

Actividad Comentar Ejemplos Tema 3.

- **Ejemplo IP:**

Código:

```
import java.net.*;
import java.lang.*;

public class IP {

    public static void main(String[] args) throws NullPointerException {

        InetAddress dirIP = null;

        String cadena1, cadena2, cadena3;

        cadena1 = cadena2 = cadena3 = "";

        try {

            dirIP = InetAddress.getLocalHost();

            //getByName()= devuelve el nombre de la máquina y la IP.

            dirIP = InetAddress.getByName("www.madrid.org");

            //getLocalHost()=Nos devuelve el nombre y la IP del equipo.

            System.out.println("Mi IP: " + dirIP.getLocalHost());

            //InetAddress.getByName()=devuelve el nombre de la máquina y la IP

            System.out.println("IP de Google " + InetAddress.getByName("www.google.es"));

            //Devuelve la IP como una cadena.

            cadena1 = dirIP.getHostAddress();

            //getHostName()=devuelve el nombre del host.

            cadena2 = dirIP.getHostName();

            //getCanonicalHostName()=devuelve el nombre canónico completo de la máquina.

            cadena3 = dirIP.getCanonicalHostName();

            System.out.println("Ip:" + cadena1);

            System.out.println("nombre:" + cadena2);

            System.out.println("nombre canonico: " + cadena3);
```

```

//InetAddress.getByNome()=devuelve el nombre de la máquina y la IP
dirIP = InetAddress.getByNome("www.hotmail.es");

//InetAddress.getAllByName()=devuelve un array de objetos,son las distintas IPs que
tiene la máquina indicada

InetAddress[] direcciones = InetAddress.getAllByName(dirIP.getHostName());

int i;

cadena1 = "";

for (i = 0; i < direcciones.length; i++){

    cadena1 = direcciones[i].toString();

    System.out.println("\t\t" + cadena1);

}

} catch (UnknownHostException e) {

    System.out.println("ERROR CON LAS IP");

} catch (NullPointerException e) {

    System.out.println("error de cadenas ");

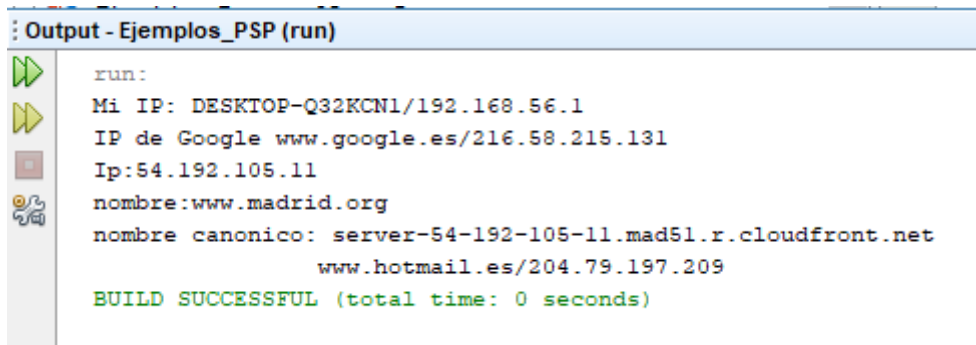
}

}

}

```

- Salida por pantalla.



```

run:
Mi IP: DESKTOP-Q32KCN1/192.168.56.1
IP de Google www.google.es/216.58.215.131
Ip:54.192.105.11
nombre:www.madrid.org
nombre canonico: server-54-192-105-11.mad51.r.cloudfront.net
www.hotmail.es/204.79.197.209
BUILD SUCCESSFUL (total time: 0 seconds)

```

- **Ejemplo URL 1:**

Código:

```
import java.net.*;
```

```
public class EjemploURL {
```

```
    static void metodos(URL direWeb) {
```

```
        //toString()=coge el String del direWeb
```

```
        System.out.println("URL COMPLETA " + direWeb.toString());
```

```
        //getProtocol()=,nos da el protocolo de la URL puede ser los protocolos=(http, https,...).
```

```
        System.out.println("protocolo " + direWeb.getProtocol());
```

```
        //getHost()=nos da la dirección de la página.
```

```
        System.out.println("HOST " + direWeb.getHost());
```

```
        //getPort()=nos da devuelve el puerto utilizado, si lo indicas en la URL. Devolverá -  
1 si el puerto no viene especificado
```

```
        System.out.println("PUERTO " + direWeb.getPort());
```

```
        //getFile()=nos da el fichero a consultar con su ruta si la hemos puesto
```

```
        System.out.println("fichero " + direWeb.getFile());
```

```
        //getUserInfo()=devuelve el usuario o la palabra null sino existe usuario.
```

```
        System.out.println("USUARIO " + direWeb.getUserInfo());
```

```
        //getPath()=nos devuelve la ruta
```

```
        System.out.println("RUTA " + direWeb.getPath());
```

```
        //getAuthority()=suele dar la dirección de la página.
```

```
        System.out.println("AUTORIZADO " + direWeb.getAuthority());
```

```
        //getQuery()=devuelve null si no existe consulta.
```

```
        System.out.println("CONSULTA " + direWeb.getQuery());
```

```
    }
```

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

```
        try {
```

```
            URL url = new URL("https", "bankia.es", "inicio.html");
```

```
            //url = new URL("http://www.terra.es");
```

```
            metodos(url);
```

```
        } catch (MalformedURLException e) {
```

```
            System.out.println("ERROR");
```

```
        }
```

```
    }
```

```
}
```

```

run:
URL COMPLETA https://bankia.esinicio.html
protocolo https
HOST bankia.es
PUERTO -1
fichero inicio.html
USUARIO null
RUTA inicio.html
AUTORIZADO bankia.es
CONSULTA null
BUILD SUCCESSFUL (total time: 0 seconds)

```

- Ejemplo URL 2:

Código:

```

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;
/**
 *
 * @author david
 */
public class EjemploURL2 {
    public static void main(String[] args) {
        URL url = null;
        //Añades aqui la url que quieras
        try {
            url = new URL("https://www.google.com/");
        } catch (MalformedURLException e) {
            System.out.println("ERROR");
        }
        BufferedReader in;
        try {
            //Le pasas todo el "esqueleto"/el html de la pagina que pongas en el url
            InputStream inputStream = url.openStream();
            //Se va añadiendo aqui mediante el BufferedReader
            in = new BufferedReader(new InputStreamReader(inputStream));
            String inputLine;
            //Aqui con este bucle se para hasta que llegue al final de pagina en plan el html
            while ((inputLine = in.readLine()) != null) {
                System.out.println(inputLine);
            }
            //Cierro el Buffered Reader y luego ya la excepción

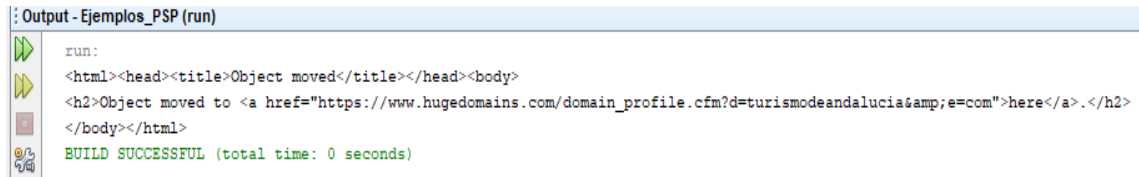
```

```

        in.close();
    } catch (IOException e) {
        System.out.println("ERROR 2");
    }
}
}

```

- Por lo consola sale lo siguiente:



```

Output - Ejemplos_PSP (run)
run:
<html><head><title>Object moved</title></head><body>
<h2>Object moved to <a href="https://www.hugedomains.com/domain_profile.cfm?d=turismodeandalucia&e=com">here</a>.</h2>
</body></html>
BUILD SUCCESSFUL (total time: 0 seconds)

```

- **Ejemplo Servidor-Cliente.**

Código EjemploServidor:

```

import java.io.*;
import java.net.*;

public class EjemploServidor {
    public static void main(String[] args) throws IOException {
        //Declarar un puerto
        int puerto = 6000;
        //ServerSocket=crea un socket e indicamos el puerto por el que va a actuar
        ServerSocket servidor = new ServerSocket(puerto);
        //Socket=Es la clase para el cliente
        //En este caso crea un socket sin ningún puerto asociado
        Socket cliente = null;
        System.out.println("Esperando conexión: ");
        /*
        accept()=Escucha una solicitud de conexión de un cliente y la acepta.
        Una vez establecida la conexión devuelve un objeto tipo Socket por el que se
        creara la conexión. El ServerSocket sigue disponible para aceptar nuevas
        conexiones
        */
        cliente = servidor.accept();
        //Flujo de entrada del cliente
        InputStream entrada = null;
        //cliente.getInputStream()=permite leer bytes desde el socket
        entrada = cliente.getInputStream();
        DataInputStream flujoentrada = new DataInputStream(entrada);
        System.out.println("Recibiendo del cliente: " + flujoentrada.readUTF());
        // Flujo de salida del cliente
        OutputStream salida = null;
        //cliente.getOutputStream()=permite escribir bytes sobre el socket
        salida = cliente.getOutputStream();
    }
}

```

```

        DataOutputStream flujosalida = new DataOutputStream(salida);
        // Envio datos al cliente
        flujosalida.writeUTF("saludos al cliente");
        //Cerrar canales y socket
        entrada.close();
        flujoentrada.close();
        salida.close();
        flujosalida.close();
        cliente.close();
        servidor.close();
    }
}

```

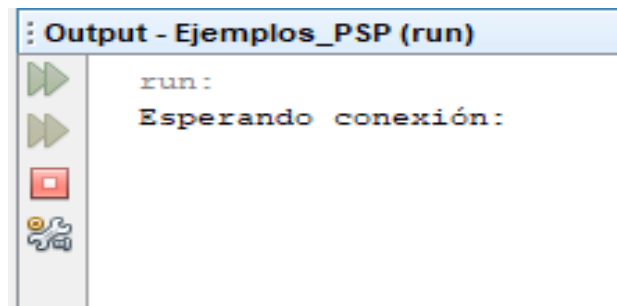
Código EjemploCliente:

```

public class EjemploCliente {
    public static void main(String[] args) throws IOException {
        //Le pongo localhost porque es la forma de comunicacion
        String host = "localhost";
        //Creo el puerto
        int puerto = 6000;
        System.out.println("Iniciando conexion: ");
        //Socket=Es la clase para el cliente
        //Crea un socket al puerto indicado y al nombre del host indicado
        Socket cliente = new Socket(host, puerto);
        //Flujo de salida al servidor
        //cliente.getOutputStream()=permite escribir bytes sobre el socket
        DataOutputStream flujosalida = new
DataOutputStream(cliente.getOutputStream());
        flujosalida.writeUTF("SALUDOS al servidor ");
        // Flujo de entrada
        //cliente.getInputStream()=Permite leer bytes desde el socket
        DataInputStream flujoentrada = new DataInputStream(cliente.getInputStream());
        System.out.println("recibiendo del servidor:" + flujoentrada.readUTF());
        //Cerrar canales y socket
        flujoentrada.close();
        flujosalida.close();
        cliente.close();
    }
}

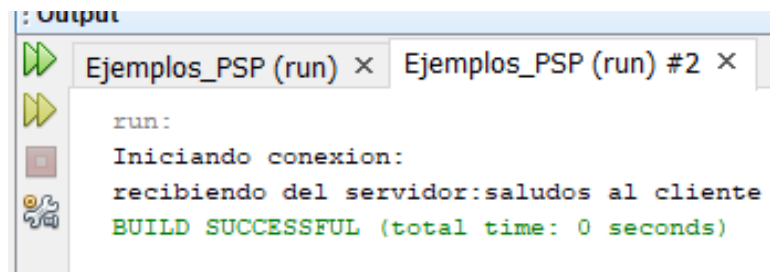
```

- 1º Ejecutas la clase EjemploServidor y te saldrá esto:



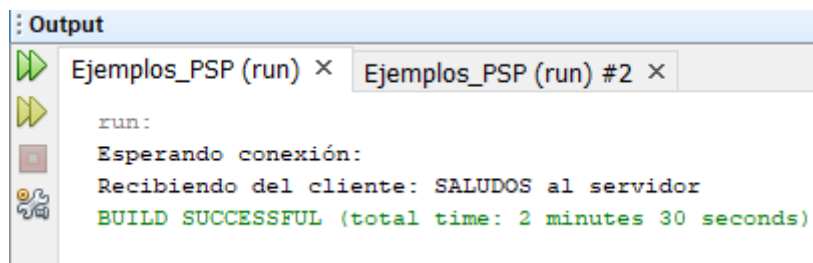
```
run:
Esperando conexión:
```

- 2º Ejecutas la clase EjemploCliente y te sale esto:



```
run:
Iniciando conexion:
recibiendo del servidor:saludos al cliente
BUILD SUCCESSFUL (total time: 0 seconds)
```

- 3º Te pones en la clase EjemploServidor y aparece esto:



```
run:
Esperando conexión:
Recibiendo del cliente: SALUDOS al servidor
BUILD SUCCESSFUL (total time: 2 minutes 30 seconds)
```