**TECNOLÓGICO DE ESTUDIOS SUPERIORES CHALCO**

**CARRERA:** INGENIERÍA EN SISTEMAS COMPUTACIONALES

**MATERIA:** INTRODUCCIÓN A LA CIENCIA DE DATOS

**DOCENTE**: MTRO. IVAN AZAMAR PALMA

**ALUMNOS:** VAZQUEZ GARCIA LUZ LIZETH

**MATRÍCULA:** 202227005

**SEMESTRE:** SEPTIMO SEMESTRE

**GRUPO:** 4771 SEMI-PRESENCIAL

**TEMA:** PRÁCTICA 2

**FECHA:** OCTUBRE 2025

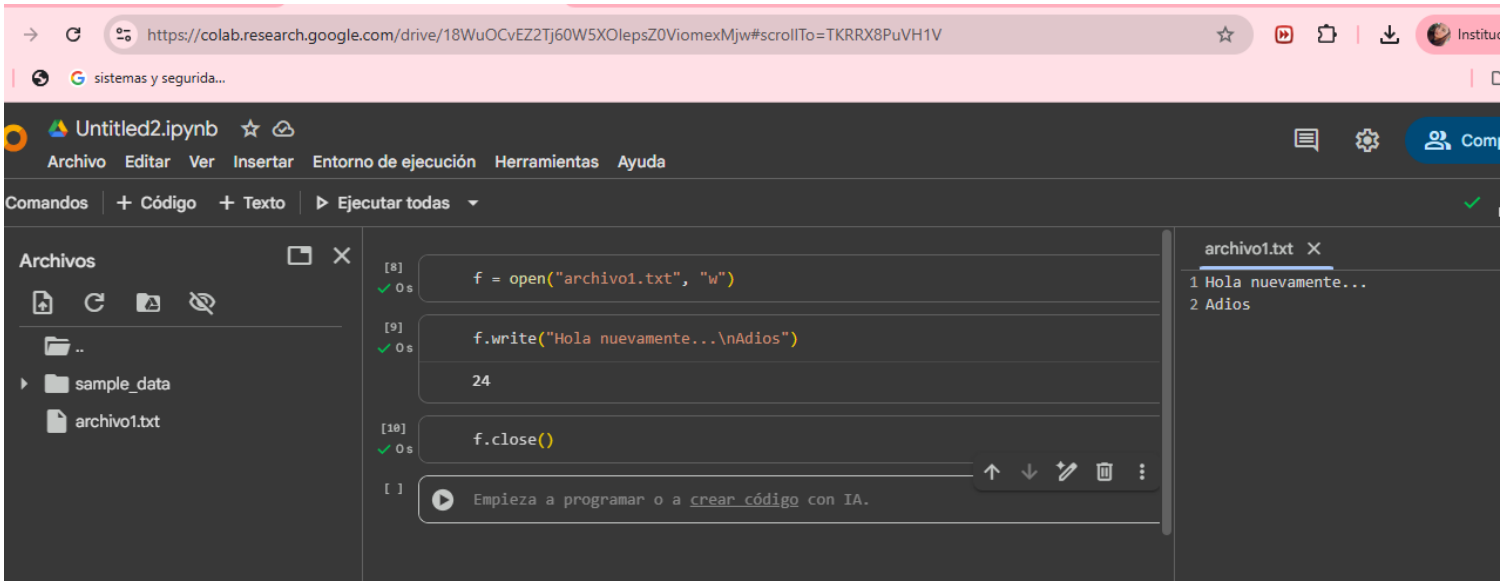
# INDICE

## INTRODUCCIÓN

En esta práctica se exploraron conceptos fundamentales del manejo de archivos y procesamiento de datos utilizando el lenguaje de programación Python. Python es una herramienta poderosa y versátil, ampliamente utilizada en el análisis de datos, automatización y desarrollo de aplicaciones. A través de ejercicios prácticos, se trabajó con la creación, lectura y escritura de archivos de texto y archivos CSV, lo cual es esencial para la manipulación de grandes volúmenes de información y su posterior análisis. Además, se aplicaron técnicas para generar datos aleatorios y estructurarlos en formatos estándar, como archivos CSV, facilitando su lectura y procesamiento con módulos especializados. Estas actividades permitieron comprender la importancia del manejo eficiente de datos y cómo Python facilita estas tareas con su sintaxis sencilla y bibliotecas integradas, consolidando así conocimientos clave para el desarrollo de proyectos en ciencia de datos y programación general.

# DESARROLLO DE LA PRÁCTICA

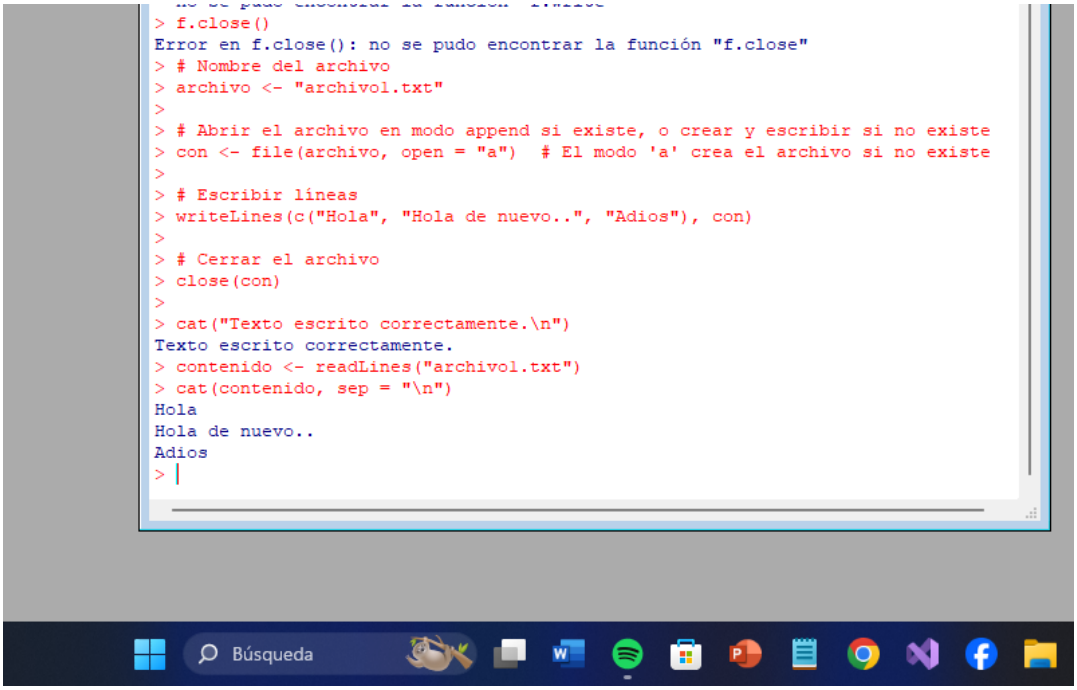
## 1. OPERACIÓN BÁSICA DE ESCRITURA:



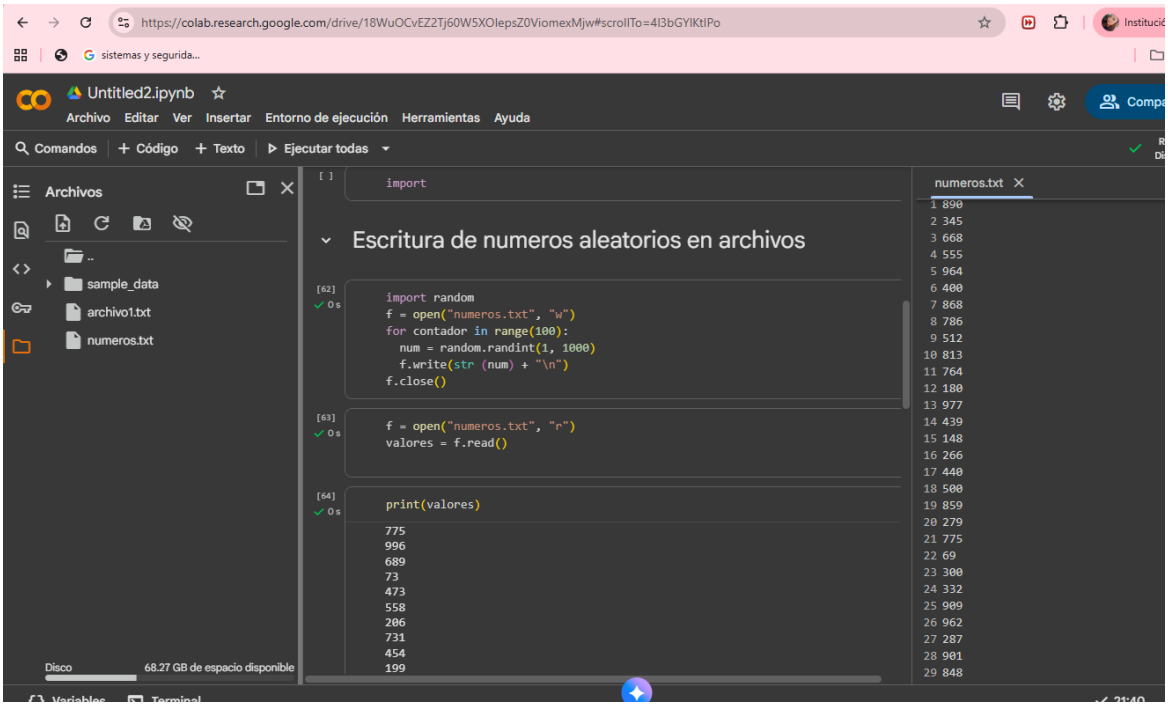

- open(): Abre un archivo.
- "archivo1.txt": Es el nombre del archivo que se va a abrir.
- "a": Es el modo en que se abre el archivo:
  - "a" significa append (agregar). Esto quiere decir que:
    - Si el archivo no existe, Python lo crea.
    - Si el archivo ya existe, el contenido nuevo se agrega al final sin borrar lo anterior.

El archivo se abre y se guarda en la variable f.

### En lenguaje R

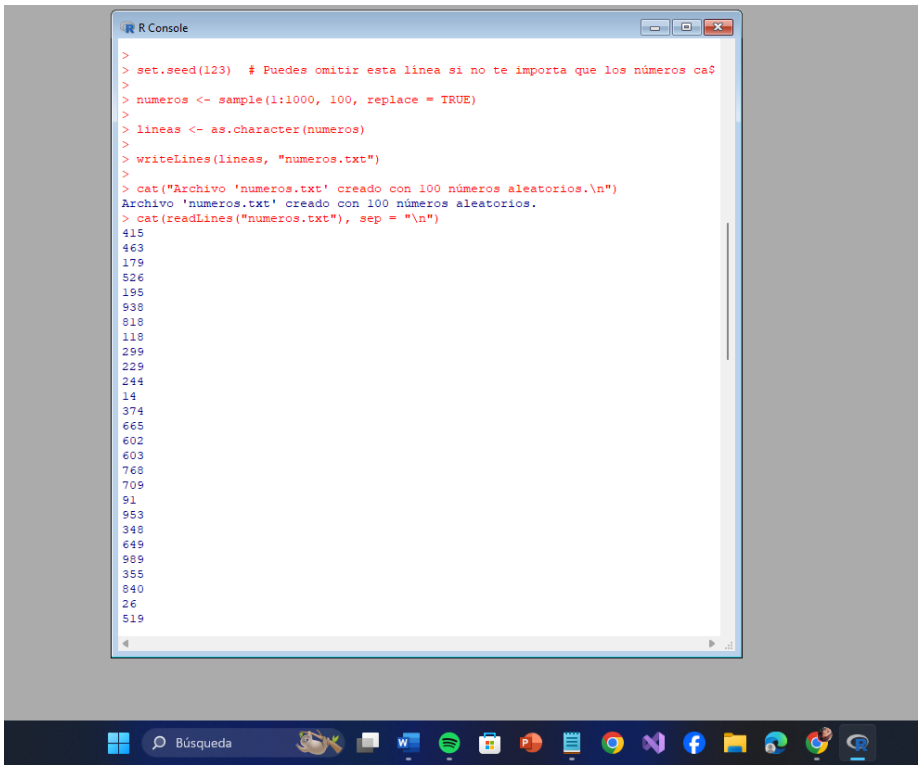


## 2. ESCRITURA DE NÚMEROS:

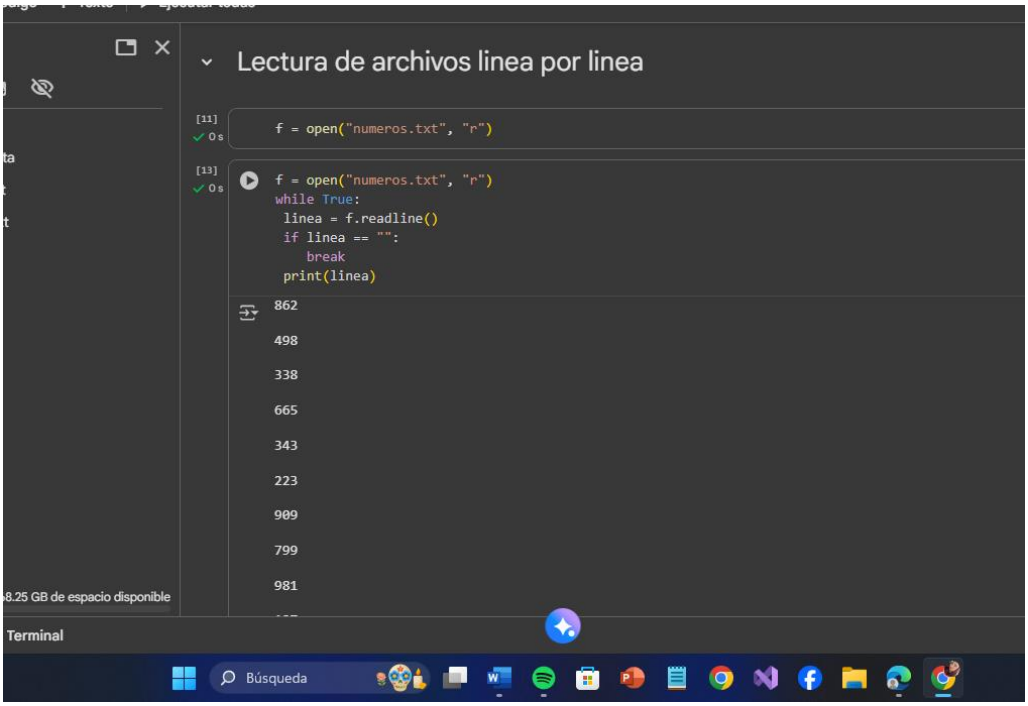



- Este código genera **100** números aleatorios entre 1 y 1000 y los guarda en un archivo llamado numeros.txt, escribiendo un número por línea.
- Primero se importa el módulo random para poder generar números aleatorios. Luego se abre el archivo en modo escritura ("w"), lo que borra su contenido si ya existe.
- Dentro de un bucle for que se repite 100 veces, se genera un número aleatorio con random.randint(1, 1000), se convierte a texto y se escribe en el archivo seguido de un salto de línea.
- Finalmente, se cierra el archivo con f.close() para guardar los cambios.

### Lenguaje R

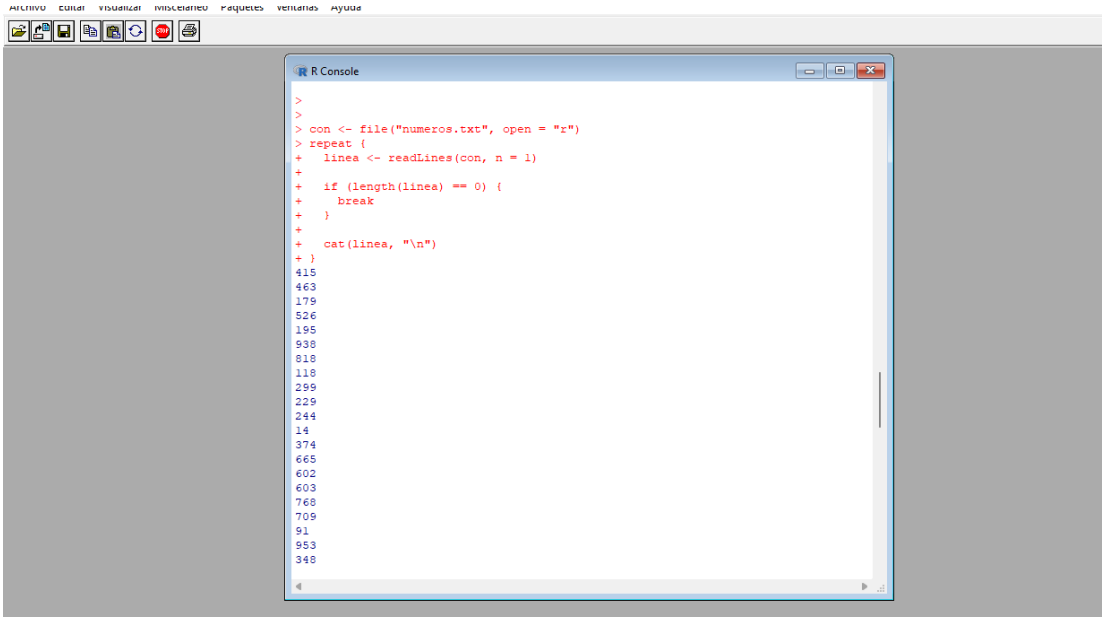


## 3. LECTURA DE ARCHIVOS LÍNEA POR LÍNEA

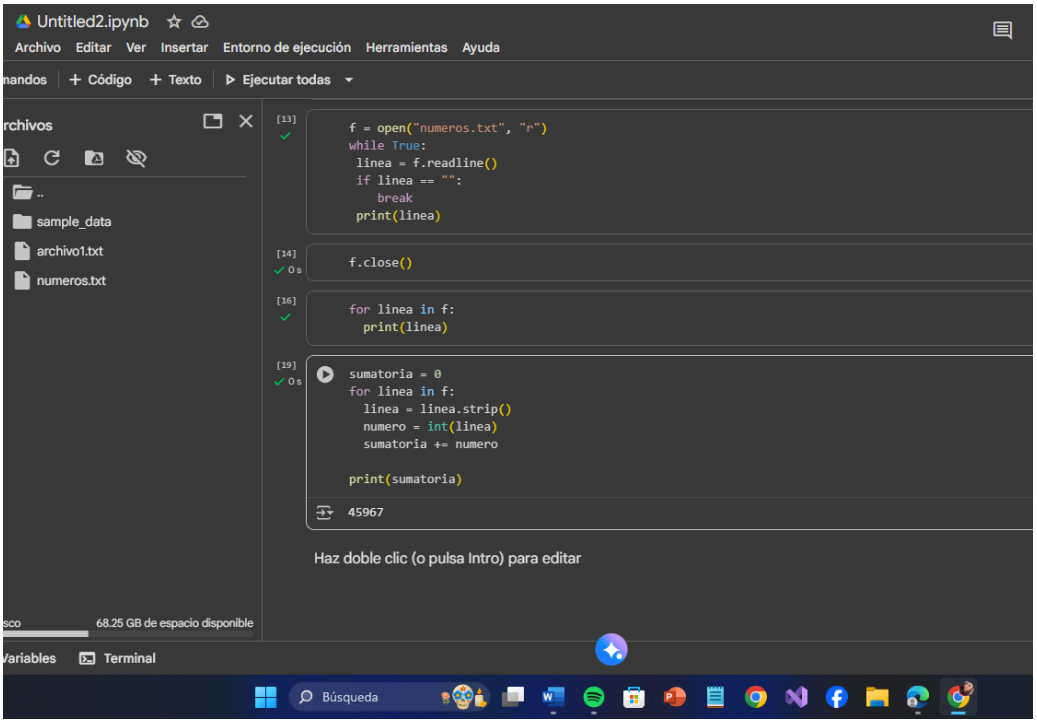



- Este código abre el archivo numeros.txt en modo lectura ("r") y lee su contenido línea por línea. Usa un bucle while True que se ejecuta indefinidamente hasta que ya no hay más líneas por leer.
- Dentro del bucle, f.readline() lee una línea del archivo; si la línea está vacía (""), significa que se llegó al final del archivo y se sale del bucle con break.
- Si no, imprime la línea con print(linea).
- Como cada línea ya incluye un salto de línea (\n), el resultado puede mostrarse con líneas en blanco entre ellas, a menos que se use print(linea.strip()) para limpiar los saltos.

### Lenguaje R

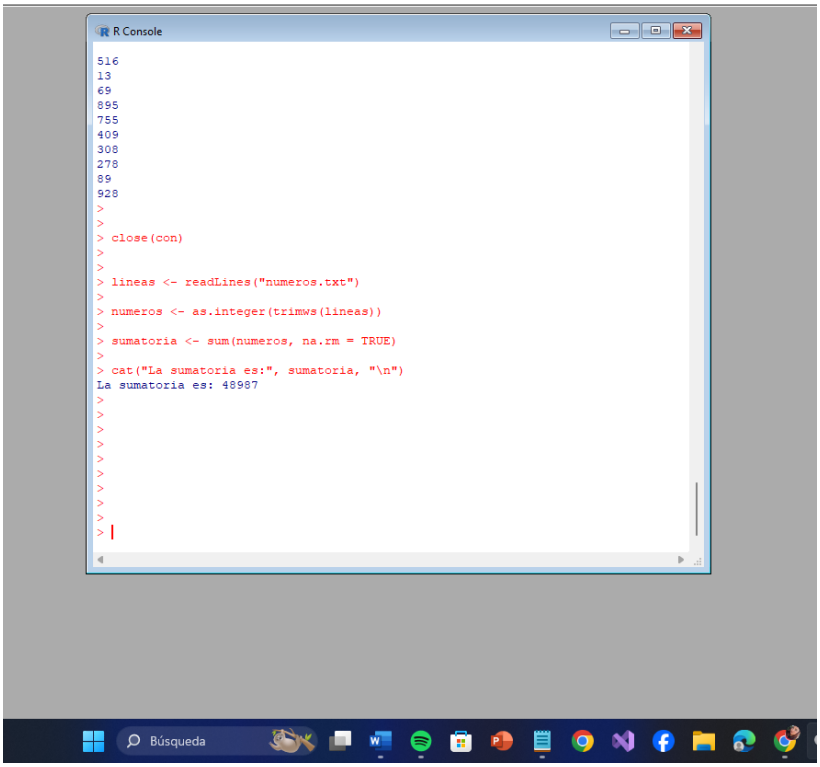


## 4. SUMATORIA

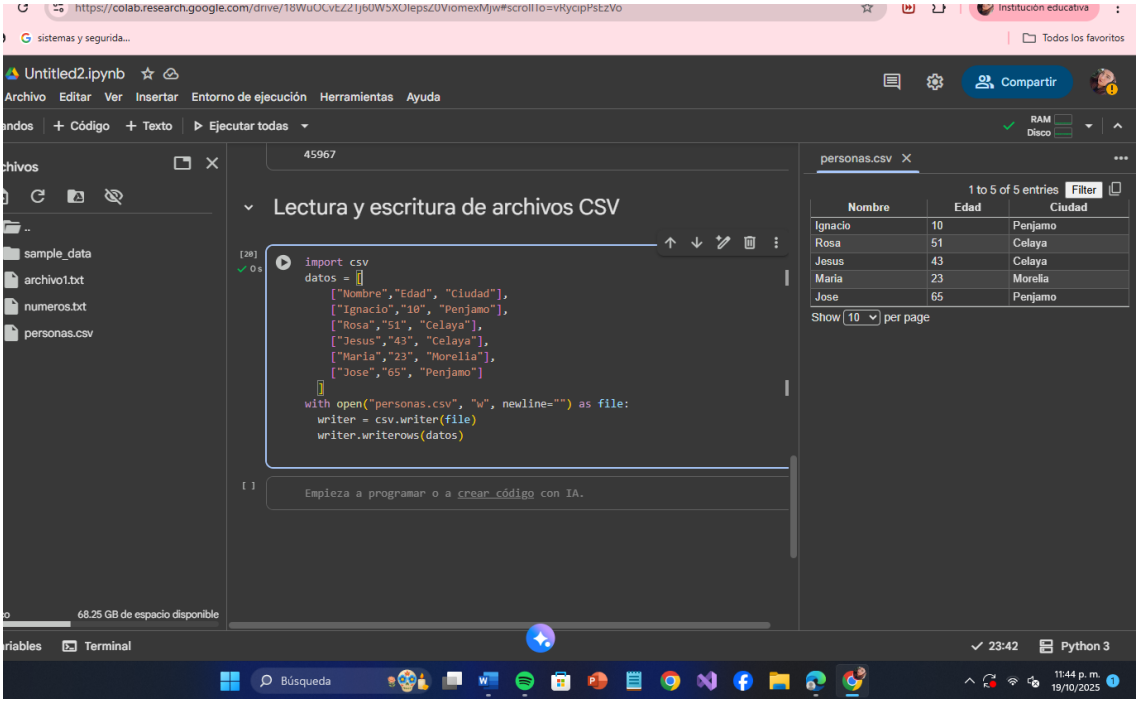



- Este código calcula la suma de todos los números guardados en el archivo numeros.txt. Supone que f ya está abierto en modo lectura.
- Recorre cada línea del archivo con un bucle for, elimina los espacios en blanco y saltos de línea con strip(), convierte el texto a número entero con int(), y lo va sumando en la variable sumatoria.
- Al final, imprime el total de la suma de todos los números del arch

### Lenguaje R

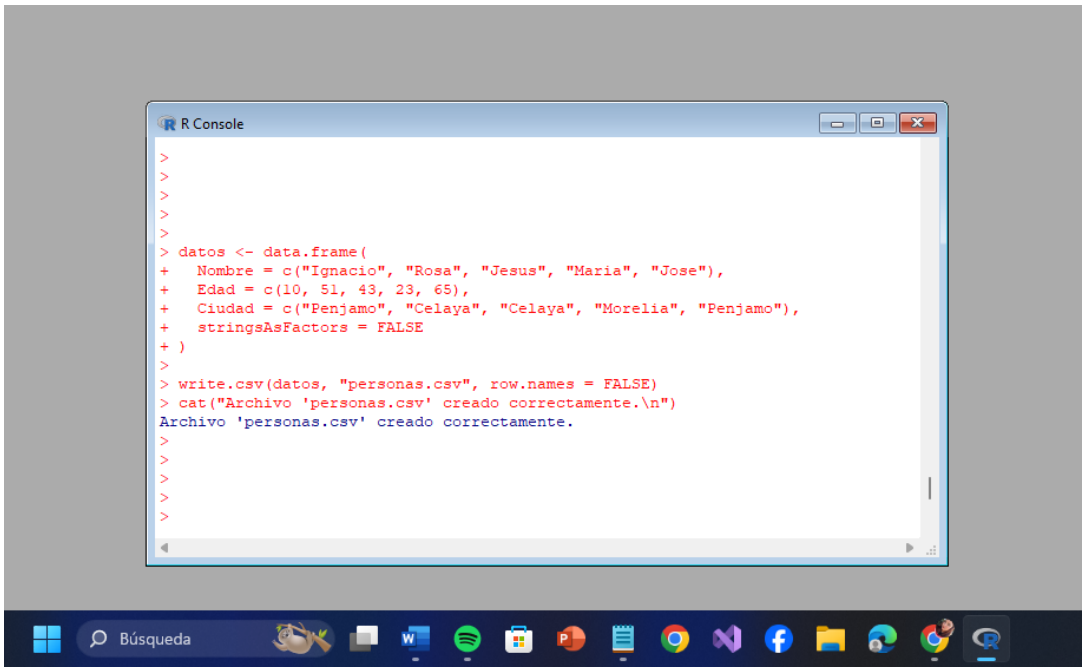


## 5. ARCHIVOS CSV EN MODO ESCRITURA

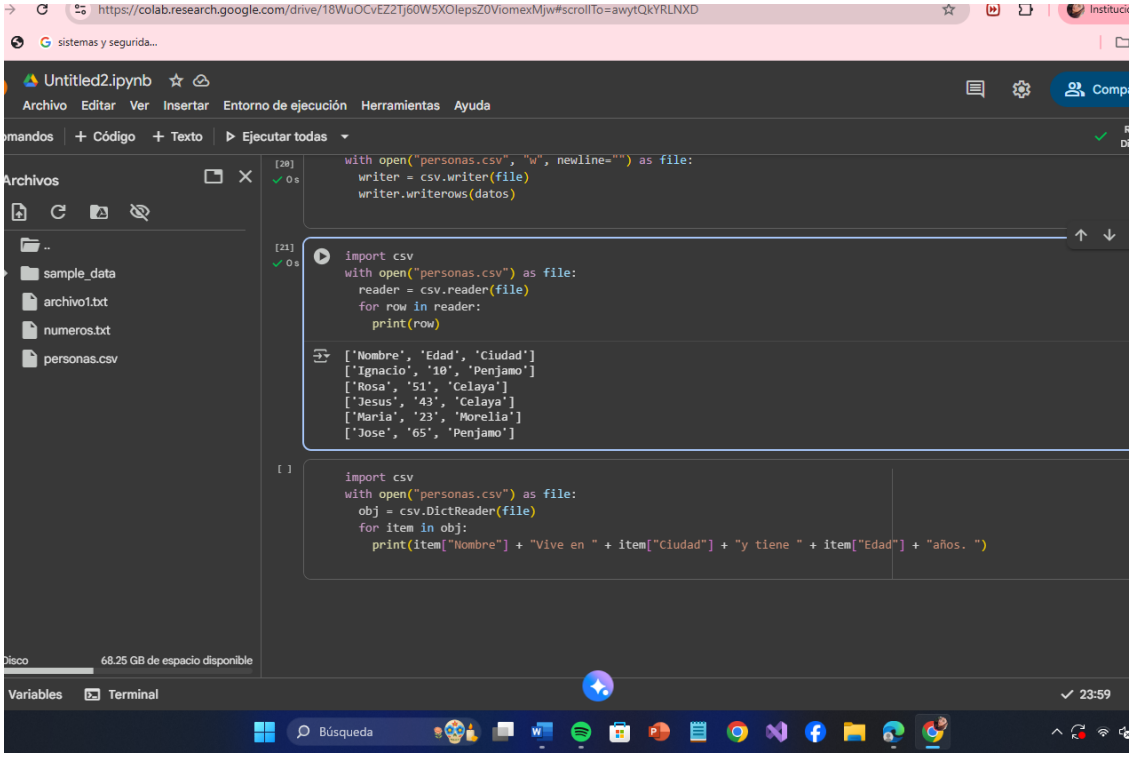



- Abre (o crea) el archivo personas.csv en modo escritura ("w").
- Usa csv.writer() para escribir datos en formato CSV.
- writer.writerows(datos) escribe todas las filas del arreglo datos en el archivo.
- newline="" evita que se inserten líneas vacías extra entre cada fila

### Lenguaje R:

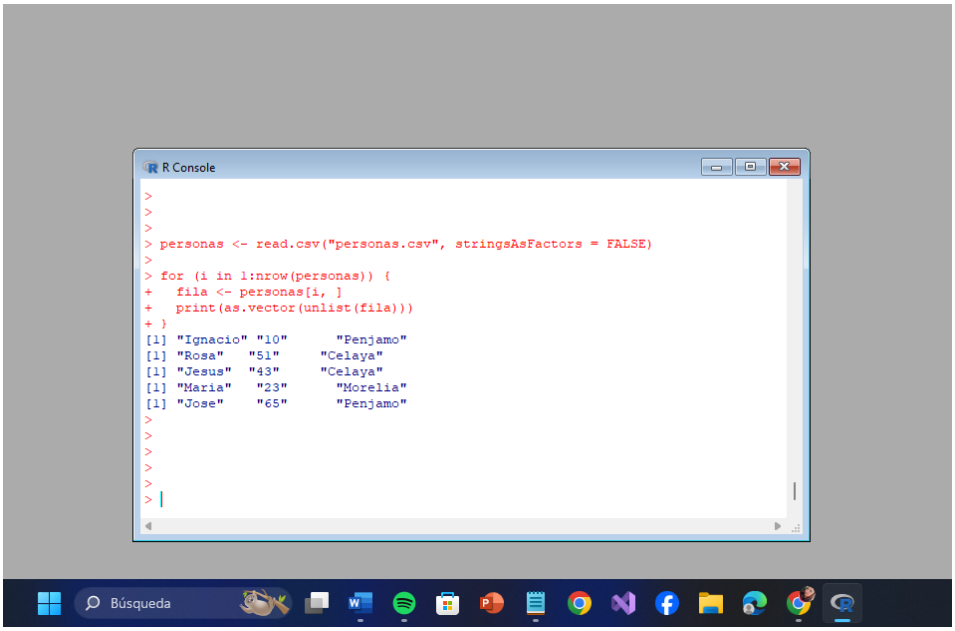


## 6. ARCHIVOS CSV CONTENIDO LÍNEA POR LÍNEA

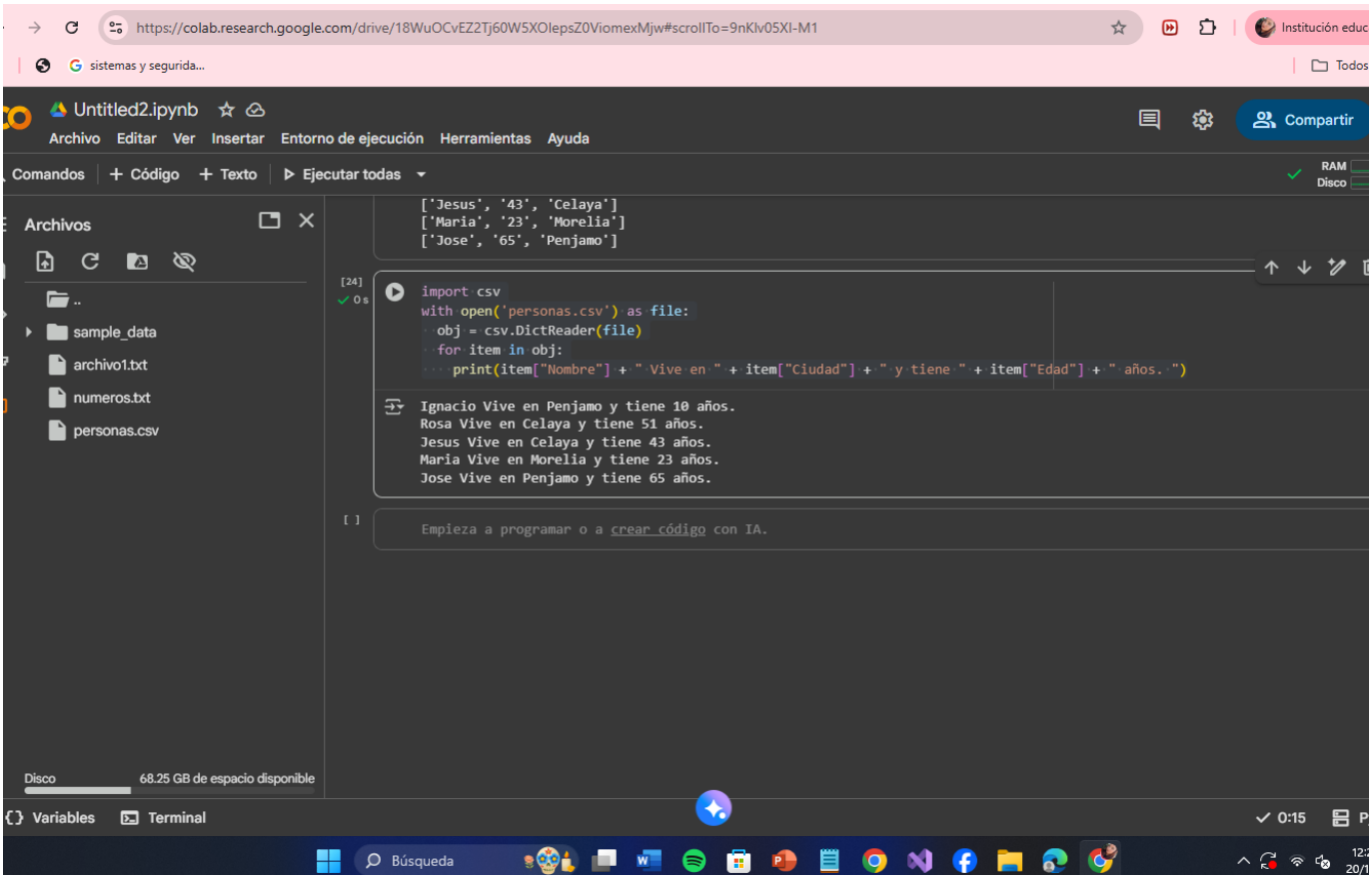



- Este código lee el archivo personas.csv y muestra su contenido línea por línea.
- Primero se importa el módulo csv, que permite trabajar con archivos CSV de forma sencilla.
- Luego se abre el archivo personas.csv en modo lectura usando with open(...), lo que asegura que el archivo se cierre automáticamente al terminar.
- Con csv.reader(file) se crea un lector que interpreta cada línea del archivo como una lista de valores separados por comas.
- Finalmente, se recorre cada fila del archivo con un bucle for y se imprime cada fila como una lista, mostrando por pantalla el contenido completo del archivo CSV.

### Lenguaje R:

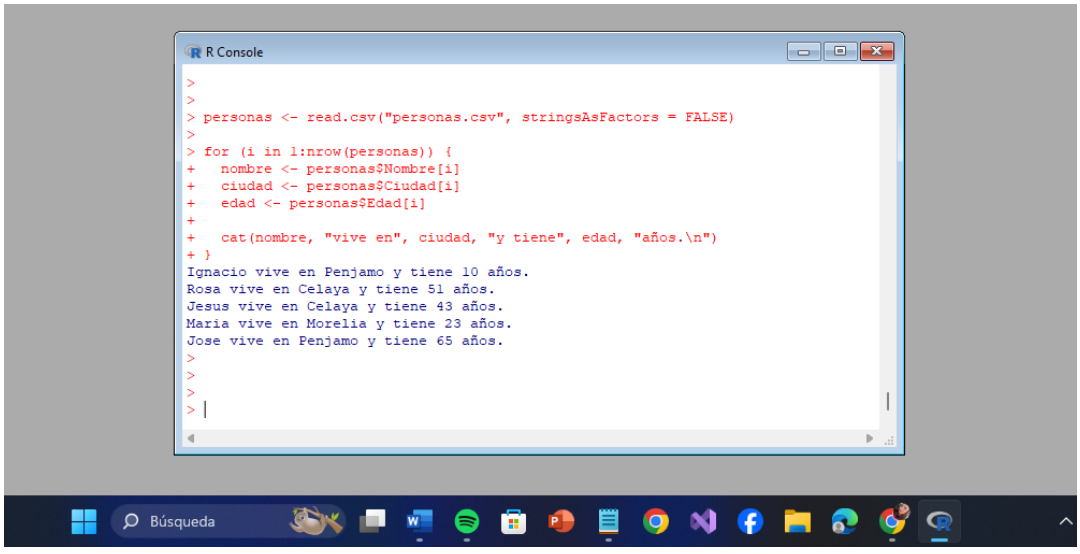


## 7. ARCHIVOS CSV LEER CON DICTREADER




- Este código lee el archivo personas.csv y muestra un mensaje personalizado para cada persona usando los datos del archivo.
- Primero se importa el módulo csv. Luego se abre el archivo con with open(...), y se utiliza csv.DictReader(file) para leer cada fila del CSV como un diccionario, donde las claves son los encabezados de la primera fila del archivo (por ejemplo: "Nombre", "Edad", "Ciudad").
- En el bucle for, cada fila se guarda en la variable item, y se accede a los datos por su nombre de columna. Finalmente, se imprime una frase para cada persona como: "Ignacio Vive en Penjamo y tiene 10 años.", usando los valores del diccionario.

## Lenguaje R



## CONCLUSIÓN

Python y R son dos lenguajes de programación ampliamente utilizados en el análisis de datos, la ciencia de datos y la programación en general, cada uno con sus fortalezas particulares. Python se destaca por su sintaxis clara y sencilla, su gran comunidad, y la gran cantidad de bibliotecas que facilitan tareas de manipulación de datos, automatización y desarrollo web, entre otras. En contraste, R es especialmente fuerte en análisis estadístico y visualización avanzada, siendo muy popular en la comunidad académica y de investigación debido a su enfoque en la estadística y las gráficas.

Durante las actividades realizadas, se exploraron varias funciones básicas de Python, especialmente orientadas al manejo de archivos y la generación de datos aleatorios. Se aprendió a abrir, escribir y leer archivos de texto, cómo manipular datos línea por línea y cómo trabajar con archivos CSV usando el módulo csv, incluyendo tanto lectura básica como lectura avanzada con DictReader para un acceso más semántico a los datos. Estas actividades son fundamentales para entender cómo Python facilita el procesamiento y análisis de datos en formatos comunes y cómo se puede automatizar el trabajo con grandes volúmenes de información.

Estas tareas demuestran la versatilidad de Python para actividades relacionadas con la manipulación de datos, un campo donde R también es muy competente pero con una orientación más estadística. Aprender a manejar archivos y a utilizar bibliotecas estándar de Python como csv es un paso esencial para cualquier persona que quiera desenvolverse en programación orientada a datos y análisis. En resumen, el uso de Python en estas actividades confirma su rol como una herramienta poderosa y accesible para programadores de todos los niveles, complementando las capacidades que ofrece R en entornos donde el análisis estadístico riguroso es prioridad.

## BIBLIOGRAFÍA:


- Python Software Foundation. (2023). *Python documentation*. https://docs.python.org/3/tutorial/inputoutput.html
- Real Python. (2023). *Working with CSV Files in Python*. https://realpython.com/python-csv/
- The R Project for Statistical Computing. (2023). *R Documentation*. https://cran.r-project.org/manuals.html
- Welling, L. (2021). *Python vs R for Data Science*. Towards Data Science. https://towardsdatascience.com/python-vs-r-for-data-science-4d348e401dd4
- GeeksforGeeks. (2023). *File Handling in Python*. https://www.geeksforgeeks.org/file-handling-python/