**ACTIVIDAD 1**

**Enunciado**

Realizar un programa que obtenga todas las direcciones IP asociadas con el dominio [www.spotify.com](http://www.spotify.com/), utilizando la clase Java InetAddress.

El nombre de la clase principal será Actividad1.

El resultado esperado será equivalente al siguiente, teniendo en cuenta que las direcciones IP pueden variar:

Direcciones asociadas a Spotify:

www.spotify.com/194.132.198.118

www.spotify.com/194.132.198.149

[www.spotify.com/194.132.197.198](http://www.spotify.com/194.132.197.198)

**Solución**

**// Actividad1.java**

//Importamos la clase InetAddress del paquete java.net

import java.net.InetAddress;

import java.net.UnknownHostException;

//Definimos la clase principal Actividad1

public class Actividad1 {

// Método principal del programa

public static void main(String[] args) {

try {

// Obtenemos todas las direcciones IP asociadas al dominio "www.spotify.com"

InetAddress[] addresses = InetAddress.getAllByName("www.spotify.com");

// Imprimimos las direcciones obtenidas

System.out.println("Direcciones asociadas a Spotify:");

for (InetAddress address : addresses) {

System.out.println(address);

}

} catch (UnknownHostException e) {

// Si no se puede encontrar el dominio, se imprime un mensaje de error

System.out.println("No se pudo encontrar el dominio especificado.");

}

}

}

**Resolución de problemas**

**No me han surgido problemas a la hora de realizar el ejercicio.**

**ACTIVIDAD 2**

**Enunciado**

Realizar un programa que pase como parámetro una dirección y obtenga todas las direcciones asociadas a ese parámetro, utilizando la clase Java InetAddress.

El nombre de la clase principal será Actividad2.

El resultado esperado cuando se haya pasado como parámetro una dirección existente será equivalente al siguiente, teniendo en cuenta que las direcciones IP pueden variar:

Las direcciones asociadas a web.gencat.cat son:

web.gencat.cat/84.53.132.41

web.gencat.cat/84.53.132.18

El resultado esperado cuando no se haya pasado ningún parámetro es el siguiente:

Se necesita una URL para obtener su dirección

**Solución**

**// Actividad2.java**

//Importamos la clase InetAddress del paquete java.net

import java.net.InetAddress;

import java.net.UnknownHostException;

//Definimos la clase principal Actividad2

public class Actividad2 {

// Método principal del programa

public static void main(String[] args) {

// Comprobamos si se ha pasado algún parámetro en la línea de comandos

if (args.length == 0) {

// Si no se ha pasado ningún parámetro, se imprime un mensaje de error

System.out.println("Se necesita una URL para obtener su dirección");

} else {

// Si se ha pasado algún parámetro, se toma el primero como el dominio a

// consultar

String domain = args[0];

try {

// Obtenemos todas las direcciones IP asociadas al dominio proporcionado

InetAddress[] addresses = InetAddress.getAllByName(domain);

// Imprimimos las direcciones obtenidas

System.out.println("Las direcciones asociadas a " + domain + " son:");

for (InetAddress address : addresses) {

System.out.println(address);

}

} catch (UnknownHostException e) {

// Si no se puede encontrar el dominio, se imprime un mensaje de error

System.out.println("No se pudo encontrar el dominio especificado.");

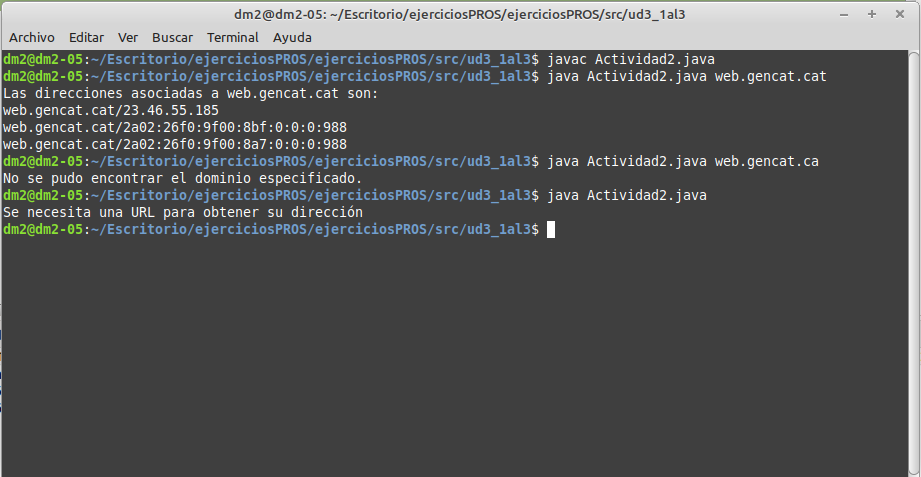
}

}

}

}

**Pantallazo de la ejecucion del archivo pasandole una web que exista, una web inexistente y sin pasarle nada:**

**Resolución de problemas**

**No me han surgido problemas a la hora de realizar el ejercicio.**

**ACTIVIDAD 3**

**Enunciado**

Realiza un programa que reciba como parámetro un nombre de dominio y te devuelva su dirección IP. Si no se le pasa ningún parámetro válido, devolverá las direcciones IP locales.

La clase Java a utilizar es InetAddress. El nombre de la clase principal será Actividad3.

El resultado esperado cuando no se pase ningún parámetro, será equivalente al siguiente:

Dirección IP:172.20.105.120

Nombre:DM2-PROF

DM2-PROF/172.20.105.120

172.20.105.120.

DM2-PROF/172.20.105.120

DM2-PROF/192.168.56.1

DM2-PROF/192.168.191.1

DM2-PROF/192.168.150.1

DM2-PROF/fe80:0:0:0:80f7:f52e:1842:bb04%9

DM2-PROF/fe80:0:0:0:680c:9d5e:2450:df7f%5

DM2-PROF/fe80:0:0:0:71f6:e794:453c:5895%12

DM2-PROF/fe80:0:0:0:64cb:7ced:4f85:4e81%4

El resultado esperado cuando se pase como parámetro una dirección existente será el siguiente:

Dirección IP:204.93.177.191

Nombre:www.ciudadjardin.com

www.ciudadjardin.com/204.93.177.191

204.93.177.191.

www.ciudadjardin.com/204.93.177.191

**Solución**

**// Actividad3.java**

import java.net.InetAddress;

import java.net.UnknownHostException;

public class Actividad3 {

public static void main(String[] args) {

try {

// Si no se pasan parámetros, se muestran las direcciones IP locales

if (args.length == 0) {

InetAddress localHost = InetAddress.getLocalHost();

System.out.println("Dirección IP: " + localHost.getHostAddress());

System.out.println("Nombre: " + localHost.getHostName());

InetAddress[] localAddresses = InetAddress.getAllByName(localHost.getHostName());

for (InetAddress address : localAddresses) {

System.out.println(address);

}

} else {

// Si se pasan parámetros, se muestra la dirección IP del dominio especificado

String domain = args[0];

InetAddress[] addresses = InetAddress.getAllByName(domain);

System.out.println("Dirección IP: " + addresses[0].getHostAddress());

System.out.println("Nombre: " + addresses[0].getHostName());

for (InetAddress address : addresses) {

System.out.println(address);

}

}

} catch (UnknownHostException e) {

// Si no se puede encontrar el dominio o la dirección local, se imprime un

// mensaje de error

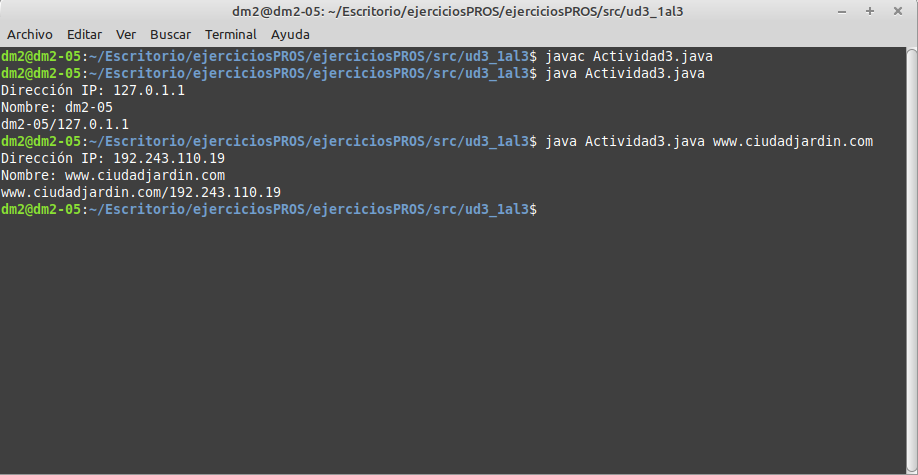
System.out.println("No se pudo encontrar el dominio o la dirección local especificada.");

}

}

}

**Adjunto un pantallazo probando las distintas opciones que nos da este programa (sacar la ip de una url/dominio especificado o las direcciones ip locales si no le proporcionamos ninguna url/dominio):**

****

**Resolución de problemas**

**No me han surgido problemas a la hora de realizar el ejercicio.**

**CRITERIOS DE CORRECCIÓN**

1. Nivel de ajuste a la nomenclatura de objetos definida en el enunciado de la actividad y nivel de ajuste al almacenamiento de la información requerida en las subcarpetas definidas para la actividad y unidad didáctica correspondientes (15%).
2. Hacer lo que se indica en el enunciado (55%).
3. Claridad del código Java (15%).
4. Documentación del código Java (15%).
5. Explicación detallada de cómo se han resuelto problemas que hayan podido surgir en la realización de la actividad (hasta un 15% adicional sobre la nota obtenida).   
   Dicha explicación se incluirá al principio de los archivos de código fuente Java mediante líneas de comentarios.   
   La primera línea de comentarios sólo incluirá lo siguiente:  
    // Resolución de problemas.
6. Variables no usadas (-15%).
7. Librerías no usadas (-15%).