

PROMETEO

PROGRAMACIÓN

Persistencia de datos

Problema: programas sin memoria

¿Qué ocurre cuando cerramos un programa?

Todo lo que estaba en:

- Variables
 - Listas
 - Maps
-  Se pierde

Idea clave

Si no guardamos datos, el programa empieza siempre desde cero.

¿Qué es la persistencia de datos?

Definición formal:

- Capacidad de almacenar datos de forma permanente para recuperarlos más adelante.

Definición práctica:

- Guardar información fuera del programa.

Donde se guarda:

- Disco duro
- SSD
- Archivos
- BBDD

Idea clave:

-  Sin persistencia → datos se pierden
-  Con persistencia → datos sobreviven

RAM vs Persistencia

Los programas trabajan en memoria, pero recuerdan gracias a la persistencia.

Datos en memoria

→ Características:

- ◆ Muy rápida
- ◆ Temporal
- ◆ Volátil

→ Se usa para:

- ◆ Variables
- ◆ Objetos
- ◆ Colecciones

Datos Persistentes

→ Características:

- ◆ Más lenta
- ◆ Permanente
- ◆ No volátil

→ Ejemplos:

- ◆ Archivos.txt
- ◆ Archivos.dat
- ◆ BBDD

Tipos de persistencia

Java ofrece varias opciones:

- 1.** Archivos de texto
- 2.** Archivos binarios
- 3.** Serialización de objetos
- 4.** Bases de datos

No todo sirve para lo mismo.

Persistencia en texto

¿Qué se guarda? → Texto plano

```
1;El Quijote;Cervantes;1605  
2;1984;Orwell;1949
```

¿Qué implica? → Elegir separador, mantener orden y convertir tipos.

¿Qué clase se usa? → BufferedReader / BufferedWriter

¿Qué problemas presenta? → No guarda objetos, hay que separar, parsear y validar.

Persistencia binaria

¿Qué se guarda? → Contenido en formato binario

¿Qué implica? → Serializar objetos.

¿Qué se puede serializar? → Objetos, Listas, Maps.

¿Qué clase se usa? → ObjectOutputStream / ObjectInputStream

¿Qué problemas presenta? → Contenido no editable, cambios rompen compatibilidad.

Conclusión

- Persistencia = memoria a largo plazo
- Sin persistencia no hay aplicación real
- Es la base de:
 - Logs
 - Bases de datos
 - Aplicaciones profesionales

Frase final

Un buen programa no solo funciona... **recuerda.**



PROMETEO