

Síntesis conceptual

Asignatura: Programación
Unidad: 12. Lectura y escritura de información

Resumen

Los **flujos de comunicación** permiten interactuar con elementos de entrada y/o salida para dar tratamiento de la información necesaria para nuestro programa. En java los flujos se denominan Stream. Los tipos de flujo se pueden **clasificar** según el tipo de datos, dirección o forma en la que se accede. Según el tipo de datos, nos podemos encontrar flujos de datos de bytes, caracteres o predeterminados.

Para utilizar los flujos se debe pasar por los estados de apertura, luego la transmisión de la información y finalmente el cierre. Las principales clases que disponemos para trabajar con flujo de datos de bytes son: FileOutputStream, FileInputStream, BufferedOutputStream, BufferedInputStream, DataOutputStream, DataInputStream, ByteArrayOutputStream, ByteArrayInputStream y PrintStream. Para flujo de datos de caracteres son: FileWriter, FileReader, BufferedWriter, BufferedReader, CharArrayWriter, CharArrayReader y PrintWriter.

Con los flujos de datos, podemos trabajar cualquier tipo primario para almacenarlo en bytes o caracteres. Pero también podemos guardar un objeto entero, para ello es necesario serializar la clase. La **serialización** permite transformar el objeto en un conjunto de bytes y tenerlo almacenado. Si posteriormente se quiere recuperar hay que deserializarlo.

Se puede hacer uso del **sistema de ficheros** del sistema operativo, usando la clase *File*. Podemos crear ficheros y directorios, como recorrer una ruta y conocer los permisos de lectura, escritura o ejecución.

Conceptos fundamentales

- **Flujo:** Secuencia de datos que se transmite por un elemento de entrada y se dirige a un elemento de salida.
- **Deserialización:** Transformación de una secuencia de bytes a objeto.
- **Buffer:** Contenedor de datos de forma temporal para ser leído en cualquier momento.
- **Path:** Ruta o dirección donde se encuentra un fichero o directorio.