



Procesadores del Lenguaje

Analizador de archivos Dockerfile

Alumnos:

1.- Raúl Negro Carpintero

2.- Yeray Sardón Ibáñez

## Tabla de contenido

|                              |          |
|------------------------------|----------|
| <b>INTRODUCCIÓN .....</b>    | <b>3</b> |
| <b>DOCKERFILE.....</b>       | <b>3</b> |
| <b>Características .....</b> | <b>3</b> |
| <b>ANALIZADOR .....</b>      | <b>4</b> |
| <b>BIBLIOGRAFÍA.....</b>     | <b>5</b> |

## INTRODUCCIÓN

Los archivos Dockerfile se utilizan para facilitar la creación automática de contenedores Docker. Docker es una herramienta que puede empaquetar una aplicación y sus dependencias en un contenedor virtual que se puede ejecutar en cualquier servidor Linux.

## DOCKERFILE

Un archivo Dockerfile está compuesto por una serie de comandos que el usuario quiere ejecutar en la línea de comandos. Los comandos son los siguientes:

- ADD
- COPY
- ENV
- EXPOSE
- FROM
- LABEL
- STOPSIGNAL
- USER
- VOLUME
- WORKDIR
- CMD
- RUN
- MAINTAINER
- ENTRYPOINT
- ARG
- ONBUILD
- HEALTHCHECK
- SHELL

Estas instrucciones son no case-sensitive. Sin embargo, para diferenciarlas sus argumentos, se escriben en mayúsculas por convención.

### Características

Un archivo Dockerfile siempre tiene que empezar por el comando FROM. El resto de instrucciones se ejecutan en orden.

Los comentarios se señalan con el símbolo # siempre al comienzo de línea. Si este símbolo aparece en cualquier otro lado, se considera como parte de los argumentos. En nuestro caso, no hemos podido analizar el símbolo # como parte de las cadenas, solo como inicio de comentarios, por lo que, si se prueba a introducirlo en medio de un argumento, el parser dará error.

Se utiliza \ para señalar que se pasa a la línea siguiente.

## ANALIZADOR

Hemos hecho el analizador en JavaCC para poder hacer uso de las estructuras de datos.

Las tareas que realiza el analizador son:

- a) Encontrar el elemento RUN con mayor número de líneas.
- b) Listar los path de destino de los comandos COPY.
- c) Listar los path de destino de los comandos ADD.
- d) Encontrar el comando COPY con más archivos de origen.
- e) Contar los constructores de cada tipo utilizados.

Para ejecutar, escribir los siguientes comandos en orden:

- 1) javacc TrabajoDockerfile.jj
- 2) javac \*.java
- 3) java -cp . Dockerfile < ejemplos.txt

Los ficheros de ejemplo son: dockerfile.txt, ejemplo1.txt y ejemplo2.txt.

## BIBLIOGRAFÍA

Dockerfile reference. (s,f). En Docker Docs. Recuperado el 27 de diciembre de 2017 de <https://docs.docker.com/engine/reference/builder/#format>

Docker (software). (s,f). En Wikipedia. Recuperado el 27 de diciembre de 2017 de [https://es.wikipedia.org/wiki/Docker\\_\(software\)](https://es.wikipedia.org/wiki/Docker_(software))

Staken, Kimbro. Dockerfile-examples. (s,f). En GitHub. Recuperado el 7 de Enero de 2018 de <https://github.com/kstaken/dockerfile-examples>

Dockerfile examples. (s,f). En Google Images. Recuperado el 4 de Enero de 2018 de [https://www.google.es/search?q=dockerfile+examples&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiom-ezhM3YAhWqCsAKHS5CCfkQ\\_AUICigB&biw=1680&bih=901](https://www.google.es/search?q=dockerfile+examples&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiom-ezhM3YAhWqCsAKHS5CCfkQ_AUICigB&biw=1680&bih=901)