Práctica 3 - Crawler

En esta documentación vamos a tratar los distintos aspectos relevantes de práctica.

1. Requisitos

Para poder compilar y ejecutar la práctica con éxito se recomienda:

- Poseer un IDE de java (como eclipse o IntelliJ IDEA).
- Abrir el directorio /Practica3 como proyecto.
- Descargar la librería JSoup (disponible en la entrega, ruta /Practica3/lib)
- Una vez cargue comprobar si la librería JSoup está añadida al proyecto. Si no fuera el caso, añadirla manualmente como podemos ver en las Figuras 1 y 2.
- Por último, se recomienda leer el Javadoc contenido dentro del directorio /Practica3/javaDoc. Para ello diríjase al directorio y abra el archivo index.html

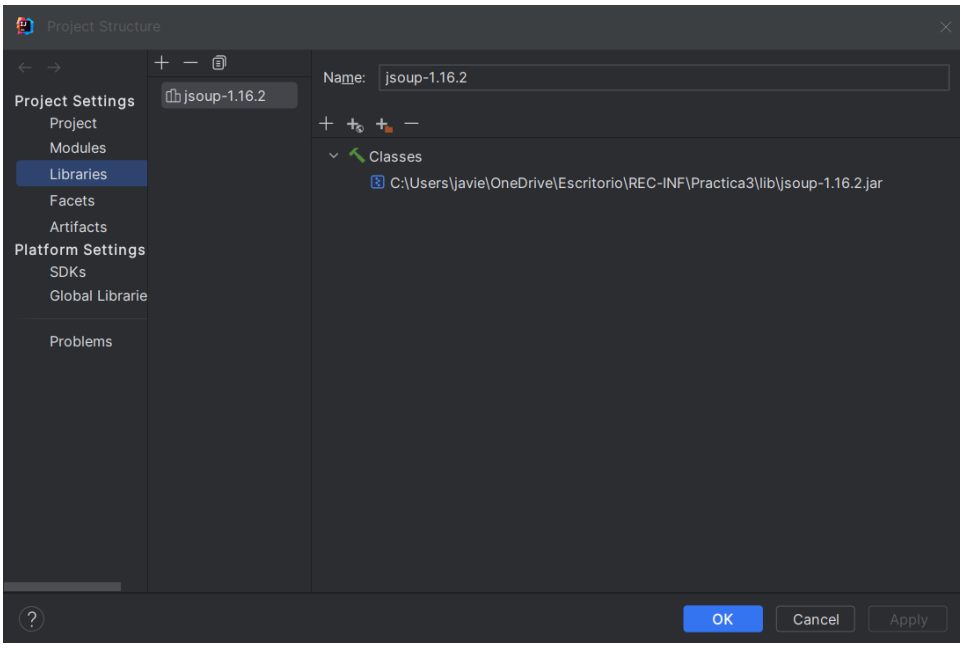

Figura 1 Añadir librería externa al proyecto usando IntelliJ IDEA

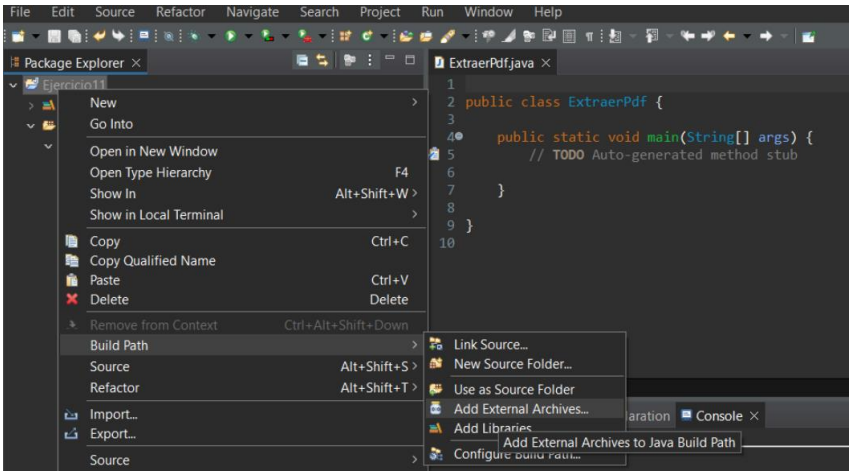

Figura 2 Añadir librería externa al proyecto usando IntelliJ IDEA

2. Especificación

Vamos a realizar 2 clases, la primera implementará un Crawler con todos sus métodos (Crawler.java). Está clase NO es ejecutable, solo compilable. La otra clase será una clase compilable y ejecutable que use la anterior clase (Main.java).

2.1. Clase Main

Método “public static void main(String[] args)”: Realiza una prueba de la clase Crawler. Puedes cambiar el código para probar los constructores de la clase Crawler cambiando el limite de webs y la web “semilla”.

2.2. Clase Crawler

Atributos:

- private Queue<String> urlQueue; (Cola de prioridad que controla las URLS que hacen “crawl”).
- private Vector<String> visitedUrl; (Vector que contiene las URLS visitadas, tiene como objetivo evitar webs ya visitadas).
- private String seedUrl; (Cadena que simboliza la URL por la que empezamos, la url “semilla”).
- private static int actualFile = 0; (Usado para dar a cada archivo HTML un identificador único).
- private Pattern esEspaniolaAbsoluta = Pattern.compile("https?:\\/\\/es\\.wikipedia\\.org\\/wiki\\/.*"); (Patrón usado para saber si una URL pertenece a Wikipedia, es española y es absoluta).
- private Pattern esEspaniolaRelativa = Pattern.compile("\\/wiki\\/.*"); (Patrón usado para saber si una URL pertenece a Wikipedia, es española y es relativa).

Constructores:

- public Crawler(String seedUrl) Constructor omitiendo el limite del crawler. Asigna la URL semilla pasada como argumento y le da un límite al crawler con valor 100000.
- public Crawler(String seedUrl, long maxUrls) Constructor con semilla y limite. Asigna la URL semilla pasada como argumento y asigna al limite de URLS el valor pasado como argumento.

Métodos:

- public void crawl() throws java.io.IOException Método principal de la clase Crawler. Este método es similar a un algoritmo BFS (Búsqueda en anchura). Mientras haya elementos en la cola, se obtiene el primer elemento (elemento prioritario) de la cola, se descarga y guarda en el directorio /Practica3/data, se registran las URLS que poseen, se añade el link actual al vector de visitados y se añaden a la cola de prioridad (siempre y cuando no esté en el vector de visitados).

- public void saveDocument(Document doc) throws IOException Método que usa la librería JSoup para almacenar el documento con su identificador en el directorio /Practica3/data con el siguiente formato: <idDocumento>.html
- public Vector<String> getLinks(Document doc) Método que recorre todos los links del documento y, guarda y devuelve un vector con las URLS que cumplan con las condición de los patrones (Debe ser española, de Wikipedia y absoluta o relativa).
- public void logActualUrl(Document doc) throws IOException Método que guarda en un fichero log, almacenado en el directorio /Practica3/log, la URL que se visita junto al identificador de la URL. Con esto conseguimos saber que identificador corresponde con que URL.