



Ciclo 2

Semana 5

Persistencia en Archivos

Lectura 1 - Persistencia en archivos

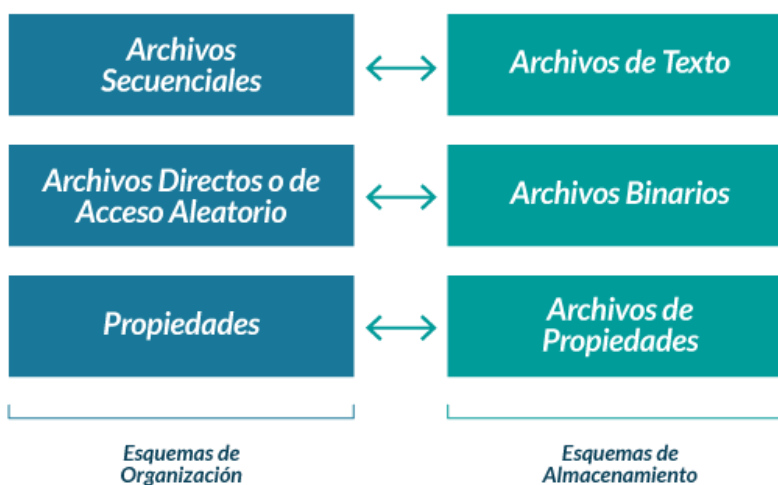


| Persistencia en archivos

Recordemos que la persistencia se define como la propiedad de un objeto por la que su existencia trasciende el tiempo. Es decir, aquella operación que se realiza para resguardar la información de un objeto permanentemente y también de recuperar su información para que esta sea reutilizada.



Los tipos de persistencia y los esquemas que encontramos en Java son:

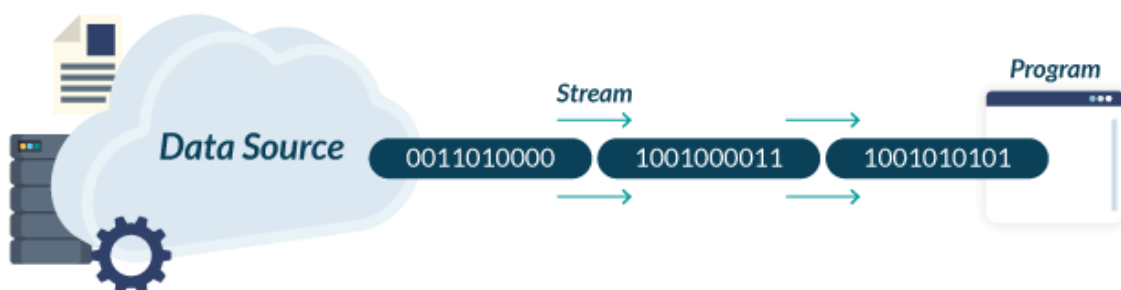


Normalmente, las aplicaciones necesitan leer o escribir información desde o hacia una fuente externa de datos. La información puede estar en cualquier parte, en un fichero, en disco, en algún lugar de la red, en memoria o en otro programa. También puede ser de cualquier tipo: objetos, caracteres, imágenes o sonidos.

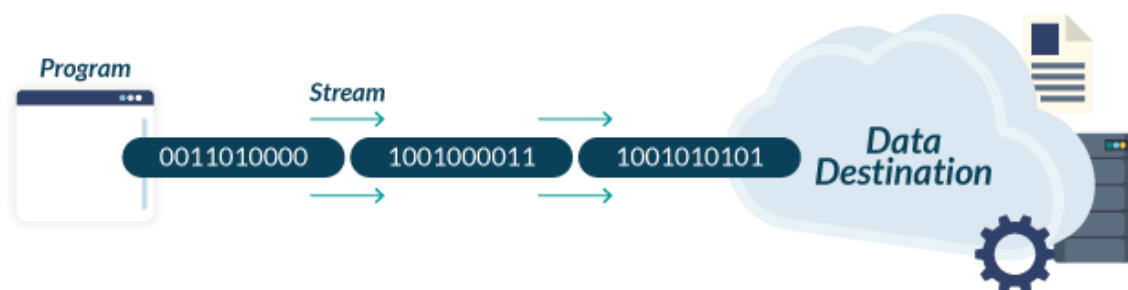


La comunicación entre el origen de cierta información y el destino se realiza mediante un stream (flujo) de información. Un stream es un objeto que hace de intermediario entre el programa y el origen o destino de la información.

Para traer la información, un programa abre un stream sobre una fuente de información (un fichero, memoria, un socket) y lee la información, de esta forma:



De igual forma, un programa puede enviar información a un destino externo abriendo un stream sobre un destino y escribiendo la información en este, de esta forma:



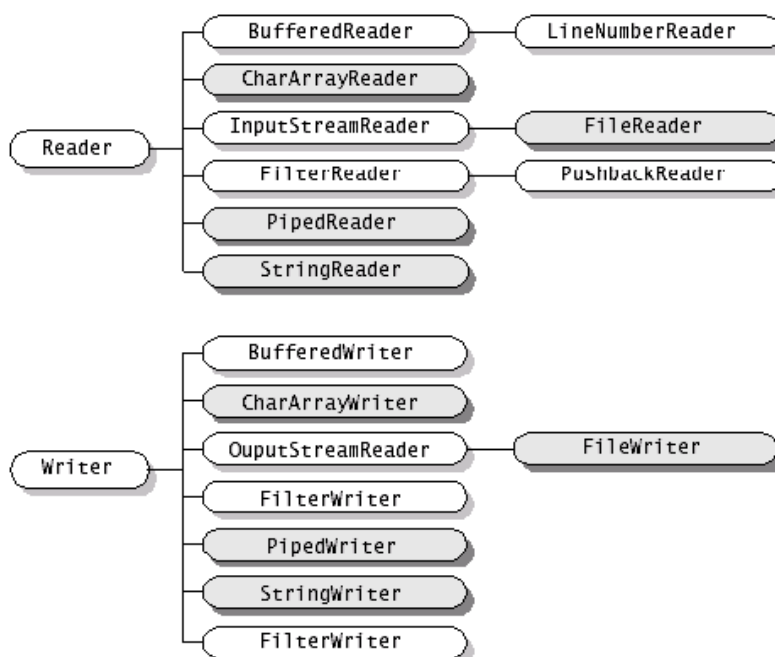
Un algoritmo general para las funcionalidades de leer y escribir es:

1. Abrir un stream
2. Mientras haya información
3. Leer o escribir información
4. Cerrar el stream



El paquete java.io contiene una colección de clases stream que soportan estos algoritmos para leer y escribir. Estas clases están divididas en dos árboles basándose en los tipos de datos (caracteres o bytes) sobre los que opera.

- **Streams de caracteres:** Reader y Writer son las superclases abstractas para streams de caracteres en java.io. Reader proporciona el API y una implementación para readers (streams que leen caracteres de 16-bits) y Writer proporciona el API y una implementación para writers (streams que escriben caracteres de 16-bits).



- Streams de Bytes:** los programas deberían usar los streams de bytes, descendientes de `InputStream` y `OutputStream`, para leer y escribir bytes de 8-bits. Estos streams se usan normalmente para leer y escribir datos binarios como imágenes y sonidos.

