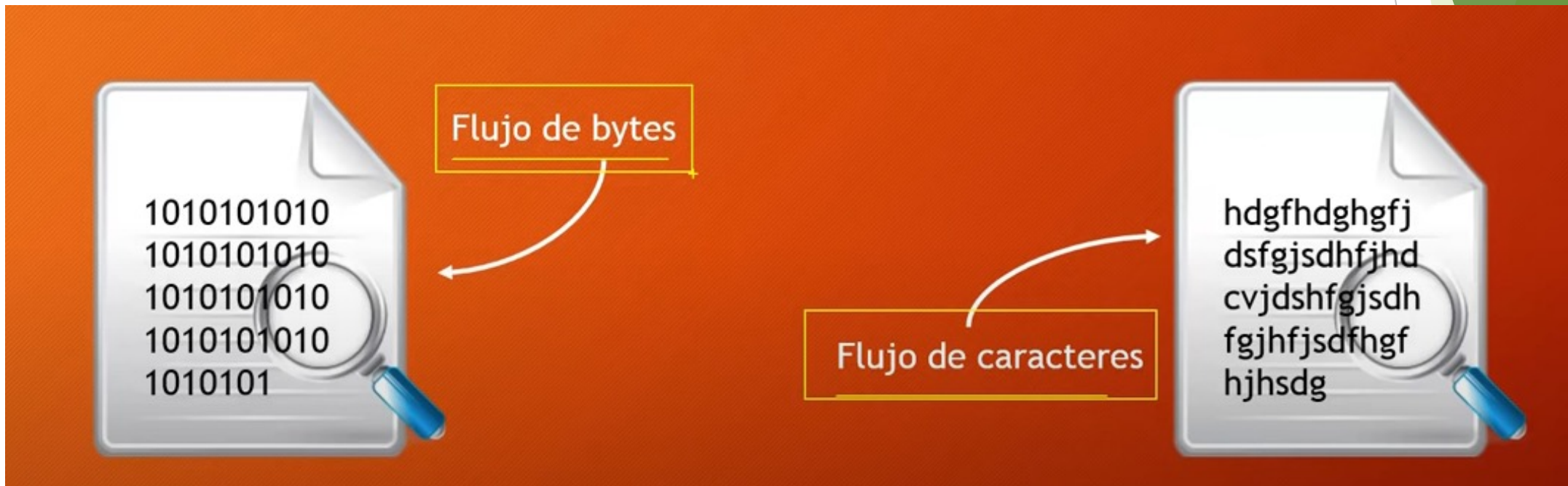


ACCESO A DATOS

UT 1 - MANEJO DE FICHEROS II

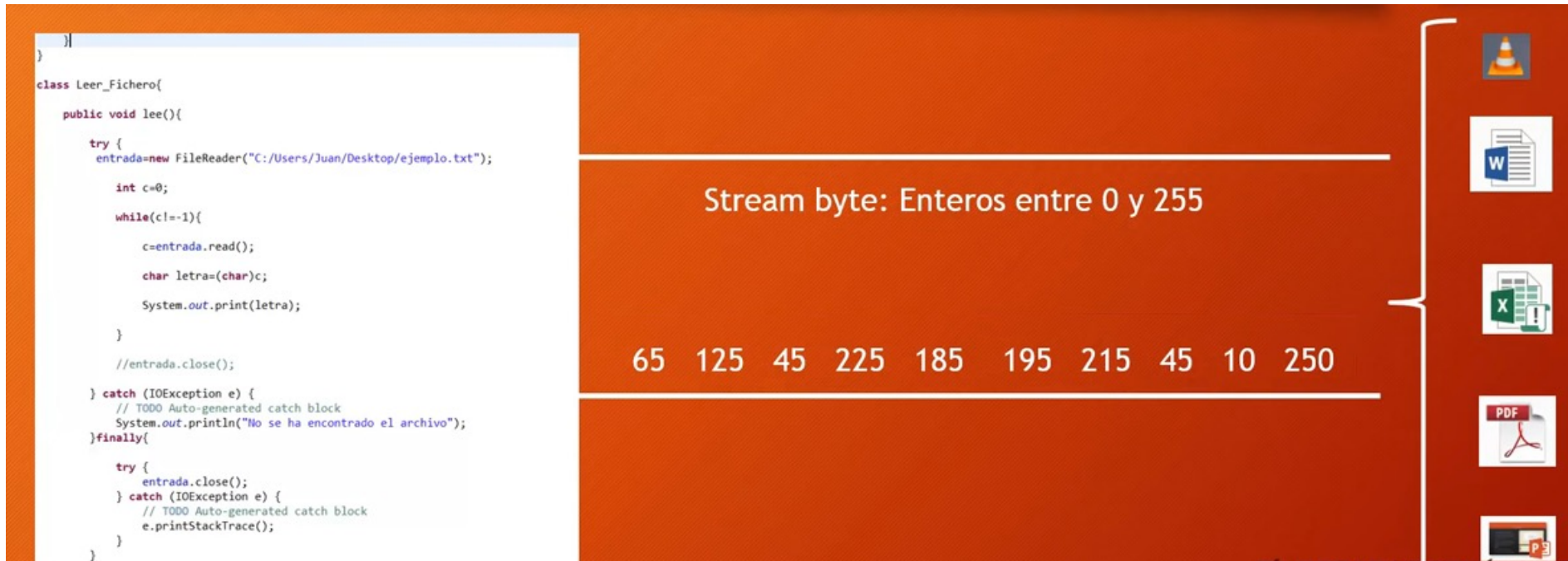
Formas de acceso a un fichero



Utilizamos el paquete java.io.*

Formas de acceso a un fichero

- Leyendo flujo de Bytes:
 - Permite enviar o recibir cualquier tipo de archivo (no solo de texto) desde o hacia nuestro programa Java



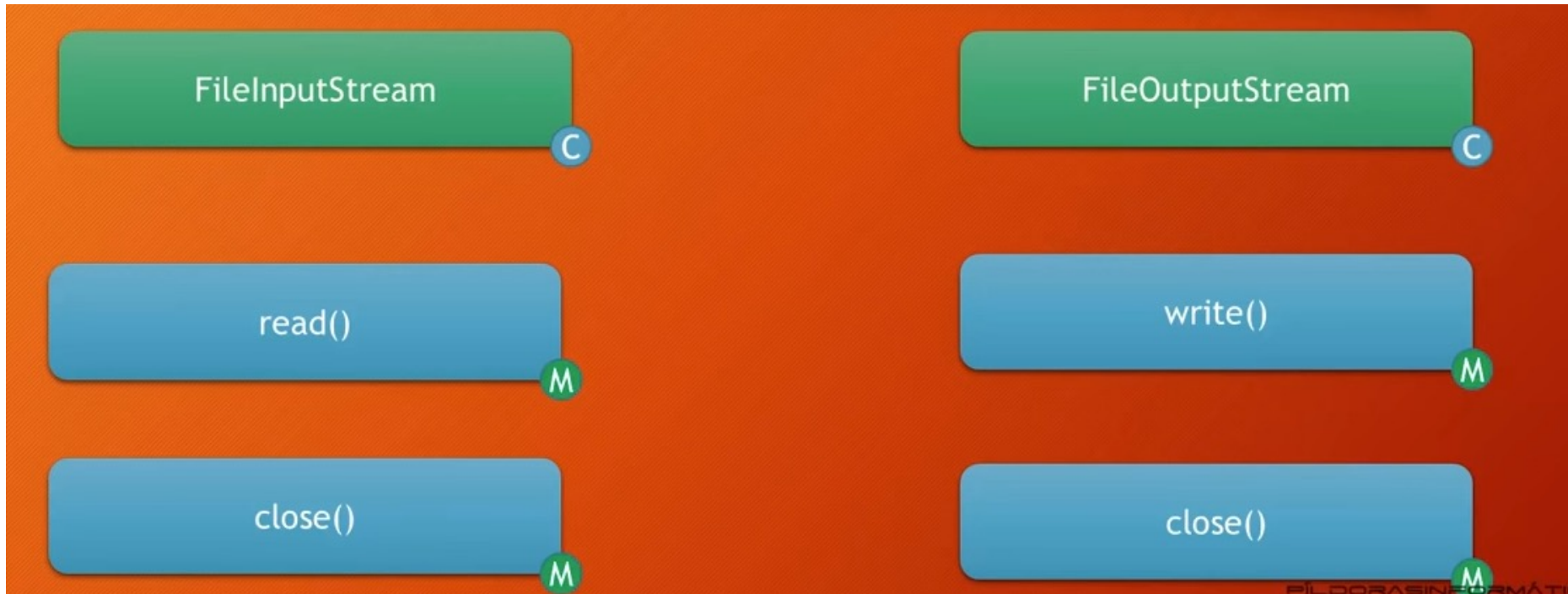
The image shows a Java IDE window on the left with the following code:

```
class Leer_Fichero{  
    public void lee(){  
        try {  
            entrada=new FileReader("C:/Users/Juan/Desktop/ejemplo.txt");  
            int c=0;  
            while(c!=-1){  
                c=entrada.read();  
                char letra=(char)c;  
                System.out.print(letra);  
            }  
            //entrada.close();  
        } catch (IOException e) {  
            // TODO Auto-generated catch block  
            System.out.println("No se ha encontrado el archivo");  
        }finally{  
            try {  
                entrada.close();  
            } catch (IOException e) {  
                // TODO Auto-generated catch block  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

On the right, a desktop environment is visible with a red background. It features a vertical stack of application icons: VLC media player, Microsoft Word, Microsoft Excel, Adobe PDF Reader, and a web browser. A large orange rectangle is overlaid on the desktop, containing the text "Stream byte: Enteros entre 0 y 255" and a sequence of numbers: 65 125 45 225 185 195 215 45 10 250.

Formas de acceso a un fichero

- Leyendo flujo de Bytes:
 - Visita la API y familiarízate con las clases y sus métodos



Formas de acceso a un fichero

► Leyendo flujo de bytes:

java.io

Class FileInputStream

java.lang.Object

java.io.InputStream

java.io.FileInputStream

All Implemented Interfaces:

Closeable, AutoCloseable

```
public FileInputStream(String name)  
    throws FileNotFoundException
```

Methods

Modifier and Type	Method and Description
int	available() Returns an estimate of the number of remaining bytes that can be read (or skipped over) from this input stream without blocking by the next invocation of a method for this input stream.
void	close() Closes this file input stream and releases any system resources associated with the stream.
protected void	finalize() Ensures that the <code>close</code> method of this file input stream is called when there are no more references to it.
FileChannel	getChannel() Returns the unique FileChannel object associated with this file input stream.
FileDescriptor	getFD() Returns the FileDescriptor object that represents the connection to the actual file in the file system being used by this FileInputStream .
int	read() Reads a byte of data from this input stream.
int	read(byte[] b) Reads up to <code>b.length</code> bytes of data from this input stream into an array of bytes.
int	read(byte[] b, int off, int len) Reads up to <code>len</code> bytes of data from this input stream into an array of bytes.
long	skip(long n) Skips over and discards <code>n</code> bytes of data from the input stream.

Constructors

Constructor and Description

FileInputStream(File file)

Creates a **FileInputStream** by opening a connection to an actual file, the file named by the **File** object `file` in the file system.

FileInputStream(FileDescriptor fdObj)

Creates a **FileInputStream** by using the file descriptor `fdObj`, which represents an existing connection to an actual file in the file system.

FileInputStream(String name)

Creates a **FileInputStream** by opening a connection to an actual file, the file named by the path name `name` in the file system.

Formas de acceso a un fichero

- ▶ ACTIVIDAD 1: Realiza el siguiente programa
 - ▶ Ajusta la ruta y la imagen

```
package accesoafichero;
import java.io.*;
public class LecturaEscrituraBytes {
    public static void main(String[] args) {
        try{
            boolean salida=false;
            int byteEntrada;
            int cantidadBytes=0;

            FileInputStream flujoArchivo = new FileInputStream("/Users/alejandrogonzalezredondo/Desktop/"
                + "EjemplosLecturaFicheros/EjemploBytes/kakashi.jpg");
            while (salida==false){
                byteEntrada=flujoArchivo.read();
                System.out.println(byteEntrada);
                cantidadBytes++;
                if (byteEntrada==-1)
                    salida=true;
            }
            flujoArchivo.close();
            System.out.println("El archivo tiene " + cantidadBytes + " bytes");
        }
    }
}
```

Formas de acceso a un fichero

- ▶ ACTIVIDAD 2: Crea una copia del programa anterior y modifícalo para que guarde los bytes en un array.

Formas de acceso a un fichero

► Escribiendo flujo de bytes:

```
public FileOutputStream(String name)  
    throws FileNotFoundException
```

java.io

Class **FileOutputStream**

java.lang.Object

java.io.OutputStream

java.io.**FileOutputStream**

All Implemented Interfaces:

Closes the file and releases any system resources associated with the file.

Methods

Modifier and Type	Method and Description
void	close() Closes this file output stream and releases any system resources associated with this stream.
protected void	finalize() Cleans up the connection to the file, and ensures that the <code>close</code> method of this file output stream is called when there are no more references to this stream.
FileChannel	getChannel() Returns the unique FileChannel object associated with this file output stream.
FileDescriptor	getFD() Returns the file descriptor associated with this stream.
void	write(byte[] b) Writes <code>b.length</code> bytes from the specified byte array to this file output stream.
void	write(byte[] b, int off, int len) Writes <code>len</code> bytes from the specified byte array starting at offset <code>off</code> to this file output stream.
void	write(int b) Writes the specified byte to this file output stream.

Constructors

Constructor and Description

FileOutputStream(File file)

Creates a file output stream to write to the file represented by the specified File object.

FileOutputStream(File file, boolean append)

Creates a file output stream to write to the file represented by the specified File object.

FileOutputStream(FileDescriptor fdObj)

Creates a file output stream to write to the specified file descriptor, which represents an existing connection to an actual file in the file system.

FileOutputStream(String name)

Creates a file output stream to write to the file with the specified name.

FileOutputStream(String name, boolean append)

Creates a file output stream to write to the file with the specified name.

Formas de acceso a un fichero

- ▶ ACTIVIDAD 3: Modifica el programa anterior para que sea capaz de crear un fichero exactamente igual que el anterior en la misma carpeta y con el mismo nombre_copia.