



**ISIS-1221**

# **INTRODUCCIÓN A LA PROGRAMACIÓN**

## **Nivel 4 – Laboratorio 3 Matrices y análisis de datos**

### **Objetivos**

1. Manipular matrices representando información estructurada.
2. Visualizar y analizar datasets usando Pandas y Matplotlib.

### **Preparación del ambiente de trabajo**

Para este laboratorio debe crear dos módulos: `matrices.py` para las actividades de matrices y `analisis_musica.py` para las actividades de visualización.

Se le proporcionará un archivo `cupicharts.csv` con datos de canciones en las listas de popularidad.

## Parte 1: Matrices

### Actividad 1: Búsqueda en cuadrantes

Escriba una función que reciba una matriz cuadrada de enteros positivos y el nombre de un color. El color determina en qué cuadrante de la matriz se debe buscar. La función debe retornar el primer número en el cuadrante especificado que sea par, cuando se recorra por filas y luego por columnas. Si no existe tal número, debe retornar `None`.

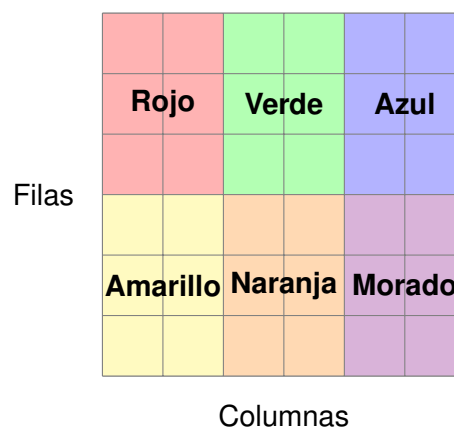
La matriz se divide en seis cuadrantes nombrados por colores:

#### Mitad superior de la matriz:

- **Rojo:** tercio izquierdo de columnas
- **Verde:** tercio central de columnas
- **Azul:** tercio derecho de columnas

#### Mitad inferior de la matriz:

- **Amarillo:** tercio izquierdo de columnas
- **Naranja:** tercio central de columnas
- **Morado:** tercio derecho de columnas



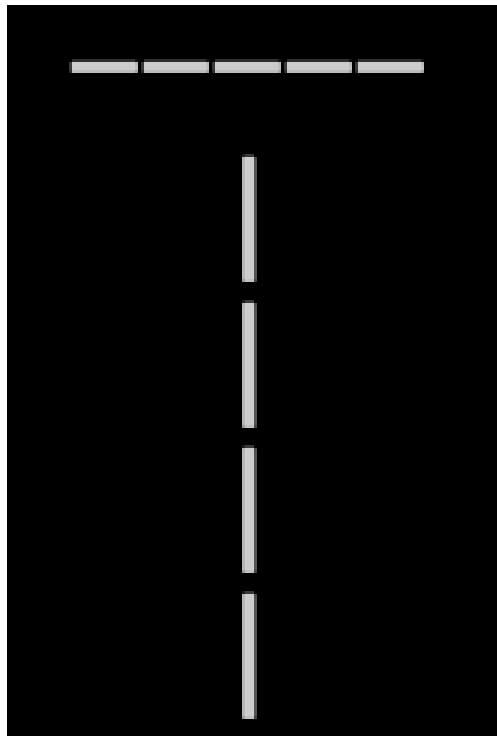
**E.g.:** En una matriz 6x6, el cuadrante verde abarca las filas 0-2 y columnas 2-3.

## Actividad 2: Modificación de patrones

Escriba una función que reciba una matriz cuadrada y la modifique pintando la letra "T".

- La primera fila de la matriz debe contener únicamente el carácter -
- La columna del medio debe contener el carácter | desde la segunda fila hasta la última fila
- Todas las demás posiciones de la matriz deben contener el carácter espacio ' '

Ejemplo para una matriz 5x5:



## Actividad 3: Análisis de datos estructurados

Escriba una función que reciba una tupla con información de ventas y el nombre de una ciudad, y retorne una tupla con el producto que tiene mayores ventas en esa ciudad y el valor de esas ventas.

La tupla de información contiene tres elementos:

- Una matriz donde cada fila representa un producto y cada columna representa una ciudad
- Una lista con los nombres de los productos (en el mismo orden que las filas)
- Una lista con los nombres de las ciudades (en el mismo orden que las columnas)

Si la ciudad especificada no existe en los datos, la función debe retornar la tupla `(None, None)`.

**E.g.:** Si la matriz es `[[1500, 2000], [800, 1500]]`, los productos son `["Laptop", "Mouse"]`, las ciudades son `["Bogotá", "Medellín"]`, y se consulta por `"Medellín"`, la función debe retornar `("Laptop", 2000)` porque 2000 es mayor que 1500.

## Actividad 4: Simulación de salón de clase

Escriba una función que reciba una matriz cuadrada representando un salón de clase y retorne las coordenadas del primer estudiante introvertido que empezará a hablar en clase. Si ningún introvertido empezará a hablar, debe retornar `(None, None)`.

En la matriz, cada posición puede contener:

- 0: Puesto vacío
- 1: Estudiante introvertido (callado)
- 2: Estudiante extrovertido (hablando)

Un estudiante introvertido empezará a hablar si cumple estas dos condiciones simultáneamente:

1. Tiene estudiantes extrovertidos en las dos posiciones diagonales inmediatamente detrás de él (diagonal inferior izquierda y diagonal inferior derecha). Si alguna de estas diagonales no existe porque el estudiante está en el borde del salón, esta condición no se cumple.
2. La posición inmediatamente adelante no contiene un estudiante extrovertido. Si la posición adelante no existe porque el estudiante está en la primera fila, esta condición sí se cumple.

El recorrido debe realizarse de adelante hacia atrás (de arriba hacia abajo) y de izquierda a derecha, como si estuviera mirando el salón desde el tablero.

E.g.: Si hay un introvertido en posición `(1,1)`, tiene extrovertidos en `(2,0)` y `(2,2)`, y la posición `(0,1)` no contiene un extrovertido, entonces ese introvertido empezará a hablar y sus coordenadas deben ser retornadas.

## Parte 2: Visualización de datos con Pandas

### Actividad 5: Análisis de popularidad musical

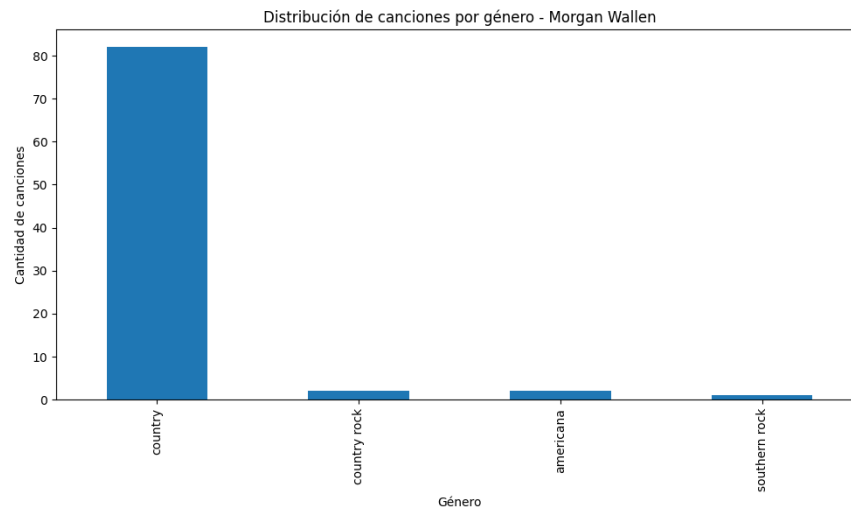
Escriba un programa que cargue el archivo `cupicharts.csv` usando Pandas y genere dos gráficas de análisis sobre las canciones en las listas de popularidad.

#### Gráfica 1: Distribución de canciones por género para un artista

Implemente una función que reciba el DataFrame y el nombre de un artista, filtre las canciones de ese artista, y genere una gráfica de barras verticales mostrando cuántas canciones tiene en cada género musical.

La gráfica debe incluir:

- Título indicando el nombre del artista
- Etiqueta del eje X: "Género"
- Etiqueta del eje Y: "Cantidad de canciones"



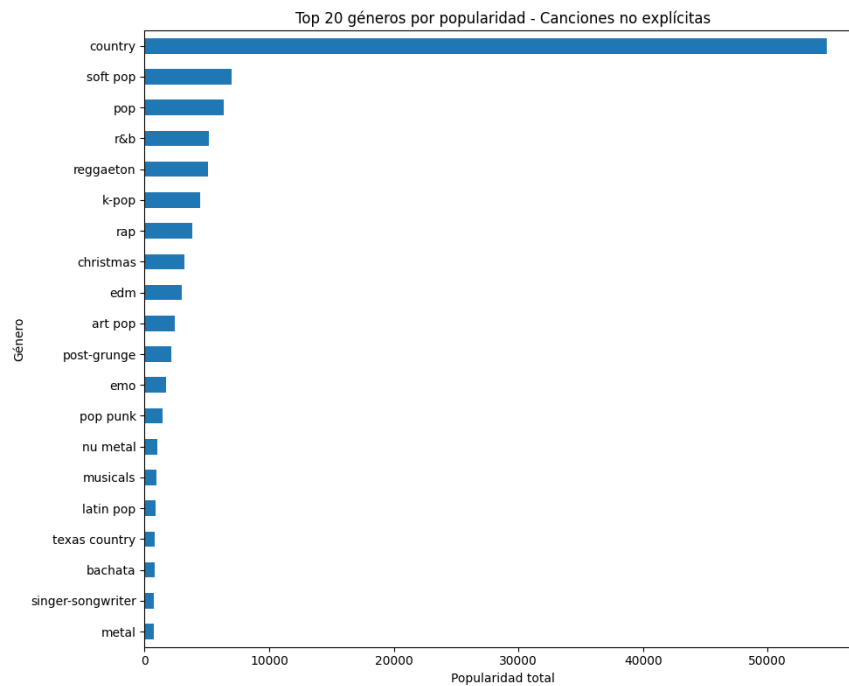
#### Gráfica 2: Popularidad total por género para un nivel de explicitud

Implemente una función que reciba el DataFrame y un valor booleano indicando si se quieren analizar canciones explícitas o no explícitas. La función debe filtrar las canciones según ese criterio, agrupar por género, sumar la popularidad total de cada género, ordenar de mayor a menor, tomar los 20 géneros más populares, y generar una gráfica de barras horizontales.

La gráfica debe incluir:

- Título indicando si son canciones explícitas o no explícitas

- Etiqueta del eje X: “Popularidad total”
- Etiqueta del eje Y: “Género”
- El eje Y invertido para mostrar el género más popular en la parte superior
- Solo los 20 géneros con mayor popularidad total



## Entrega

Entregue un archivo comprimido (ZIP) que contenga:

1. **matrices.py**: Implementación de las Actividades 1-4
2. **analisis\_musica.py**: Implementación de la Actividad 5

Suba el archivo comprimido a través de Brightspace en el laboratorio del Nivel 4 designado como "N4-L3: Matrices y análisis de datos".