

INSTITUTO TECNOLÓGICO SUPERIOR DE CALKINÍ EN EL ESTADO DE  
CAMPECHE.

**Carrera:**

INGENIERÍA EN SISTEMAS  
COMPUTACIONALES.

**Asignatura:**

TOPICOS AVANZADOS DE  
PROGRAMACION

**Docente:**

JOSE LUIS LIRA TURRIZA

**Actividad:**

REPORTE DE PRÁCTICA NO. 3.-  
BIBLIOTECAS Y PAQUETES A  
PARTIR DE REQUERIMIENTOS

**Equipo:**

CARLOS FERNANDO CHAN CAUICH  
(7010)

**Fecha:**

23/03/2023

**Ciclo escolar:**

2022-2023P

## **PRÁCTICA NO. 3.- BIBLIOTECAS Y PAQUETES A PARTIR DE REQUERIMIENTOS**

### **INTRODUCCIÓN**

Para ayudar a explicar el tema de paquetes se debe imaginar una ciudad en la cual hay varios bloques de apartamentos que pertenecen a una única empresa inmobiliaria. Además de los departamentos, la empresa tiene diferentes servicios como comercios, zonas de recreo y almacenes. La empresa tiene organizadas todas sus propiedades como una lista de referencias, para que pueda hacer uso de ella en el momento en que lo necesite.

Si ahora se mira lo anterior en términos de Java, la empresa inmobiliaria es el paquete. Los paquetes agrupan a bibliotecas de clases, como las bibliotecas que contienen información sobre distintas propiedades comerciales. Un paquete será entonces, la mayor unidad lógica de objetos en Java.

Los paquetes se utilizan en Java de forma similar a como se utilizan las bibliotecas en C++, para agrupar funciones y clases, sólo que en Java agrupan diferentes clases y/o interfaces. En ellas las clases son únicas, comparadas con las de otros paquetes, y además proporcionan un método de control de acceso. Los paquetes también proporcionan una forma de ocultar clases, evitando que otros programas o paquetes accedan a clases que son de uso exclusivo de una aplicación determinada.

### **OBJETIVO**

Al finalizar la práctica, el alumno podrá tener la capacidad de crear paquetes dentro de un sistema informático de acuerdo a un conjunto de requerimientos.

### **LUGAR**

AULA

### **SEMANA DE EJECUCIÓN**

Semana SIETE (Parcial 2)

## MATERIAL Y EQUIPO

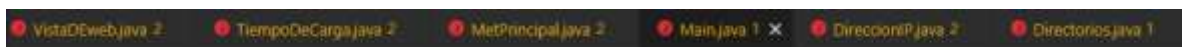
- Sistema Operativo
- Procesador de Textos
- Software para el desarrollo de aplicaciones “eclipse”.
- Cañón
- Plumones
- Pizarrón.
- Equipos de cómputo para todos los estudiantes de la asignatura.

## DESARROLLO DE LA PRÁCTICA

Los paquetes que pertenecen a un lenguaje fortalecen y aceleran el desarrollo de las aplicaciones en él. En esta práctica deberán elaborar una aplicación que permita consultar una página web (itescam de preferencia) y traer los datos informativos (IP, documento principal, directorios existentes, tiempo de carga de la página, etc) de ella.

1. De acuerdo a las necesidades de la aplicación crear un paquete llamado información web, que contenga las clases que representen la información web que se pueda traer de una página web. De ser posible crear paquetes internos que permita clasificar las clases de acuerdo a sus características.
2. Realiza la tarea tanto en Java como en C#.
3. Realiza una comparativa con los dos lenguajes remarcando sus diferencias.

### Documentación y funcionamiento del programa.



se nos presentan 6 clases, las cuales nos servirán para obtener todos y cada uno de los datos que se nos solicitaron, todos estos datos serán obtenidos de la página del itescam como se mencionó; esto se hizo con la ayuda del socket, que como bien se sabe es un canal de comunicación que nos permiten leer y/o escribir información obtenida desde la red, por lo tanto, se hizo uso de este para poder tomar y mostrar la información de la página web de la itescam.

En el llamado MetPrincipal lo que se hace es definir un socket con un determinado Host, esto para poder obtener la información de la página, en términos simples es la que nos permite leer los datos que se encuentran en la página con la que queremos trabajar.

Para la obtención de los directorios se usó la clase de Directorios, en la que se importaron varios paquetes como org.jsoup.nodes.Document, entre otros, los cuales nos permitirán obtener los datos de una página web, como se puede observar primero se definen varias variables, una para almacenar los directorios, otra para ir almacenándolos y una para el tamaño, se observa que se tienen dos métodos, en el primero de ellos, se van guardando los directorios en un arreglo para luego ser mostrados, al igual que se obtiene el tamaño o número total de directorios para ser retornado en el segundo método, esto porque nos servirá más adelante.

En la clase de DireccionIP como su nombre lo indica, esta nos permite obtener la dirección IP de la página web, se hace mediante un método que recibe un String denominado host el cual contiene la dirección de la página, en este método se obtiene la dirección IP implementando un socket y guardando en un String el IP con el método de getInetAddress, luego simplemente la retorna

Luego tenemos la clase de TiempoDeCarga en la cual como su nombre lo indica obtiene el tiempo de carga de la página, esto se hace de manera similar al de la dirección IP ya que se tiene un método que recibe el Host y con la ayuda del socket se obtiene la información de esta página

La clase siguiente es la de VistaDEweb, en esta simplemente se crea toda la parte visual del código, es decir se importan los paquetes correspondientes a los mencionados anteriormente para poder ser usados, entonces simplemente se crea una ventana y se definen los Host como String y se pone la dirección de la página web de la itescam, en este caso se usa el "https://itescam.edu.mx" y el "itescam.edu.mx", luego solo se definen cada uno de los componentes visuales, en esta se crean 5 botones, correspondientes a la obtención del tiempo de carga, de los directorios, del número total directorios, de la dirección IP y un botón para limpiar el textArea. Como ya se mencionó de igual manera se tiene un textArea en el cual

se mostrarán los datos. Al final solo se le da funcionalidad a cada botón, esto no se explica a mayor detalle, debido a que el funcionamiento ya es bien conocido; entonces solo se le da función a cada botón, llamando a cada método correspondiente a la información que se quiere obtener al pulsar el botón correspondiente, en resumen, es el encargado de presentar la parte visual del programa.

Al final tenemos la clase Main, la cual simplemente crea un objeto de VistaDEweb y esta es la que se encarga de ejecutar y el programa, ya que en la parte visual es en la que se le da función y se llama a cada método y el Main es la que ejecuta esta parte visual y todos los métodos relacionados con este.

## EJECUCION



Como se observa, al momento de ejecutar el programa, se nos muestran los 5 botones que se mencionaron anteriormente, así como el textArea en el que se mostrarán todos los datos.



Obtención de directorios.

En este caso se puede ver que al momento de presionar el botón de Directorios se nos muestran todos y cada uno de los directorios que tiene la página, presentando así una de las 4 funciones que tiene este programa.



Obtención de ip

En la imagen, como se ve, se nos muestra la IP de la página, esto se consigue oprimiendo el botón correspondiente, en este caso el que dice IP



En este caso se nos muestra el número de directorios, esto ya se mencionó anteriormente como se realiza, en este caso solo se llama al método correspondiente de la clase Directorios al momento de hacer clic al botón y este es mostrado en el textArea

## **Resultados.**

Como se pudo observar, en este caso la práctica numero 3 solamente se hizo en el lenguaje de Java, ya que este presentaba una gran complejidad y realizarla si nos llevó mucho tiempo, por lo tanto, se presenta lo solicitado en un único lenguaje, presentado de manera visual, por lo tanto, se considera que esto fue lo mejor que se pudo realizar.

Cabe mencionar que se inició el proceso de creación del programa en el lenguaje de C#, pero debido al tiempo y al poco entendimiento que se tenía sobre este lenguaje, así como la falta de comprensión que se tuvo a la información que se estuvo leyendo de esto, no se comprendió del todo, así que no se pudo realizar.

## **Conclusión.**

Luego de haber realizado esta práctica, no me quedan más que buenos aprendizajes, ya que al ser un tema del que se habló muy poco en clase o se mencionó de manera general, sí presentó un problema el realizarse de manera adecuada, por lo tanto no me quedan más que sentimientos positivos al poder presentar esta práctica en al menos un lenguaje como es el de Java, el cual es el que se dominaba de una mejor manera y en el que se logró rescatar la información de una mejor manera, debido a que en el lenguaje de C# aún nos queda mucho por aprender y por cuestiones de tiempo, así como por falta de información, no se pudo completar

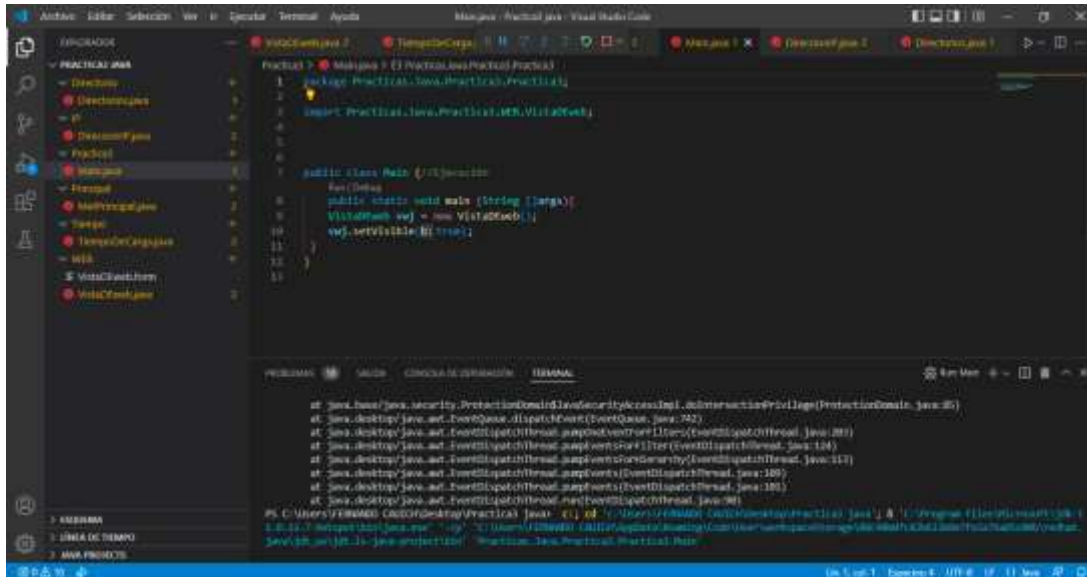
## **EVALUACIÓN Y RESULTADOS**

Se describe la forma de evaluar la práctica desarrollada mediante la entrega del Informe Técnico solicitado por el Profesor, el cual puede contener tablas, planos, prototipos, gráficas, diagramas o dibujos, observaciones, conclusiones, cuestionario y referencias.

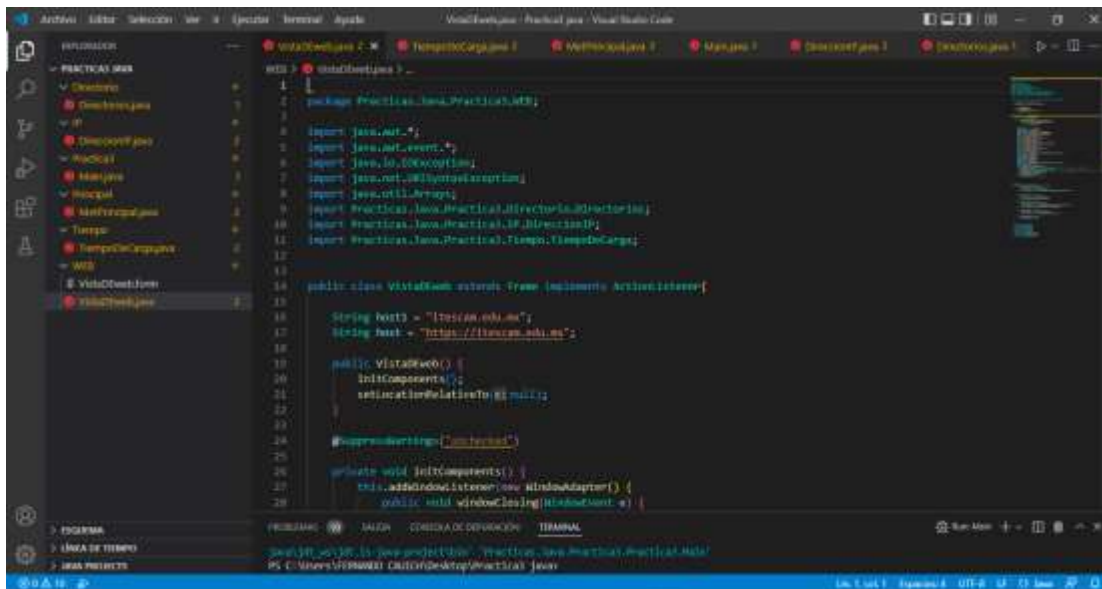
## **REFERENCIAS**

Agustín Froufe. Paquetes y declaración de paquetes. Link: <http://dis.um.es/~bmoros/Tutorial/parte5/cap5-14.html>. Febrero de 2016

## ANEXOS



## Capturas de código





The screenshot shows the Visual Studio Code editor with the file `Directory.java` open. The code implements a class `Directorio` that fetches a directory from a web server. The `conjuntoDirectorio` method uses `HttpClient` to get a document from `https://www.edu.es`, selects all hrefs, and returns them as an array of strings. The `main` method is commented out.

```
1 import org.jsoup.select.Elements;
2
3
4 public class Directorio { //repositorio de Directorio
5
6     String[] dirs;
7     int i = 0;
8     int tam = 0;
9
10    public String[] conjuntoDirectorio(String host) {
11        try {
12            String url = "https://www.edu.es";
13
14            Document doc = Jsoup.connect(host).get();
15            Elements imports = doc.select("a[href]");
16            dirs = new String[imports.size()];
17            tam = imports.size();
18
19            for (Element link : imports) {
20                dirs[i] = (link.tagName() + link.attr("href") + "\n");
21                i++;
22            }
23            return dirs;
24        } catch (IOException e) {
25            return null;
26        }
27    }
28 }
```

The screenshot shows the Visual Studio Code editor with the file `Tiempo.java` open. The code implements a class `TiempoCarga` that measures the time taken to load a page from a web server. The `main` method is commented out.

```
1 package PracticasJava.Practica1.Tiempo;
2
3 import java.net.*;
4
5 public class TiempoCarga {
6     public int carga(String host) //repositorio de tiempo de carga
7     {
8         try {
9             Socket socket = new Socket(host, port(8080));
10            int tiempo = socket.getSoTimeout();
11            return tiempo;
12        } catch (IOException e) {
13            return 0;
14        }
15    }
16 }
```