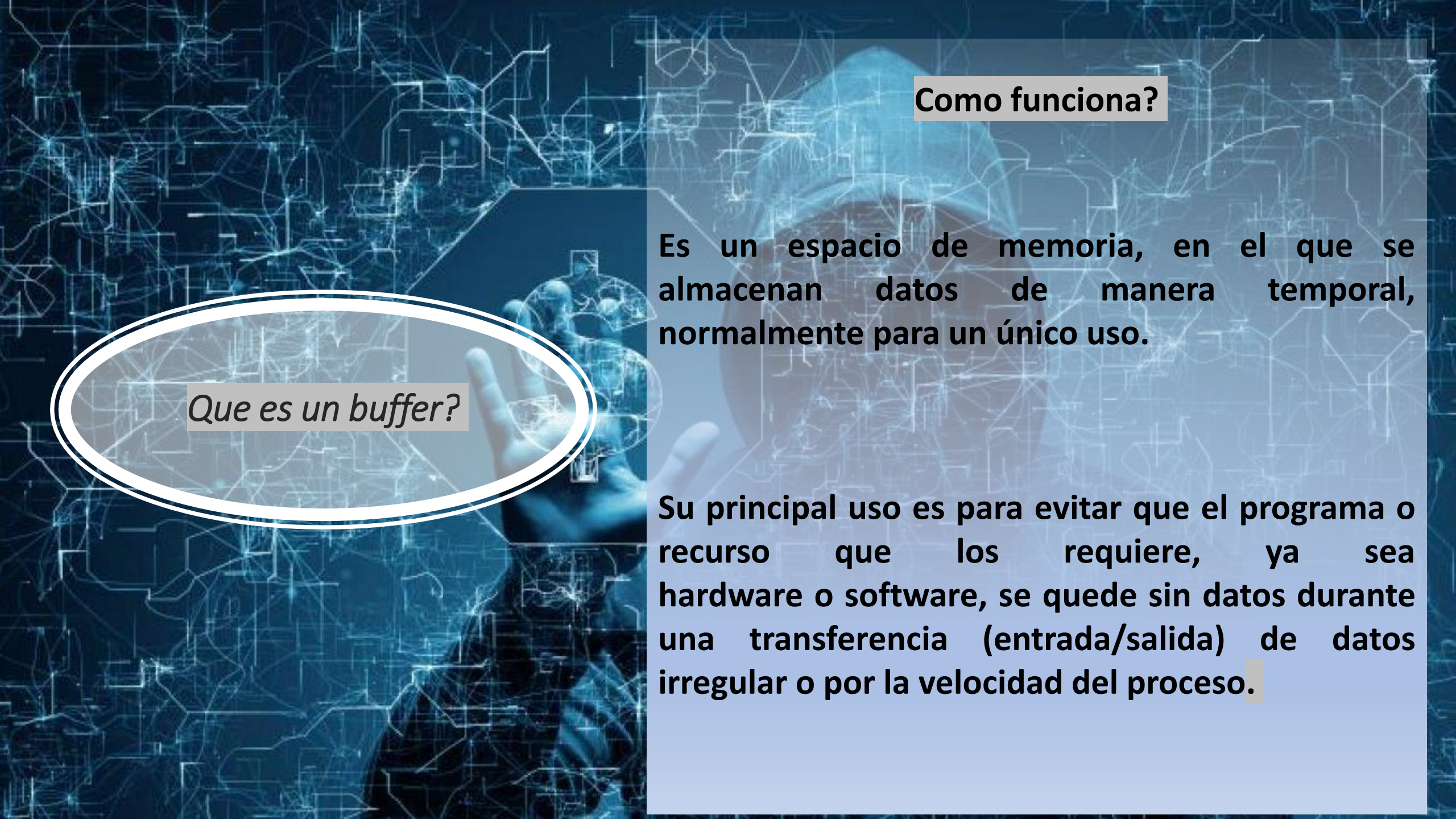
Introducción

Monitoria

Frase Motivadora

Monitoria


Que es un buffer?



Que es un buffer?

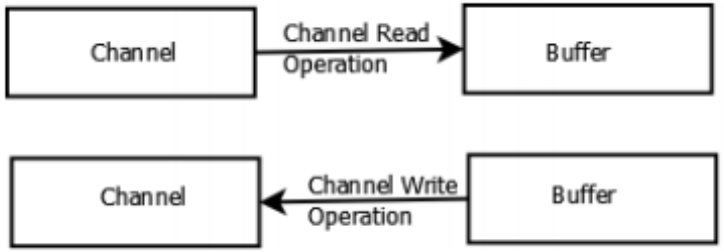
Como funciona?

Es un espacio de memoria, en el que se almacenan datos de manera temporal, normalmente para un único uso.

Su principal uso es para evitar que el programa o recurso que los requiere, ya sea hardware o software, se quede sin datos durante una transferencia (entrada/salida) de datos irregular o por la velocidad del proceso.

Que es un buffer?

Como funciona?



The table for channels is as follows:

Channel class	Purpose
`FileChannel`	This connects to a file
`DatagramChannel`	This supports datagram sockets
`SocketChannel`	This supports streaming sockets
`ServerSocketChannel`	This listens for socket requests
`NetworkChannel`	This supports a network socket
`AsynchronousSocketChannel`	This supports asynchronous streaming sockets

The table for buffers is as follows:

Buffer class	Data type supported
`ByteBuffer`	`byte`
`CharBuffer`	`char`
`DoubleBuffer`	`double`
`FloatBuffer`	`float`
`IntBuffer`	`int`
`LongBuffer`	`long`
`ShortBuffer`	`short`

BufferedReader
e InputStreamReader

Declaracion

```
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
```

Como funciona?

BufferedReader tiene la misma funcion del FileReader (es decir leer y escribir en ficheros) , pero estas funciones son mas optimas. Su uso tambien puede variar. Algunas personas los utilizan para lectura por consola.

BufferedWriter
y
OutputStreamReader

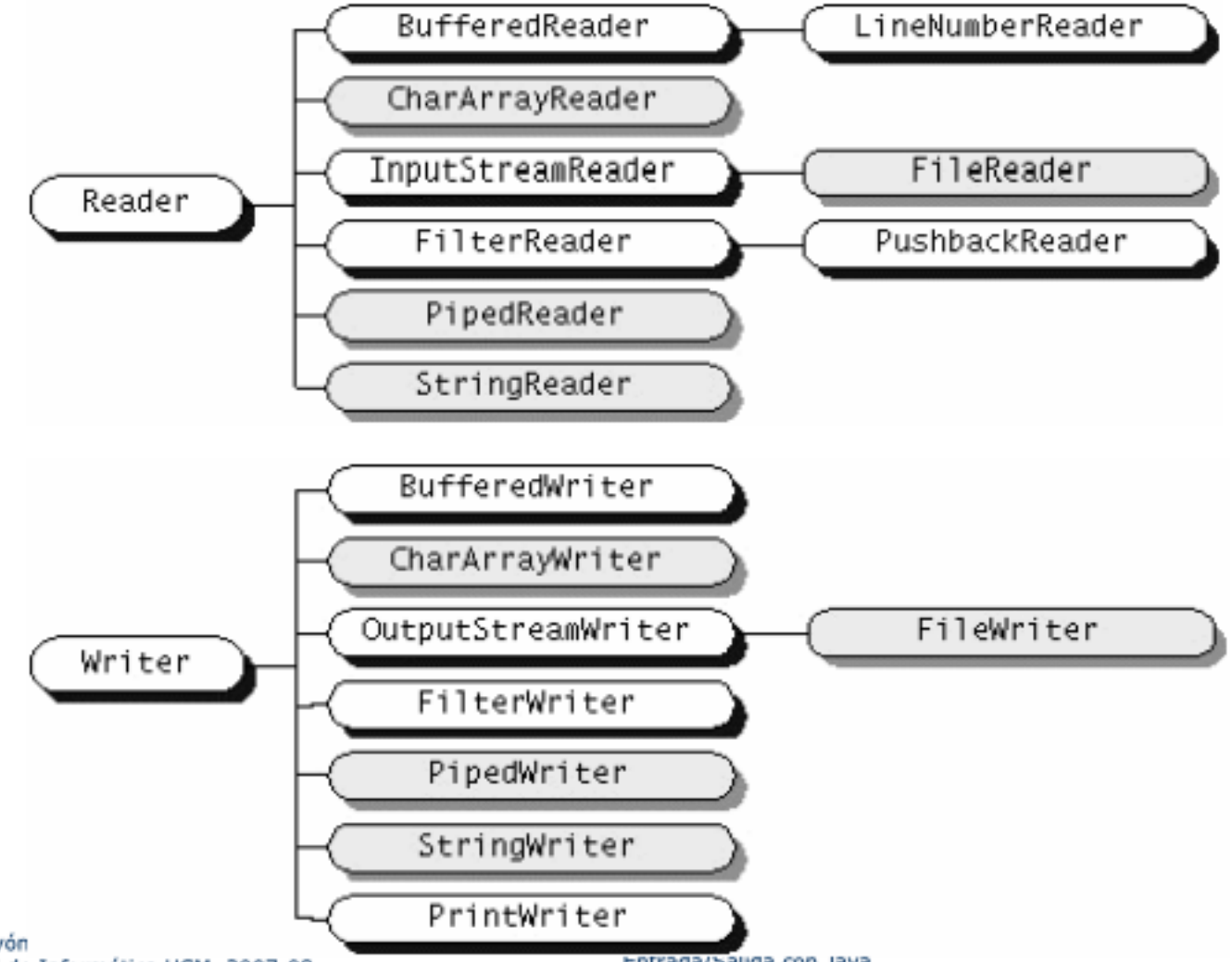
Declaracion

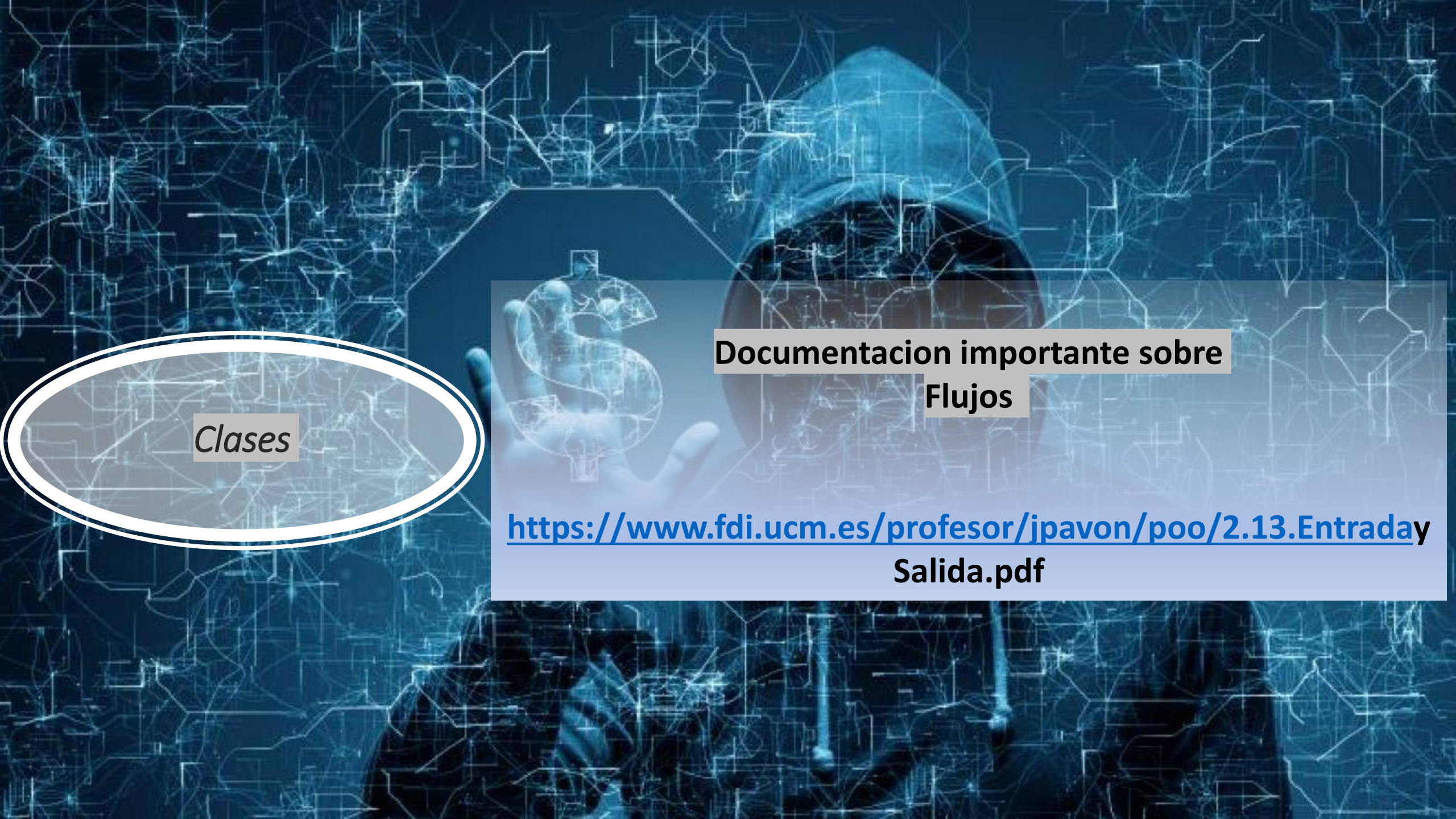
```
BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(System.out));
```

Como funciona?

La clase implementa un flujo de salida en búfer. Al configurar tal flujo de salida, una aplicación puede escribir bytes en el flujo de salida subyacente sin necesariamente generar una llamada al sistema subyacente para cada byte escrito.

Clases


Reader
BufferedReader
LineNumberReader
CharArrayReader
InputStreamReader
FileReader
FilterReader
PushbackReader
PipedReader
StringReader
Writer
BufferedWriter
CharArrayWriter
OutputStreamWriter
FileWriter
FilterWriter
PipedWriter
StringWriter
PrintWriter



Documentacion importante sobre Flujos

https://www.fdi.ucm.es/profesor/jpavon/poo/2.13.EntradaySalida.pdf

BufferedWriter y BufferedReader

Ejemplo, (lectura y escritura por consola)

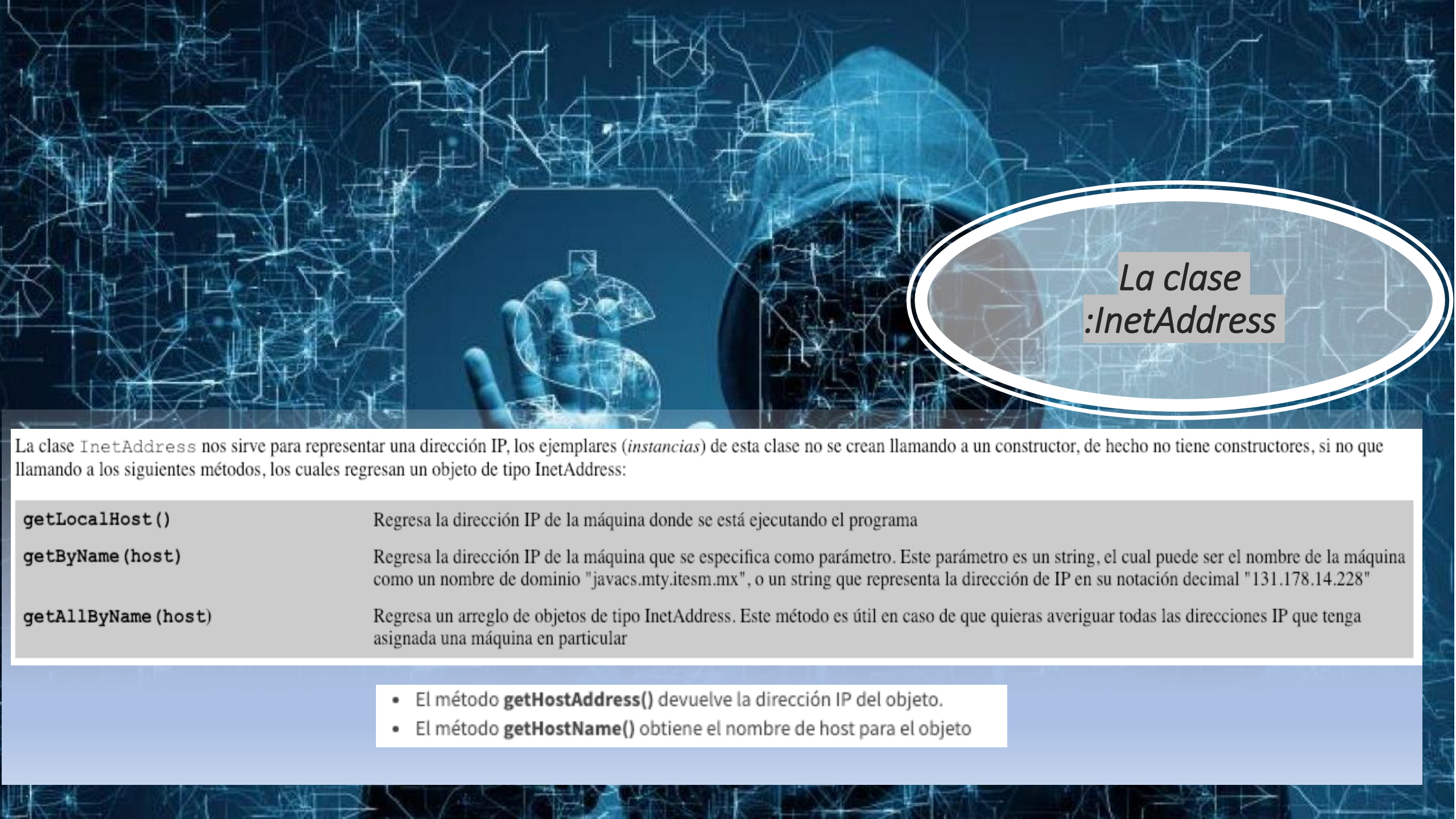
```
public static void main(String[] args) {

     BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
     BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(System.out));


     try {
        String[] cadena = br.readLine().split(";");
        int suma = sumaNumeros(Integer.parseInt(cadena[0]), Integer.parseInt(cadena[1]))
        bw.write(suma + "");
        bw.flush();
        br.close();
        bw.close();

     }

     catch (IOException e) {
        System.out.println("Se presento un error en la lectura de consola");
     }
}
```



La clase :InetAddress

La clase `InetAddress` nos sirve para representar una dirección IP, los ejemplares (*instancias*) de esta clase no se crean llamando a un constructor, de hecho no tiene constructores, si no que llamando a los siguientes métodos, los cuales regresan un objeto de tipo InetAddress:

`getLocalHost()`	Regresa la dirección IP de la máquina donde se está ejecutando el programa
`getByName(host)`	Regresa la dirección IP de la máquina que se especifica como parámetro. Este parámetro es un string, el cual puede ser el nombre de la máquina como un nombre de dominio "javacs.mty.itesm.mx", o un string que representa la dirección de IP en su notación decimal "131.178.14.228"
`getAllByName(host)`	Regresa un arreglo de objetos de tipo InetAddress. Este método es útil en caso de que quieras averiguar todas las direcciones IP que tenga asignada una máquina en particular

- El método **getHostAddress()** devuelve la dirección IP del objeto.
- El método **getHostName()** obtiene el nombre de host para el objeto

La clase :InetAddress

https://docs.oracle.com/javase/7/docs/api/java/net/InetAddress.html