Almacenamiento datos

Formatos de texto plano

La importancia de los datos

- La humanidad está transitando por un período en que la **digitalización** está siendo parte de cada vez **más actividades del quehacer humano**, como consecuencia de la cuarta revolución industrial.
- La **producción/recolección masiva de datos** es una característica en **crecimiento** en todos los ámbitos del quehacer de nuestra civilización (incluyendo la investigación científica).
- La **astronomía** es una ciencia clave dada su amplia **experiencia** trabajando en grandes conjuntos de datos con el fin de obtener la mayor información posible y realizar descubrimientos científicos.

El definir estándares para almacenar y compartir datos se ha convertido en una necesidad en todas las dimensiones del comportamiento humano.

Puntos a considerar sobre sus datos

1. **Almacenamiento**
   Definir dónde permanecerán luego del uso, proyectar tiempo de vida, riesgos de pérdida.
2. **Composición**
   Tipo de dato almacenado: numérico/categórico, relacional/no-relacional, homogéneo/heterogéneo)
3. **Tamaño**
   Tener una idea de la cantidad de Bytes que ocupa la información contenida.
4. **Comunicación**
   Considerar el público objetivo que usará los datos. Fijar estándares de almacenamiento adecuados. Incorporar metadatos descriptivos,
5. **Escalamiento**
   Determinar si los datos son estáticos o se modificarán con el tiempo. Proyectar crecimiento de los mismos.

Almacenamiento en texto plano

Son archivos donde la **representación de los datos** se realiza en **texto sin formato**. Usualmente siguen estructuras similares a tablas con *filas* y *columnas*, donde cada *fila* un solo registro, mientras que las *columnas* representan los campos.

Al ser un formato simple, son ideales para **compartir información** entre diferentes aplicaciones. Son el estándar en muchos software y apps web. Suelen ser utilizados para almacenamiento temporal. Presentan **complicaciones con escalamiento hacia datos masivos**.

Algunos tipos de estructuras (tipos de archivo) de este formato utilizados:

.TXT
Archivo básico y libre. Puede contener texto sin ningún formato particular.

.CSV (.TSV)
Usan comas (tabs) como delimitadores de campos. CSV es muy utilizado por la comunidad.

Formatos estructurados

.XML
Es un lenguaje diseñado para representar datos estructurados en un formato legible por/para humanos

.JSON
Formato muy utilizado en apps web para compartir información compleja y estructurada.

Almacenamiento en binario

Formato binario se refiere a guardar datos **en una representación que es directamente legible por la máquina**, en lugar de una representación legible por humanos como el texto. Es decir, se almacena directamente el conjunto de bytes que representa el dato en la memoria.

Ventajas

Tamaño: se suele ocupar menos espacio de almacenamiento.
Rapidez: la lectura y escritura (I/O) es directa, sin procesamiento.
Precisión: el dato se almacena sin reducción de cifras significativas.

Desventajas

Estructura: es necesario conocer exactamente los tipos de datos almacenados y su orden.
Arquitectura: algunas PCs pueden usar representaciones distintas de los datos.

Almacenamiento en binario

```python
# Escritura de datos:

import struct

# Datos a guardar: una int y un float
data = (5, 2.5)

# Empaquetar en formato binario:
# 'i' representa un integer y
# 'f' un float en la notación struct
binary_data = struct.pack('if', *data)

# Escribir los datos empaquetados
# en archivo binario:

with open('data.bin', 'wb') as f:
    f.write(binary_data)
```

```python
# Leer datos desde un archivo
binario:

with open('data.bin', 'rb') as f:
    binary_data = f.read()

# Desempaquetar los datos desde
# el formato binario:
data = struct.unpack('if',
                     binary_data)

print(data)  # Salida: (5, 2.5)
```

Almacenamiento en HDF5

HDF5 es un formato de archivo y una biblioteca diseñada para **almacenar** y **organizar grandes cantidades de datos**. El nombre "Hierarchical Data Format" se refiere a la capacidad de HDF5 para almacenar datos en una **estructura jerárquica**, similar a un sistema de archivos de un disco.
HDF5 es ampliamente utilizado en diversas disciplinas y aplicaciones, desde la simulación computacional y el análisis de datos experimentales hasta la astronomía y la genómica.

- Sitio web oficial (librería también disponible desde sus repositorios el SO): https://www.hdfgroup.org/solutions/hdf5/
- Visualizador de archivos (Java): https://www.hdfgroup.org/downloads/hdfview/
- Acceso desde Python: https://www.h5py.org/

Formato HDF5

El formato interno de estos archivos se organiza mediante tres tipos de elementos:

- **Grupos**
  Son equivalentes a directorios en un sistema de archivos. Permiten organizar los datos almacenados de forma jerárquica, partiendo desde un grupo raíz “/”.
- **Datasets**
  Arrays multidimensionales de **tipo de dato fijo** que contienen la información (datos) almacenada, similares a los arrays de numpy.
- **Atributos**
  Pequeños fragmentos de información que describen las características o propiedades de los datos. Pueden estar asociados a cualquier grupo o dataset.