

Introducción al lenguaje de programación Python

Juan Carlos Barrera Guevara, código:160005003

Diego Alejandro Machado Tovar, código:160005017

Universidad de los Llanos

Villavicencio, Colombia

Abstract— Este informe presenta el desarrollo de una práctica introductoria en Python aplicada al análisis de datos de vacunación antirrábica en Bogotá. El programa implementado permitió comprender el uso de variables, operadores, estructuras condicionales, ciclos y listas, así como la importación y procesamiento de datos mediante la librería Pandas. Posteriormente, se generaron diferentes tipos de gráficos con Matplotlib (línea, barras y dispersión) para visualizar la información entre sectores público y privado, así como su relación con el censo poblacional. Los resultados evidencian la utilidad de Python como herramienta versátil para el análisis de datos y la visualización de información en el contexto académico y profesional de la ingeniería de sistemas.

I. INTRODUCCIÓN

El lenguaje Python es una de las herramientas más utilizadas en la actualidad por su simplicidad sintáctica y la amplia disponibilidad de bibliotecas para el análisis de datos, inteligencia artificial y desarrollo de software. En esta práctica de laboratorio se implementó un programa que integra los conceptos básicos de la programación (variables, operadores, estructuras condicionales, ciclos y listas), así como la importación y análisis de datos externos mediante la librería Pandas y la visualización con Matplotlib.

El propósito central de la práctica fue aplicar los fundamentos de Python en un entorno práctico, logrando manipular datos de un archivo CSV relacionado con la vacunación antirrábica en Bogotá, para posteriormente generar gráficos que apoyen la interpretación de los resultados.

II. PROCEDIMIENTO

Para iniciar la práctica se configuró el entorno de desarrollo instalando Python 3 y utilizando Visual Studio Code como editor. También se instalaron las bibliotecas necesarias con el comando `pip install pandas matplotlib numpy`, asegurando la disponibilidad de herramientas para el manejo de datos y la generación de gráficos. El primer paso consistió en la definición de variables y la aplicación de operadores. Se declararon variables de diferentes tipos, tales como enteros, flotantes, cadenas de texto y booleanos, lo que permitió trabajar con distintos tipos de datos:

```
entero = 10
decimal = 3.141516
cadena = "Hello world ;)"
booleano = True
```

Con estas variables se realizaron operaciones aritméticas básicas como suma, resta, multiplicación y división, además de operadores lógicos y de comparación:

```
suma = entero + 5
resta = entero - 2
multiplicacion = entero * 3
division = entero / 2
```

```
print("¿La variable entero > 5?", entero > 5)
print("booleano AND (entero > 5):", booleano and
(entero > 5))
```

Posteriormente, se implementaron estructuras condicionales utilizando `if`, `elif` y `else`. Este bloque de código permitió evaluar si una variable numérica era mayor, igual o menor que 10, controlando así el flujo de ejecución según la condición planteada:

```
numero = 8
if numero > 10:
    print("El número es mayor que 10")
elif numero == 10:
    print("El número es igual a 10")
else:
    print("El número es menor que 10")
```

El siguiente apartado se centró en los ciclos. Con un ciclo `for` se construyó una tabla de multiplicar para un número ingresado por el usuario, demostrando cómo recorrer un rango de valores:

```
tabla = int(input("Digite un numero: "))
for i in range(0, 11):
    print(tabla, " x ", i, " = ", (tabla*i))
```

De manera complementaria, se utilizó un ciclo **while** que imprimió de manera secuencial un contador desde 1 hasta 5:

```
contador = 1
while contador <= 5:
    print(contador, end=" ")
    contador += 1
```

Luego, se trabajó con listas para demostrar operaciones sobre colecciones de datos. Se creó una lista de frutas y se aplicaron operaciones de inserción, eliminación y modificación. Finalmente, la lista fue recorrida mediante dos técnicas: con `len()` para obtener los índices y con `enumerate()` para acceder al índice y al valor simultáneamente:

```
frutas = ["manzana", "pera", "uva"]
frutas.append("naranja")
frutas.remove("pera")
frutas[1] = "sandía"

for i in range(len(frutas)):
    print(f"Índice {i} -> {frutas[i]}")

for i, fruta in enumerate(frutas):
    print(f"Índice {i} -> {fruta}")
```

A continuación, se avanzó hacia la importación de datos externos utilizando la biblioteca `pandas`. Se cargó un archivo CSV con información de la cobertura de vacunación antirrábica en Bogotá, el

cual fue obtenido del portal oficial de Datos Abiertos Bogotá de la Alcaldía Mayor. Este recurso público contiene información oficial y actualizada de las campañas de salud adelantadas en el Distrito Capital. Una vez importado, se mostraron las primeras filas del dataset con el fin de verificar su correcta lectura y garantizar que los datos estuvieran disponibles para el análisis posterior.

```
Datos = pd.read_csv("archivo.csv",
encoding="latin1", sep=";")
print(datos.head())
```

En el proceso de importación del archivo CSV se emplearon los parámetros `encoding="latin1"` y `sep=";"` dentro de la función `read_csv()` de la librería `pandas`. El primero se utilizó debido a que los datos provenían de fuentes oficiales en Colombia y contenían caracteres propios del idioma español, como tildes y la letra ñ. En muchos casos, estos archivos no están codificados en formato UTF-8, que es el estándar por defecto de Python, sino en ISO-8859-1 (latin1). Por esta razón, fue necesario especificar el tipo de codificación para evitar errores de lectura y asegurar que los caracteres se visualizaran correctamente.

El segundo parámetro, `sep=";"`, se incluyó porque el archivo no utilizaba la coma como delimitador de columnas, sino el punto y coma (;). Esta práctica es habitual en países hispanohablantes, donde la coma se emplea como separador decimal (por ejemplo, 3,14). De esta forma, indicar explícitamente que el separador es ; permitió que los datos se organizaran en columnas de manera adecuada, evitando que toda la información quedara agrupada en una sola columna.

Finalmente, se generaron distintos gráficos mediante `matplotlib` para visualizar la información. Se elaboraron un gráfico de líneas que comparó la evolución de vacunados en los sectores público y privado, un gráfico de barras para representar el número de vacunados por localidad, un gráfico de dispersión para analizar la relación entre censo poblacional y vacunación, y un gráfico comparativo de barras para contrastar sector público y privado en cada localidad:

```
plt.plot(datos["Año"], datos["Vacunados
sector_publico"], marker="o", label="Sector
Público")
plt.plot(datos["Año"], datos["Vacunados
sector_privado"], marker="s", label="Sector
Privado")
plt.title("Vacunados por Año - Sector Público vs
Privado")
plt.xlabel("Año")
plt.ylabel("Número de Vacunados")
plt.legend()
plt.show()
```

III. RESULTADOS

Con base en los datos importados y procesados, se generaron diferentes visualizaciones que permitieron analizar la cobertura de vacunación en Bogotá desde dos perspectivas: temporal y geográfica. A continuación, se presentan las gráficas obtenidas junto con su respectivo análisis.

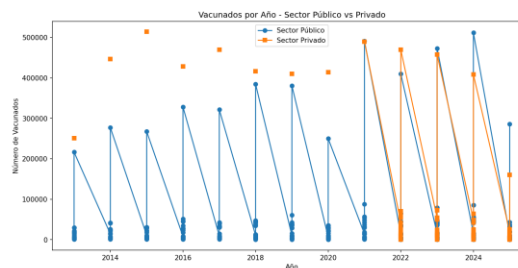


Fig. 1 Vacunados por Año – Sector Público vs Privado

Este gráfico de líneas muestra la evolución de la vacunación en ambos sectores a lo largo de los años. Se observa que el sector público presenta una tendencia más fluctuante, mientras que el privado exhibe picos significativos en determinados periodos. La conclusión principal es que la cobertura ha variado de manera considerable en el tiempo, reflejando posiblemente diferencias en la gestión de campañas o en la asignación de recursos.

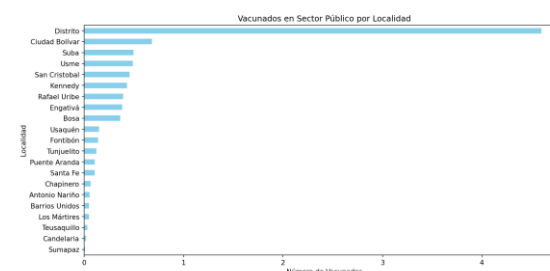


Fig. 2 Vacunados en Sector Público por Localidad

La gráfica de barras horizontales evidencia que el “Distrito” concentra la mayor cantidad de vacunados en el sector público, superando con amplitud al resto de localidades. Además, zonas como Ciudad Bolívar, Suba y Usme presentan cifras relativamente altas en comparación con otras. En conclusión, la distribución es desigual y responde directamente a la densidad poblacional de cada localidad.

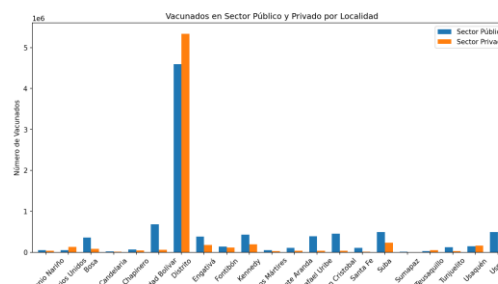


Fig. 3 Vacunados en Sector Público y Privado por Localidad

El gráfico comparativo refleja que en la mayoría de localidades el sector público lidera la vacunación frente al privado. Sin embargo, en algunas zonas específicas ambos sectores muestran cifras similares. El caso más destacado vuelve a ser el “Distrito”, donde ambos sectores concentran una cobertura significativamente mayor. La conclusión que se desprende es que la vacunación depende principalmente de la gestión pública, aunque el sector privado cumple un papel de apoyo complementario.

IV.CONCLUSIONES

El manejo de datos externos con Pandas permitió trabajar con información real (vacunación en Bogotá), mostrando la importancia de la programación en el análisis de datos.

La visualización con Matplotlib permitió contrastar la evolución de la vacunación en sectores público y privado, así como la relación con censos poblacionales.

REFERENCIAS

- [1] Secretaría Distrital de Salud – Alcaldía Mayor de Bogotá D.C., *Cobertura de vacunación antirrábica en Bogotá D.C.*, Datos Abiertos Bogotá. Disponible en: <https://datosabiertos.bogota.gov.co/dataset/cobertura-de-vacunacion-antirrabica-en-bogota-d-c/resource/7dae7b92-76bf-4eae-856d-3519ed203a2b>.