

PRACTICA 1

Sistemas Información Distribuidos



Universidad
Zaragoza

16/03/2016

Alejandro Solanas Bonilla NIP: 647647

Rubén Gabás Celimendiz NIP : 590738

Diseño:

El programa tiene la siguiente funcionalidad, se conecta a la url destino indicada, descarga el html y procede a aplicarle el parser a través de JavaCC, lo cual devuelve una estructura de *EMPRESA VALOR* en forma de String de las empresas que conforman el Ibex35, donde empresa es el nombre identificador, y valor el último valor registrado en bolsa. Una vez preparados los datos, se pide al usuario que introduzca el identificador de empresa que desea conocer su cotización actual, así como el valor obtenido en la última ejecución del programa. Hecho esto, se procede a actualizar el fichero de texto con los valores obtenidos.

Estructura:

El Proyecto, consta de dos ficheros principales, p1.jj, fichero asociado a JavaCC y last.txt, fichero de texto usado como almacén de los resultados obtenidos en la última ejecución del programa. Para poder lanzar el proyecto hay que compilar el fichero .jj (ver comando) , para después ejecutar los ficheros java generados normalmente. (Por defecto se ejecuta desde una carpeta raíz con un directorio src donde se encuentran el resto de ficheros). Además se ha incluido un parser completo de html, usado para entender el funcionamiento de JavaCC.

```
$ java -cp ruta_javacc.jar javacc ruta_p1.jj
```

Problemas Encontrados:

- Si se desea trabajar con un IDE, Eclipse es el único (hasta donde se ha probado) que tiene un módulo funcional para JavaCC, IntelliJ IDEA no genera correctamente el código java.
- Cuidado al trabajar con cualquier IDE, es posible que se auto importe un ParseException que no es necesario para el programa funcione, y que modifica algunos de los constructores de la clase, que ocasiona que los .java generados no funcionen.
- El hecho de no conocer la sintaxis específica de JavaCC, con ciertas diferencias a las expresiones regulares, sumado a la nula experiencia en JavaCC en este caso, hace que sea inicio duro con la práctica.
- Documentación confusa relacionada con SKIP, TOKEN y estados. En la documentación de javacc se menciona que realizar cambios de estado en SKIPS no es recomendable, sin embargo, es altamente usado en los ejemplos
- Usar el propio JavaCC como parser de un HTML que está mal formado, hace que la gramática a generar sea poco precisa y demasiado abierta, lo cual no deja claro hasta qué punto hay que afinar el uso de JavaCC como parser, pudiendo usar otras herramientas especializadas como JDOM.

Todo esto suma más de 20 horas de desarrollo en la práctica.

Documentación:

- Introducción a JavaCC: <https://javacc.java.net/doc/simpleREADME.html>
- Explicación avanzada: <https://javacc.java.net/doc/lookahead.html>
- Dudas generales: <http://www.engr.mun.ca/~theo/JavaCC-FAQ/javacc-faq-moz.htm>