**ACTIVIDAD 4**

Enunciado

Realizar un programa con un método de nombre Visualizar(), que tenga un parámetro que sea un objeto URL y que visualice el resultado de aplicar diferentes métodos de la clase URL a un objeto URL creado con diferentes constructores. Se usarán las siguientes URLs:

* http://docs.oracle.com
* http://docs.oracle.com/javase/7
* http://docs.oracle.com/javase/7/docs/api/java/net/URL.html

El nombre de la clase principal será Actividad4.

El resultado esperado es el siguiente:

Constructor simple para una URL:

Método toString():http://docs.oracle.com

Método getProtocol():http

Método getHost():docs.oracle.com

Método getPort():-1

Método getFile():

Método getUserInfo():null

Método getPath():

Método getAuthority():docs.oracle.com

Método getQuery():null

Constructor para protocolo, host y directorio:

Método toString():http://docs.oracle.com/javase/7

Método getProtocol():http

Método getHost():docs.oracle.com

Método getPort():-1

Método getFile():/javase/7

Método getUserInfo():null

Método getPath():/javase/7

Método getAuthority():docs.oracle.com

Método getQuery():null

Constructor para protocolo, host, puerto y directorio:

Método toString():http://docs.oracle.com:80/javase/7

Método getProtocol():http

Método getHost():docs.oracle.com

Método getPort():80

Método getFile():/javase/7

Método getUserInfo():null

Método getPath():/javase/7

Método getAuthority():docs.oracle.com:80

Método getQuery():null

Constructor para un objeto URL y su directorio:

Método toString():http://www.oracle.com/javase/7/docs/api/java/net/URL.html

Método getProtocol():http

Método getHost():www.oracle.com

Método getPort():-1

Método getFile():/javase/7/docs/api/java/net/URL.html

Método getUserInfo():null

Método getPath():/javase/7/docs/api/java/net/URL.html

Método getAuthority():www.oracle.com

Método getQuery():null

**Solución**

package paquete;

import java.net.URL;

class Actividad4 {

public static void main(String[] args) {

URL url1, url2, url3, url4;

try {

// Constructor simple para una URL

url1 = new URL("http://docs.oracle.com");

*Visualizar*(url1);

// Constructor para protocolo, host y directorio

url2 = new URL("http://docs.oracle.com/javase/7");

*Visualizar*(url2);

// Constructor para protocolo, host, puerto y directorio

url3 = new URL("http://docs.oracle.com:80/javase/7");

*Visualizar*(url3);

// Constructor para un objeto URL y su directorio

url4 = new URL(url2, "/docs/api/java/net/URL.html");

*Visualizar*(url4);

} catch (Exception e) {

e.printStackTrace();

}

}

public static void Visualizar(URL url) {

System.***out***.println("Constructor simple para una URL:");

System.***out***.println("\tMétodo toString(): " + url.toString());

System.***out***.println("\tMétodo getProtocol(): " + url.getProtocol());

System.***out***.println("\tMétodo getHost(): " + url.getHost());

System.***out***.println("\tMétodo getPort(): " + url.getPort());

System.***out***.println("\tMétodo getFile(): " + url.getFile());

System.***out***.println("\tMétodo getUserInfo(): " + url.getUserInfo());

System.***out***.println("\tMétodo getPath(): " + url.getPath());

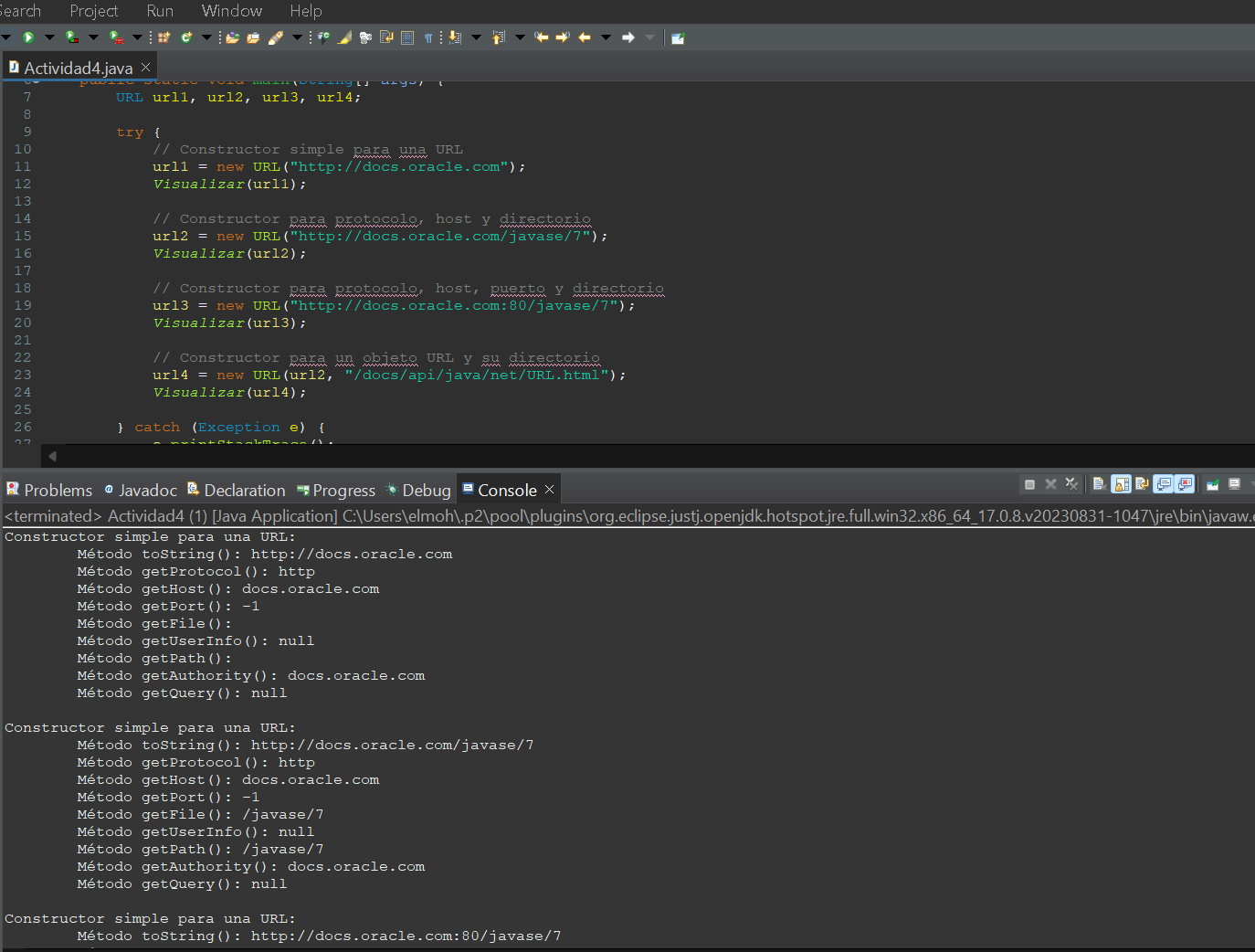
System.***out***.println("\tMétodo getAuthority(): " + url.getAuthority());

System.***out***.println("\tMétodo getQuery(): " + url.getQuery());

System.***out***.println();

}

}

****

**Resolución de problemas**

**ACTIVIDAD 5**

**Enunciado**

Realizar un programa con un método privado llamado VisualizarConexión(), que tenga un parámetro que sea un objeto URLConnetion y que visualice el resultado de aplicar diferentes métodos de la clase URLConnection.

El nombre de la clase principal será Actividad5.

La URL con la que se realizará la conexión es www.vitoria-gasteiz.org

El resultado esperado es el siguiente:

Conexión con www.vitoria-gasteiz.org

==========================

Método toString():sun.net.www.protocol.http.HttpURLConnection:http://www.vitoria-gasteiz.org

Método Fecha():1636109544000

Método getContentType():text/html; charset=iso-8859-1

Campos Cabecera con getHeaderField:

====================================

Linea 1:HTTP/1.1 302 Found

Linea 2:Fri, 05 Nov 2021 10:52:24 GMT

Linea 3:Apache

Linea 4:https://www.vitoria-gasteiz.org/

Linea 5:216

Campos Cabecera con getHeaderFields:

====================================

Keep-Alive:[timeout=10, max=100]

null:[HTTP/1.1 302 Found]

Server:[Apache]

Connection:[Keep-Alive]

Content-Length:[216]

Date:[Fri, 05 Nov 2021 10:52:24 GMT]

Content-Type:[text/html; charset=iso-8859-1]

Location:[https://www.vitoria-gasteiz.org/]

**Solución**

package paquete2;

import java.io.IOException;

import java.net.URL;

import java.net.URLConnection;

import java.util.List;

import java.util.Map;

public class Actividad5 {

public static void main(String[] args) {

String urlString = "http://www.vitoria-gasteiz.org";

try {

URL url = new URL(urlString);

URLConnection urlConnection = url.openConnection();

*VisualizarConexion*(urlConnection);

} catch (IOException e) {

e.printStackTrace();

}

}

private static void VisualizarConexion(URLConnection urlConnection) {

System.***out***.println("Conexión con " + urlConnection.getURL());

System.***out***.println("==========================");

// Métodos de la clase URLConnection

System.***out***.println("\tMétodo toString(): " + urlConnection.toString());

System.***out***.println("\tMétodo Fecha(): " + urlConnection.getDate());

System.***out***.println("\tMétodo getContentType(): " + urlConnection.getContentType());

// Campos Cabecera con getHeaderField

System.***out***.println("\nCampos Cabecera con getHeaderField:");

System.***out***.println("====================================");

for (int i = 0; ; i++) {

String headerField = urlConnection.getHeaderField(i);

if (headerField == null) break;

System.***out***.println("\t\tLinea " + (i + 1) + ":" + headerField);

}

// Campos Cabecera con getHeaderFields

System.***out***.println("\nCampos Cabecera con getHeaderFields:");

System.***out***.println("====================================");

Map<String, List<String>> headerFields = urlConnection.getHeaderFields();

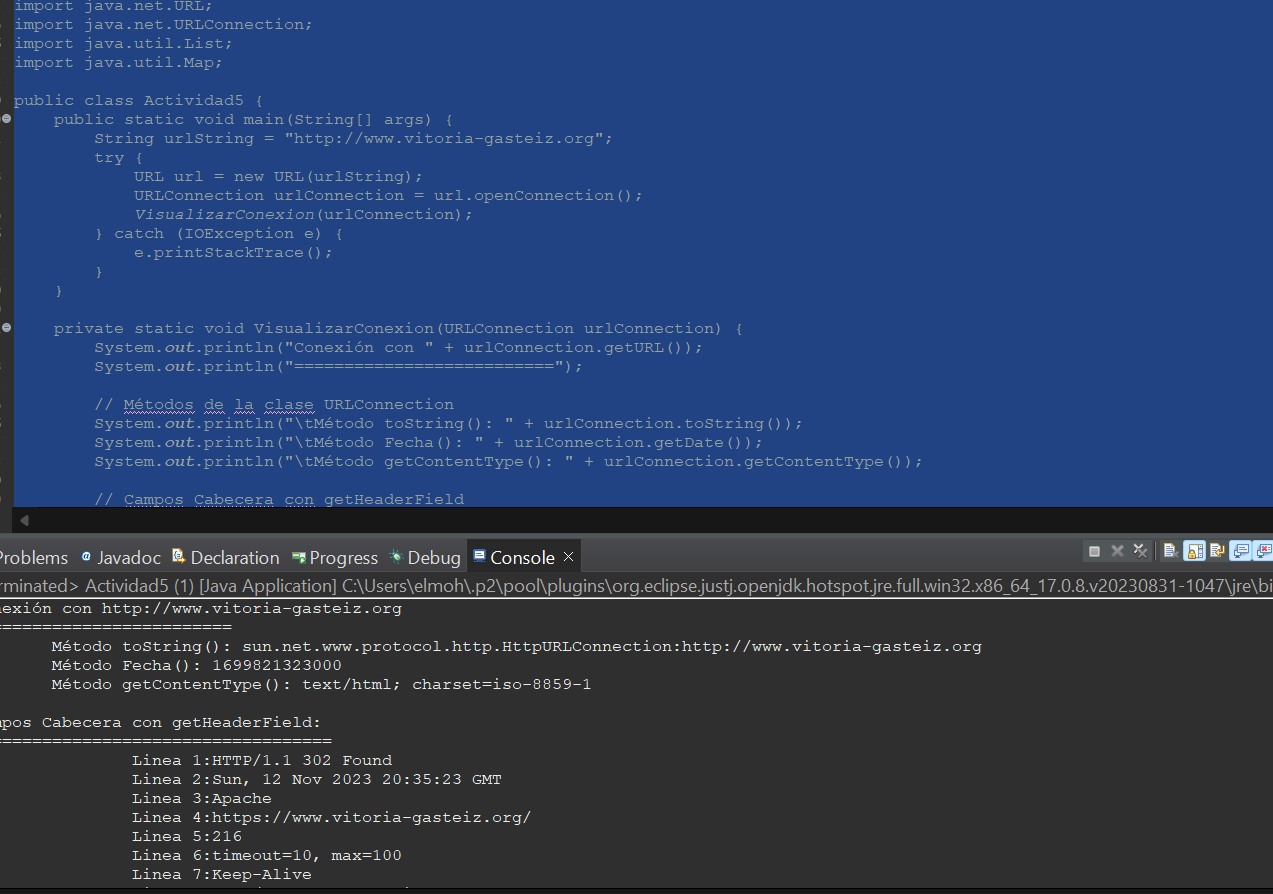
for (Map.Entry<String, List<String>> entry : headerFields.entrySet()) {

System.***out***.println(entry.getKey() + ":" + entry.getValue());

}

}

}

****

**Resolución de problemas**

**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%).