

*Universidad de Panamá
Facultad de Informática, Electrónica y Comunicación
Centro Regional Universitario de Veraguas*

*Proyecto Parte del Semestral
CLASE URL*

*Autor
Faustino Aguilar Quirós
2-732-727*

*Facilitador
Prof. Abundio Mendoza*

II° Semestre de 2015

RECURSO WEB Y ARCHIVO DE TEXTO

En este proyecto se utilizó el lenguaje de programación Java para desarrollar un programa de computadore que leyera una URL y devolvier su código para luego modificarlo y ver su resultado.

SOBRE JAVA Y LAS URL'S

Casi la mayoría de los lenguajes más populares hoy en día tiene la capacidad de manejar peticiones web y funcionar como servidor, Java es uno de ellos y a continuación se explicará una breve introducción al manejo de las URL.

Tenemos nuestra clase principal, En ella importamos la librería `java.net.*` que nos permitirá acceder a las clases para manipular las URL.

Una vez ya tengamos esta librería importada tenemos que creal un objeto URL, que una ves creado podemos llamar a su método `OpenConnection()` para obtener un objeto `URLConnection` o una de sus subclases de protocolos específicos como `HttpURLConnection`, ya que existen otros protocolos como lo son FTP, FILE, etc.

La conexión hacia la máquina y/o recurso remoto representado por el objeto URL solo se inicia cuando se llama al método `connect()` del objeto `URLConnection`, con esto iniciamos un enlace de conexión entre nuestra aplicación Java y la URL a través de la red. Cabe decir que en algunos casos no es necesario llamar explícitamente el método `connect`, ya que este es ejecutado de forma implícita cuando se ejecutan métodos como `getInputStream`, `getOutputStream` entre otros.

Como puede verse los cambios son muy pocos, como dije antes aunque `HttpURLConnection` es una clase específica para comunicaciones con el protocolo http, la clase `URLConnection` es muy http-céntrica.

Un tema que no abarcamos en este documento pero que es muy interezante mencionarlo es que las URL tienen diferentes métodos para llegar y envíar información a travez de ellas.

Algunos métodos más conocidos son:

- GET: es el más genérico y Nos permite realizar una petición de manera visible en la url del navegador.
- POST: fue inventado para lograr enviar datos de mayor capacidad y de manera mas confidencial.
- PUT & UPDATE & DELETE: Son métodos pensados en la era moderna de las API's hace algo parecido Al POST pero de manera REST que quiere decir que puede hacerse a tra vez de servidores asincrónicos. Con protocolos como Websockets

FILTRANDO SITIOS WEB

LeerWeb.java

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

class LeerWeb {

    public static void main(String[] args) {
        // Se encarga de leer el sitio web
        String url, html, file;
        System.out.println("\n\tPROGRAMA PARA GESTIÓN DE DOCUMENTO HTML\n");
        LeerWeb web = new LeerWeb();
        url = web.capturaURL();
        html = web.filtrarHTML(url);
        file = web.generarNombre(url);
        web.guardarArchivo(html, file);
        ModificarWeb neoWeb = new ModificarWeb(file);
        neoWeb.cambiarTitulo();
        neoWeb.agregarParrafo();
        web.guardarArchivo(neoWeb.data, file);
    }

    public String capturaURL() {
        // Captura la URL
        Scanner scan = new Scanner(System.in);
        System.out.print("Escribe una URL: ");
        return scan.nextLine();
    }

    public String generarNombre(String url) {
        // Genera un nombre adecuado para el archivo
        String nombre;
        Pattern dominio = Pattern.compile("http://([\\w.]+)");
        Matcher result = dominio.matcher(url);
        result.find();
        nombre = result.group(1);
        nombre += ".html";
        return nombre;
    }

    public String filtrarHTML(String url) {
        // Filtra el HTML extrayendo lo necesario usando REGEX
        String inputLine, inputText="", htmlCode="";
        boolean titleTest, tagTest;
        try {
            System.out.println("Analizando URL...");
            URL URL = new URL(url);
```

```

        InputStreamReader isr = new InputStreamReader(URL.openStream());
        BufferedReader in = new BufferedReader(isr);
        while((inputLine = in.readLine()) != null) {
            inputText += inputLine;
        }
        in.close();
        Pattern title = Pattern.compile("<title>((.|\\n)+?)(</title>)");
        Pattern tags = Pattern.compile("<(p|h1|h2|h3)(\\s(.*)?)?>(.*?)</(p|h1|h2|h3)>");
        Matcher getTitle = title.matcher(inputText);
        Matcher getTags = tags.matcher(inputText);
        htmlCode = "<!DOCTYPE html>\\n";
        htmlCode += "<html>\\n";
        htmlCode += "<head>\\n";
        htmlCode += "<meta charset='utf-8'>\\n";
        while (getTitle.find()) {
            htmlCode += getTitle.group(0);
            htmlCode += "\\n";
        }
        htmlCode += "</head>\\n";
        htmlCode += "<body>\\n";
        while (getTags.find()) {
            htmlCode += getTags.group(0);
            htmlCode += "\\n";
        }
        htmlCode += "</body>\\n";
        htmlCode += "</html>\\n";
    } catch (Exception e) {
        System.out.println("Error al filtrar URL: " + e);
        System.exit(0);
    }
    return htmlCode;
}

public void guardarArchivo(String data, String file) {
    // Guarda ekl archivo creado
    try {
        File text = new File(file);
        BufferedWriter bw = new BufferedWriter(new FileWriter(text));
        bw.write(data);
        bw.close();
        System.out.println("Datos guardados en: " + file + "\\n");
    } catch (Exception e) {
        System.out.println("Error al guardar archivo: " + e);
        System.exit(0);
    }
}
}

```

ModificarWeb.java

```
// Clase para hacer modificaciones en el código HTML

import java.io.*;
import java.util.*;
import java.util.regex.*;
class ModificarWeb {

    public String data;
    private Scanner s = new Scanner(System.in);

    ModificarWeb (String path) {
        // El constructor asigna los datos de la pagina web a una variable
        this.data = leerHTML(path);
        System.out.println("\n\tMODIFICACIONES DEL DOCUMENTO HTML\n");
    }
    private String leerHTML(String path) {
        // Se encarga de obtener los datos de un archivo
        String inputText="", inputLine;
        try {
            FileReader fr = new FileReader(path);
            BufferedReader br = new BufferedReader(fr);
            while((inputLine = br.readLine()) != null) {
                inputText += inputLine;
            }
            br.close();
        } catch (IOException e) {
            System.out.println("Error al leer archivo: " + e);
            System.exit(0);
        }
        return inputText;
    }
    public void agregarParrafo() {
        // Se encarga de agregar un párrafo al HTML
        String p="<p>";
        System.out.print("Escriba un párrafo: ");
        p += s.nextLine();
        p += "</p></body>";
        this.data = this.data.replace("</body>", p);
    }
    public void cambiarTitulo() {
        // Cambia el título del HTML
        String t;
        System.out.print("Escriba un título: ");
        t = s.nextLine();
        t = "<title>" + t + "</title>";
        Pattern title = Pattern.compile("<title>((.|\\n)+?)</title>");
        this.data = title.matcher(this.data).replaceAll(t);
    }
}
```

EVIDENCIAS

Puede ser visitado el sitio para ver el código original: <http://faustinoaq.github.io>

```
4.URL : bash
faustino@vboxsrv:~/Projects/tareasjava/4.URL$
faustino@vboxsrv:~/Projects/tareasjava/4.URL$
faustino@vboxsrv:~/Projects/tareasjava/4.URL$ java LeerWeb

PROGRAMA PARA GESTIÓN DE DOCUMENTO HTML

Escribe una URL: http://faustinoaq.github.io
Analizando URL...
Datos guardados en: faustinoaq.github.io.html

MODIFICACIONES DEL DOCUMENTO HTML

Escriba un título: Hola Mundo
Escriba un párrafo: Este es un nuevo párrafo
Datos guardados en: faustinoaq.github.io.html

faustino@vboxsrv:~/Projects/tareasjava/4.URL$
```

The screenshot shows a terminal window with a dark background. The title bar reads "4.URL : vim". The terminal content shows the creation of an HTML file named "Hola Mundo" using the `vim` editor. The HTML code is as follows:

```
<!DOCTYPE html><html><head><meta charset="utf-8"><title>Hola Mundo</title></head><body><h1>Hello!</h1><p>My name is Faustino.</p><p>I am a Computer Science student.</p><p>Este es un nuevo párrafo</p></body></html>
```

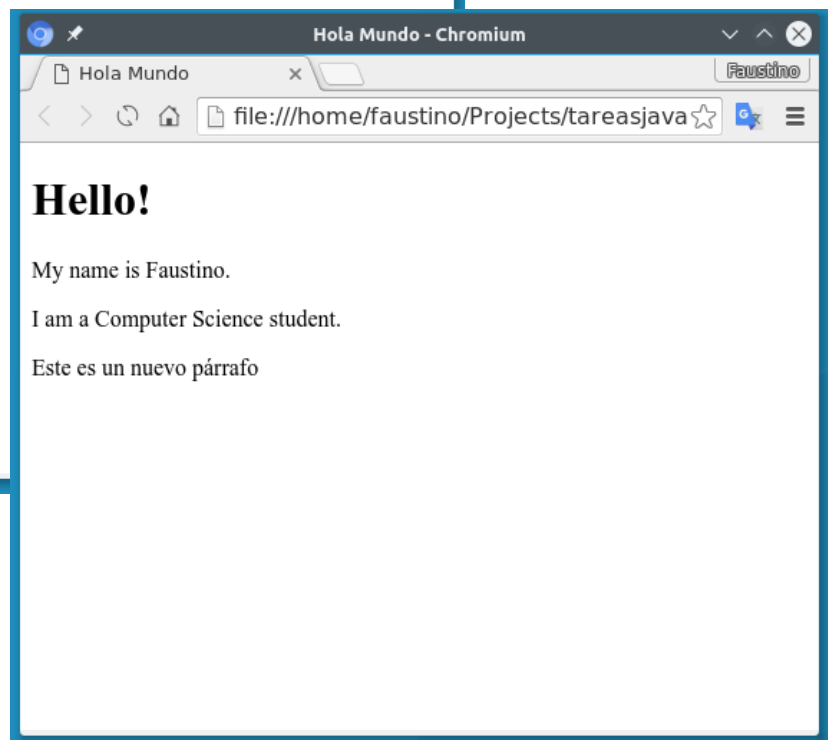
The terminal also shows the command `:set wrap` being entered. To the right of the terminal, a web browser window titled "Hola Mundo - Chrome" is open, displaying the rendered HTML file. The address bar shows the file path `file:///home/faustino/...`. The browser content displays:

Hello!

My name is Faustino.

I am a Computer Science student.

Este es un nuevo párrafo



Se captura una URL y se modifica su contenido. Se puede probar con cualquier Dirección,
¡Intente por Usted Mismo!

COMENTARIOS

Como un entusiasta de la tecnología estoy aprendiendo algo de Android y su uso con Java, lo único que me falta es experiencia, no se cual es la potencia real de este lenguaje respecto a otros, pero si se que es muy importante en el desarrollo de aplicaciones modernas.

Aunque más que el lenguaje tengo que decir que la verdadera herramienta es el POO que ayuda a tener una nueva estructura en el código realizando las cosas de manera más eficiente para resolver problemas.

Java es un buen lenguaje, muchas personas en la sociedad le deben su vida a este lenguaje, pero yo tengo que decir que hay que ser agnóstico a la tecnología y creyente en los principios físicos y matemáticos que hacen de la informática una ciencia tan importante hoy en día.

REFERENCIAS

Kurzyniec, D., & Sunderam, V. (2001, June). Efficient cooperation between Java and native codes—JNI performance benchmark. In The 2001 international conference on parallel and distributed processing techniques and applications. **Utilizado para comprender mejor ciertos puntos de eficiencia en Java.**

Odersky, M., & Wadler, P. (1997, January). Pizza into Java: Translating theory into practice. In Proceedings of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages (pp. 146-159). ACM. **Utilizado para extracción de código y ejemplos.**

Pratt, T. W., Zelkowitz, M. V., & Gopal, T. V. (1984). Programming languages: design and implementation (No. 04; QA76. 7, P8 1984.). Englewood Cliffs: Prentice-Hall. **Utilizado para extracción de código y ejemplos.**

Gundersen, G., & Steihaug, T. (2004). Data structures in Java for matrix computations. Concurrency and computation: Practice and Experience, 16(8), 799-815. **Utilizado para comprender mejor la relación entre Java y sus estructuras de datos.**