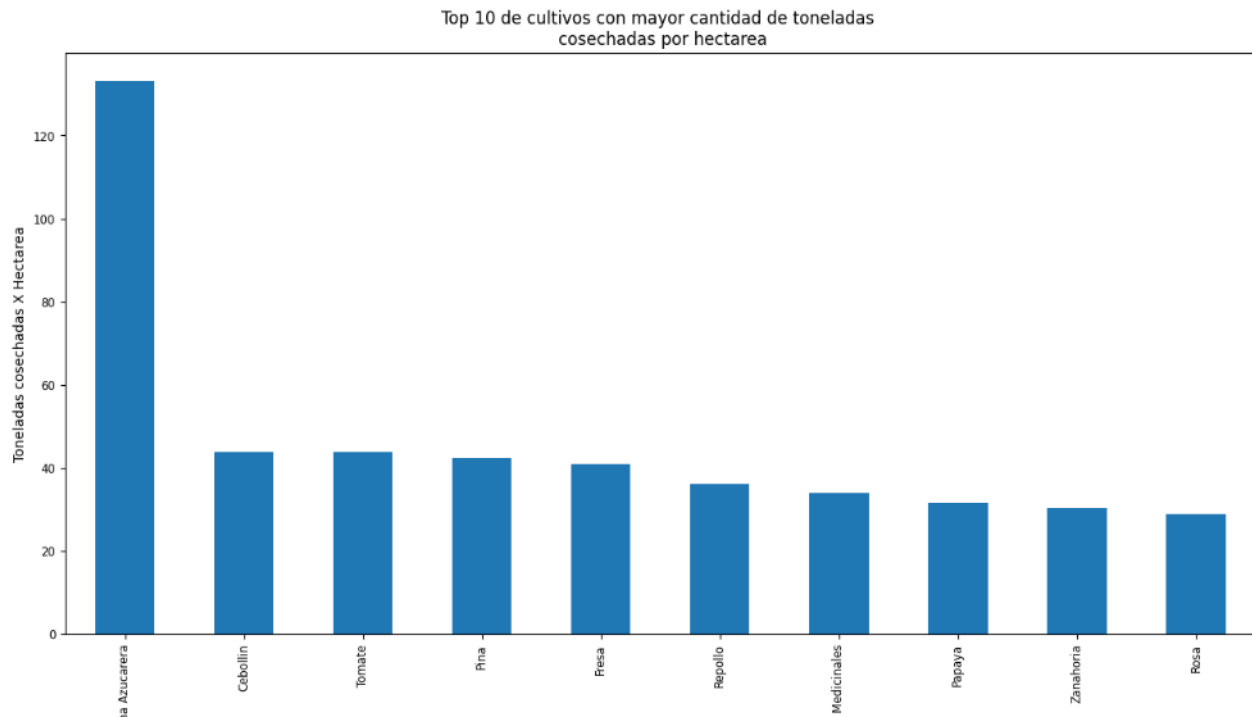
**Nota 2:** Para agregar los porcentajes en la gráfica, utilice el parámetro `autopct='%1f%%'` al momento de crear el diagrama de pastel.

Distribución de toneladas por tipo de cultivo en Vaupes



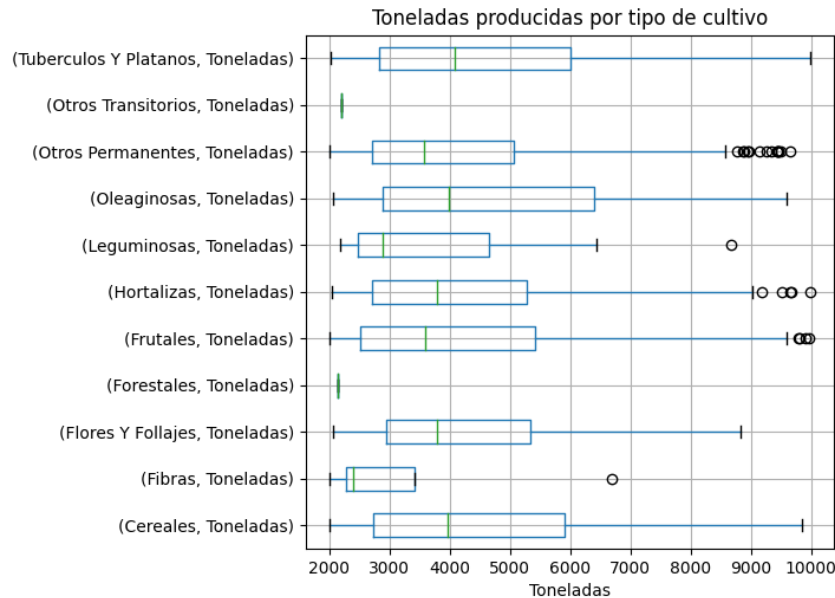
## Requerimiento 2: Top 10 cultivos toneladas cosechadas x hectárea

En este requerimiento se quiere conocer el top 10 de cultivos con mayor cantidad de toneladas cosechadas por hectárea. Para calcular la cantidad de toneladas producidas por hectárea es necesario obtener el total de *toneladas cosechadas*, el total de *hectáreas cosechadas* y después dividir entre sí estos valores para cada cultivo (*Toneladas/Hectáreas*). Se debe mostrar en un diagrama de barras vertical como el que se muestra a continuación:



**Requerimiento 3: Caja y bigotes de las toneladas producidas en los tipos de cultivos**

En este requerimiento se quiere mostrar la distribución de las toneladas producidas por cada tipo de cultivo en un rango proporcionado por el usuario. El diagrama debe verse como el que se muestra a continuación:



**Nota:** La gráfica anterior está elaborada en el rango entre 2000 y 10000 toneladas.

**Parte 3 | Estudiar la producción en toneladas por departamento**

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 **producción en toneladas por tipo de cultivo** y los **departamentos del territorio nacional**.

**Requerimiento 4: Construcción de la matriz de Departamento vs Tipo Cultivo**

Este requerimiento consiste en construir una matriz que cruce el departamento vs los tipos de cultivo que se producen en el departamento. La matriz tiene la siguiente estructura:

	Cereales	Fibras	Flores y Follajes	...	Tuberculos y Platanos
Amazonas	72	732	200	....	556
Antioquia	256	1000	34	...	411
Bolivar	14	201256	0	...	14
...	...	...	...	...	...
Vichada	71	201	554	...	532

**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 toneladas producidas en el departamento f y que son del tipo c (cereales, fibras, flores y follajes,

etc). Por ejemplo, si Amazonas está en la fila 1 y hortalizas en la columna 3, el número de toneladas producidas que son de la categoría de hortalizas en el departamento de Amazonas sería el valor en la casilla (1,3). En caso que el departamento no produzca ese tipo de cultivo, en la matriz deberá aparecer un 0.

Por otra parte, 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 tipo de cultivo correspondiente. **En el esqueleto, encontrará el fragmento de código que le permitirá construir estos diccionarios.** A continuación, se muestran el formato de los diccionarios para las filas y las columnas.

Llave	Valor
0	“Amazonas”
1	“Antioquia”
2	“Arauca”
...	...
32	“Vichada”

Llave	Valor
0	“Cereales”
1	“Fibras”
...	...
12	“Tuberculos Y Platanos”

#### 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\_columnas, dict\_filas**). Tenga en cuenta que usted deberá implementar los siguientes requerimientos utilizando como base la tupla anterior.

**Requerimiento 5: Contar la cantidad total de toneladas producidas en un departamento dado**

En este requerimiento se debe contar la cantidad total de toneladas producidas en un departamento. Para esto, el usuario debe indicar el nombre del departamento al cual se le desea consultar la información. La función que implemente esta opción debe retornar el valor que representa la cantidad de toneladas producidas en el departamento dado por el usuario.

**Requerimiento 6: Encontrar el departamento mayor o menor productor de un tipo de cultivo dado por el usuario**

En este requerimiento se debe calcular cuál departamento ha producido más o menos toneladas de un tipo de cultivo. Para esto, el usuario debe indicar si desea conocer el tipo de cultivo de mayor producción o el de menor producción, además del tipo de cultivo a buscar. Para el parámetro que indica si se desea saber el mayor o menor productor, este será de tipo booleano (True si es el mayor, False de lo contrario). La función que implemente esta opción debe retornar en una tupla el nombre del departamento y el número de toneladas producidas.

**NOTA:** Para este requerimiento, en caso de que se desee encontrar el menor productor de un tipo de cultivo, se debe encontrar el menor productor diferente de 0, es decir que debe considerarse un departamento que produzca algo de ese tipo de cultivo.

**Requerimiento 7: Encontrar departamento productor estrella dado**

En este requerimiento se debe encontrar si algún departamento tiene al menos 3 tipos de cultivo que superen un número específico de toneladas. Para esto, el usuario debe indicar el número mínimo de toneladas. La función que implemente esta opción debe retornar un diccionario con las llaves “depto” y “tipos” cuyos valores son el nombre del departamento y una lista con los nombres de los 3 tipos de cultivo que superan el número de toneladas dado por el usuario. Si varios departamentos son productores estrella, retorne la información del primero encontrado. Si el departamento estrella tiene más de 3 tipos que superen el tope, retorne los nombres de los primeros 3 encontrados. En caso de que **ningún** departamento sea un productor estrella, los valores deberán ser “Ninguno” y una lista vacía.

**Requerimiento 8: Mostrar producción por departamento**

Este último requerimiento de la aplicación consiste en mostrar la producción de un tipo de cultivo acotado en un rango de toneladas para cada departamento. En primer lugar, el usuario debe suministrar la matriz de departamentos vs. tipo de cultivo y el nombre del tipo de cultivo de su interés.

Una vez obtenida la producción para cada departamento de ese tipo de cultivo, se debe mostrar la información sobre un mapa de Colombia. Para esta visualización utilizaremos el archivo **mapa.png** (que se encuentra incluido en el esqueleto), el cual es una imagen del mapa de Colombia en blanco y negro con resolución de 800x800 píxeles. La idea es indicar dentro del mapa la producción del tipo de cultivo en cada departamento. Para cada uno de estos Departamentos, se pintará un pequeño cuadrado de 13x13 píxeles centrado en las coordenadas (en píxeles) del Departamento dentro el mapa. Las coordenadas de los 33 departamentos se encuentran en el archivo **coordenadas.txt** (que también se encuentra incluido en el esqueleto).

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()
```

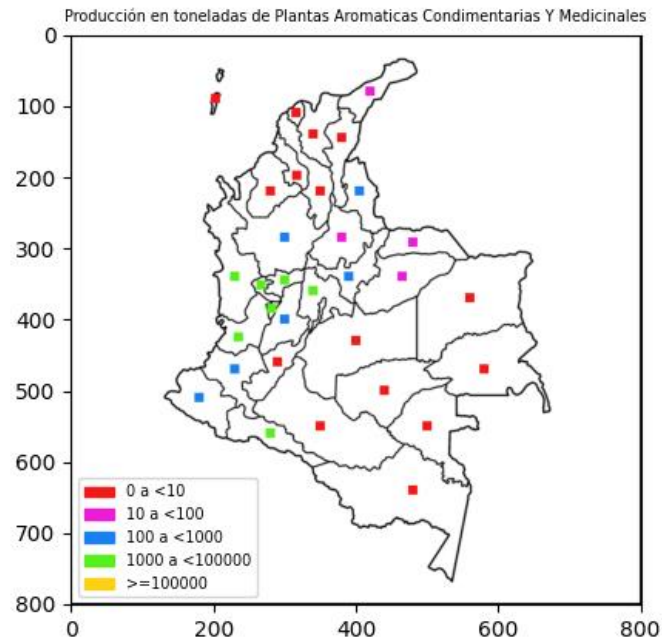
La siguiente función le permitirá cargar el archivo de coordenadas y retornar un diccionario, cuyas llaves son los nombres de los departamentos y los valores son tuplas con las coordenadas (x,y) de cada departamento. El nombre del departamento en este archivo es idéntico al que se encuentra en el DataFrame 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 la cantidad de toneladas producidas del tipo de cultivo de interés. Para ello, se han asignado los siguientes colores a cada grupo como se muestra a continuación:

Rango	Valores RGB	Color
0 a <10	[0.94, 0.10, 0.10]	Rojo
10 a <100	[0.94, 0.10, 0.85]	Magenta
100 a <1000	[0.10, 0.50, 0.94]	Azul
1000 a <100000	[0.34, 0.94, 0.10]	Verde
>=100000	[0.99, 0.82, 0.09]	Amarillo

**En resumen**, esto quiere decir que dependiendo del rango en que se encuentre la producción de cada departamento, es necesario pintar un cuadrado de 13x13 alrededor de la coordenada de dicho departamento. **Por ejemplo**, si para el departamento de Boyacá, para el tipo de cultivo “Plantas Aromaticas Condimentarias y Medicinales” la producción estuvo en el rango entre 100 y <1000 toneladas, se deben remplazar los píxeles en el cuadrado alrededor de las coordenadas de Boyacá (340,390) por píxeles del color Azul que aparece en la tabla. Recuerde que cada uno de los cuadrados en el mapa debe utilizar uno estos colores, y se debe respetar el orden de la clasificación. La siguiente gráfica muestra el mapa para el tipo de cultivo “Plantas Aromaticas Condimentarias y Medicinales”



### ATENCIÓN:

Para poder crear las leyendas de la gráfica anterior, es decir, el cuadro en la esquina inferior izquierda del mapa, se debe utilizar el paquete `mpatches` de la librería de `matplotlib`. Para poder añadir las leyendas, haga uso del siguiente código:

```
colores = {"0 a <10": [0.94, 0.10, 0.10], "10 a <100": [0.94, 0.10, 0.85], "100 a <1000": [0.10, 0.50, 0.94], "1000 a <100000": [0.34, 0.94, 0.10], ">=100000": [0.99, 0.82, 0.09]}
```

```
plt.imshow(mapa)
```

```
legends = []
for i in colores:
    legends.append(mpatches.Patch(color = colores[i], label=i))
plt.legend(handles = legends, loc = 3, fontsize='x-small')
plt.title("Producción en toneladas de " + tipo_cultivo, fontsize='x-small')
plt.show()
```

### 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 **cultivos.zip** que contiene el archivo **cultivos.csv** con los datos a procesar, los archivos **cultivos.py** y **consola\_cultivos.py** que debe usar como esqueleto para realizar el proyecto, y los archivos **mapa.png** y **coordenadas.txt**. El archivo **cultivos.py** 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, complete el archivo **cultivos.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 consolas**

5. En esta actividad usted tiene que construir la interfaz basada en consola para que el usuario interactúe con la aplicación. Para construir esta interfaz usted debe completar el archivo **consola\_cultivos.py**, la cual ya tiene una parte implementada que le facilitará su trabajo. Usted debe modificar los elementos marcados con la etiqueta **TODO**.
6. Pruebe la interfaz basada en consola ejecutando el archivo **consola\_cultivos.py**. Verifique que las funcionalidades de su aplicación se comporten de acuerdo con lo esperado.

### **Actividad 4 | Probar el funcionamiento correcto de su programa**

7. **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 **cultivos.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**

8. Comprima los dos archivos: **cultivos.py** y **consola\_cultivos.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**).
9. Entregue el archivo comprimido a través de Bloque Neón en la actividad designada como **Proyecto de Nivel 4**.