

Actividad: Tipos de Datos y Adquisición de Datos en Python

Parte 1: Tipos de Datos (Estructurados y No Estructurados)

Objetivos:

- Comprender la diferencia entre datos estructurados y no estructurados.
 - Aprender a trabajar con ambos tipos de datos en Python.
-

Instrucciones Paso a Paso

1. Investigación Teórica

Objetivo: Investigar y entender las diferencias entre datos estructurados y no estructurados.

1. **Lee y resume** la siguiente información sobre los tipos de datos:
 - **Datos estructurados:** Son aquellos datos organizados en un formato claro y predecible, como bases de datos o archivos CSV. Se pueden organizar en filas y columnas (tablas).
 - **Datos no estructurados:** Son datos que no tienen un formato específico o rígido, como textos, imágenes, videos, etc.
2. **Responde las siguientes preguntas** en tu informe:
 - ¿Cuáles son las principales características de los datos estructurados y no estructurados?
 - ¿Cuáles son las ventajas y desventajas de trabajar con cada tipo de dato?
 - Da ejemplos de datos no estructurados más allá de texto (por ejemplo, audio o imágenes).
 - ¿Qué desafíos existen al trabajar con cada tipo de dato?

2. Actividad Práctica

Objetivo: Cargar y explorar datos estructurados y no estructurados en Python.

1. Trabajando con Datos Estructurados:

- **Paso 1:** Descarga un conjunto de datos en formato CSV desde un sitio web como [Kaggle](#) o datos sintéticos.
- **Paso 2:** Carga el archivo CSV en un DataFrame de Pandas. Abre una terminal o un Jupyter Notebook y ejecuta el siguiente código:

```
import pandas as pd
# Carga de archivo CSV
df = pd.read_csv('ruta/a/tu/archivo.csv')
# Ver las primeras filas
print(df.head())
# Información general del DataFrame
print(df.info())
```

2. Trabajando con Datos No Estructurados (Texto):

- **Paso 1:** Descarga un archivo de texto (puede ser un archivo de tweets o comentarios en un foro).
- **Paso 2:** Carga el archivo de texto en Python y realiza un análisis simple. Puedes usar el siguiente código:

```
with open('ruta/a/tu/archivo.txt', 'r') as file:
    text_data = file.read()
# Muestra las primeras líneas
print(text_data[:500])
```

- **Paso 3:** Usa herramientas de procesamiento de texto básicas como `split()` para dividir el texto en palabras:

```
words = text_data.split()
print(words[:20]) # Muestra Las primeras 20 palabras
```

3. Entregable:

- Escribe un informe que incluya:
 - Respuestas a las preguntas teóricas.
 - El código utilizado para cargar y explorar los datos (tanto estructurados como no estructurados).
 - Una breve interpretación de los datos y cualquier hallazgo interesante que hayas observado.
-