

Video32 (1)

Conceptos Principales

1. a
2. b
3. c

Notas

Resumen general

El como los **programas pueden trabajar directamente con archivos** para almacenar, comunicar o transferir información. Se enfoca en los **archivos de texto plano**, que son un **tipo especial de archivo binario de acceso secuencial**, donde cada carácter ocupa **1 byte**.

Usos de los archivos

1. **Almacenamiento de información:** guardar datos que luego se pueden leer o modificar.
2. **Comunicación entre programas:** un programa escribe un archivo y otro lo lee (como intermediario).
3. **Interacción humano-computadora:** por ejemplo, los archivos de configuración de Linux (`.git/config`).

Todos los archivos son binarios, pero los de texto son aquellos que pueden ser leídos por humanos con un editor.

Tipos de acceso a archivos

1. **Secuencial:** lectura desde el inicio hacia adelante, sin retroceso (ej. archivos de texto).
 2. **Directo:** acceso a posiciones específicas.
 3. **Indexado:** acceso mediante índices (mencionado brevemente).
-

Tipos de archivos de texto plano

1. Archivos de configuración tipo clave-valor

- Estructura:

```
nombre: Fran
apellido: Bernard
dirección: San José
```

- Cada campo tiene:
 - Un **nombre de llave**.
 - Un **valor**.
 - Un **separador** (: o =).
 - **Saltos de línea** que separan registros.
- Ventaja: legibles y simples.
- Desventaja: ocupan más espacio (por etiquetas y saltos de línea).

Ejemplo: archivos .ini o .conf.

2. Archivos CSV (Comma Separated Values)

- Guardan datos **por columnas y separados por comas o tabuladores**.

Ejemplo:

```
nombre,apellido,dirección
Fran,Bernard,San José
```

- Ventajas:
 - Más compactos que los de clave-valor.
 - Compatibles con hojas de cálculo.
- Desventajas:
 - No representan estructuras complejas.
 - Pérdida mínima de bytes por separadores y saltos de línea.

3. Registros de tamaño fijo

- Cada campo ocupa una **cantidad fija de bytes**.

Ejemplo:

- **nombre**: 6 bytes
- **apellido**: 6 bytes
- **dirección**: 20 bytes

- Ejemplo visual:

```
Fran__BernarSanJose_____
```

(_ representa espacios)

- Ventajas:
 - Ocupan menos espacio que los anteriores.
 - Permiten saltar directamente a un registro con `fseek` y `fread`.
- Desventajas:
 - **Desperdicio o recorte de datos** (si no caben en el tamaño fijo).

Se usan mucho en sistemas antiguos y bases de datos bancarias (AS/400).

Conclusiones

- Todos los archivos están formados por **unos y ceros**.
- Los de **texto plano** permiten que los humanos los lean fácilmente, pero:
 - **Desperdician espacio.**
 - Son útiles solo para **pequeñas cantidades de datos**.
- Para mayor eficiencia, se pasa a archivos **binarios o de registros fijos**.