



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
Ing. En Comp. y L.S.C.	2003-1	5038	Programación Orientada a Objetos II

PRÁCTICA No.	LABORATORIO DE	Ingeniero en Computación y Licenciado en Sistemas Computacionales	DURACIÓN (HORA)
1	NOMBRE DE LA PRÁCTICA	Conexión URL	2

1. INTRODUCCIÓN

URL viene de Uniform Resource Locator y es una referencia (una dirección) a un recurso de Internet. Por ejemplo <http://www.yahoo.com>

Para realizar una conexión a una URL, se crea un objeto de tipo **URLConnection**. Este objeto permite obtener algunos de los aspectos de alto nivel de la dirección URL, como son el tipo de contenido, la fecha de la última modificación o la propia dirección URL.

2. OBJETIVO (COMPETENCIA)

Que el alumno tenga conocimiento del uso de la clase **URLConnection** y de sus métodos principales.

3. FUNDAMENTO

El protocolo TCP para conexiones de red, esta formado por la dirección IP y el número de puerto. La URL <http://www.yahoo.com> equivale a la dirección IP(205.216.146.71), se utiliza el servicio DNS -Domain Name Service para la transformación. Si no se especifica número de puerto se utiliza el 80 que es el comúnmente usado para servicios Web. <http://www.yahoo.com> : 80

En el paquete NET de Java encontramos la clase URL que contiene constructores y métodos para su manipulación.

Java permite los siguientes cuatro constructores para la clase URL:

1. public URL(String spec)
2. public URL(String protocol,String host,int port,String file)
3. public URL(String protocol,String host,String file) throws MalformedURLException;
4. public URL(URL context,String spec)

Al construir una URL con cualquiera de los constructores anteriores, se puede lanzar la excepción **MalformedURLException**, cuando los elementos del constructor son nulos o el protocolo es desconocido.

Formuló L.S.C. Lourdes E. Ramirez Fdez.	Revisó M.C. Gloria Etelbina Chavez Valenzuela y LSC Monica Lam Mora	Aprobó	Autorizó M.C. Miguel Ángel Martínez Romero
Maestro	Coordinador de Programa Educativo	Gestión de Calidad	Director de la Facultad



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

Los métodos que permiten obtener información sobre los componentes de la URL creada, son:
String getProtocol() String getHost() int getPort() String getFile()

Leer desde una URL

Una vez creada una URL, podemos establecer un canal que permita leer el contenido de la URL, para esto llamamos al método **openStream()** de la clase URL. El método retorna un objeto java.io.InputStream por lo que se puede leer normalmente de la URL utilizando los métodos normales de InputStream.

Cuando apliques este código, debes ver los comandos HTML y el contenido textual del fichero HTML localizado en la URL especificada.

URLConnection es una clase *abstracta* que puede ser *extendida*, con un constructor protegido que admite un objeto **URL** como parámetro. Esta clase pertenece al paquete **NET** de java. Su constructor recibe como parámetro un objeto **URL**.

URLConnection(URL url)

Cuenta con aproximadamente cuarenta métodos que permiten examinar y manipular el objeto creado. Algunos de ellos son:

- URL getURL() devuelve la dirección de la URL
- long getLastModified() regresa la fecha de la ultima modificación
- String getContentType() regresa el tipo de documento

```
import java.net.*;
import java.util.Date;
import java.io.*;
```

```
public class ConexionURL {
    public static void main(String[] args) {
        String cadena;
        try {
            URL url = new URL( "http://yaqui.mx1.uabc.mx" );
            // muestra el codigo del archivo
            BufferedReader paginaHtml = new BufferedReader( new InputStreamReader(url.openStream()) );
            while( ( cadena = paginaHtml.readLine()) != null ) {
                System.out.println( cadena );
            }
            // metodos que regresan información sobre la URL
            URLConnection conexion = url.openConnection();
            Date fecha = new Date( conexion.getLastModified() );
            System.out.println("FECHA DE ULTIMA MODIFICACION: " + fecha );
            System.out.println("TIPO DE CONTENIDO: " + conexion.getContentType() );
        }catch(MalformedURLException e) {
            System.out.println("Error al crear el URL " + e);
        }
        catch(IOException e) {
            System.out.println("Error de I/O " + e);
        }
    }
}
```



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formatos para prácticas de laboratorio

4. PROCEDIMIENTO (DESCRIPCIÓN)

A)	EQUIPO NECESARIO	MATERIAL DE APOYO
	Equipo de computo con SDK de Java Editor de Java - NetBeans	Practica impresa

B) DESARROLLO DE LA PRÁCTICA

1. Elabora una clase que realice una conexión a un URL , este se recibirla como parámetro y deberá contar con métodos que permitan:

- Abrir la conexión e indicar el resultado
- Verificar si el tipo de contenido del URL es texto
- Regresar el contenido del URL en una sola cadena (solo si es texto)
- Cerrar la conexión

2. Elaborar una clase con el método main:

- Dentro de un ciclo pedir una opción al usuario
 - Puede ser un URL
 - Puede ser la palabra salir

Si se elige la opción de URL, crear un objeto de la clase creada en el punto anterior, utilizar los métodos definidos para establecer la conexión y mostrar el contenido solo en caso de ser tipo texto.

3. Elaborar una clase grafica con las opciones del programa anterior, la interfaz contendrá un campo de texto para capturar el URL, un botón que cree el objeto de la clase creada en el primer punto y ejecute los métodos correspondientes, mostrar el contenido del URL en un área de texto.

C) CÁLCULOS Y REPORTE

Se realizarán preguntas al alumno para verificar la comprensión del tema.

5. RESULTADOS Y CONCLUSIONES

El alumno podrá realizar una conexión a una URL creada para leer el archivo HTML y aplicar los métodos de la clase URLConnection para obtener información sobre dicha URL.

6. ANEXOS

www.java.sun.com

7. REFERENCIAS