

# Configuración y arranque del sistema

## Contents

Requisitos previos .....	2
Instalación de JAVA .....	2
Preparación del entorno .....	2
Código del sistema .....	6

## Requisitos previos

- Un servidor Ubuntu 16.04 de 64 bits.
- Un usuario sudo que no sea root.

\*Todo se debe instalar en el directorio raíz del servidor

## Instalación de JAVA

Instalaremos el JDK de Oracle, que es la versión oficial distribuida por Oracle, En primer lugar, agregamos el PPA de Oracle, y después, actualizaremos el repositorio de paquetes.

```
sudo add-apt-repository ppa:webupd8team/java
```

```
sudo apt-get update
```

La versión del JDK que debemos instalar es la 8 por lo tanto:

```
sudo apt-get install oracle-java8-installer
```

## Preparación del entorno

En primera instancia necesitaremos el algoritmo de reconocimiento de imagen, para esto:

```
git clone
```

```
https://github.com/ComputerVisionCentre/RRC2015\_Baseline\_CV3Tess/blob/master/main.cpp
```

Para poder compilar el código y conseguir un ejecutable hay que seguir una serie de pasos. Si alguno de estos se configura mal o deriva en algún error, el algoritmo no funcionara. El código ha sido compilado en un sistema Operativo Ubuntu 16 ya que, pese a que todo el proyecto ha sido desarrollado en Windows 10, los componentes necesarios para la

compilación daban muchos errores, por lo que se ha optado por trabajar con el algoritmo desde Linux.

Lo primero debe ser instalar correctamente Tesseract OCR. Se trata de un motor de OCR con soporte para Unicode y la capacidad de reconocer más de 100 idiomas de forma inmediata. Puede ser entrenado para reconocer otros idiomas. Para descargarnos Tesseract con todos los paquetes de idiomas ejecutaremos el siguiente comando:

**sudo apt-get install tesseract-ocr-all**

En segundo lugar, deberemos instalar Leptonica. Se trata de una serie de herramientas, de código abierto que son muy útiles para el procesamiento de imágenes y aplicaciones de análisis de imágenes:

Uso de autoconf para instalar leptónica una vez descargado :

Ejecute **./configure** en este directorio para

compilar Makefiles aquí y en src. Autoconf maneja el

siguiendo automáticamente:

- \* arquitectura endianness

- \* habilitar las funciones de lectura / escritura de imagen de E / S de Leptonica que

  - depende de bibliotecas externas (si las bibliotecas existen)

- \* funciones de habilitación para redirigir la transmisión de imágenes formateadas

  - I / O a la memoria (solo en Linux)

Después de ejecutar **./configure**: **make**;

Para terminar y comprobar que todo este correcto ejecutar el comando **make check**

En tercer lugar, deberemos descargar la librería OpenCV. OpenCV es una biblioteca libre de visión por computador que fue diseñada para la eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real.

Una vez tengamos todos estos componentes, será el momento de construir e instalar OpenCV. Para realizar dicha tarea utilizaremos la herramienta CMake. Debemos tener en cuenta que el algoritmo utiliza los módulos extra de la librería OpenCV, llamados **opencv\_contrib** por lo que también deberemos descargarnos dichos módulos e incluirlos en el ensamblado. Es muy importante que para el empaquetado de la librería utilicemos correctamente los parámetros de entrada en CMake. Los pasos a seguir serán los siguientes:

Actualizamos Ubuntu:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Instalamos dependencias:

```
sudo apt-get install build-essential cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev
```

```
sudo apt-get install python3.5-dev python3-numpy libtbb2 libtbb-dev
```

```
sudo apt-get install libjpeg-dev libpng-dev libtiff5-dev libjasper-dev libdc1394-22-dev libeigen3-dev libtheora-dev libvorbis-dev libxvidcore-dev libx264-dev sphinx-common libtbb-dev yasm libfaac-dev libopencore-amrnb-dev libopencore-amrwb-dev libopenexr-dev libgstreamer-plugins-base1.0-dev libavutil-dev libavfilter-dev libavresample-dev
```

Obtenemos OpenCV

```
sudo -s
```

```
cd /opt
```

```
/opt$ git clone https://github.com/Itseez/opencv.git
```

```
/opt$ git clone https://github.com/Itseez/opencv_contrib.git
```

Construimos e instalamos OpenCV

```
/opt$ cd opencv
```

```
/opt/opencv$ mkdir release
```

```
/opt/opencv$ cd release
```

```
/opt/opencv/release$ cmake -D BUILD_TIFF=ON -D  
WITH_CUDA=OFF -D ENABLE_AVX=OFF -D WITH_OPENGL=OFF -D  
WITH_OPENCL=OFF -D WITH_IPP=OFF -D WITH_TBB=ON -D  
BUILD_TBB=ON -D WITH_EIGEN=OFF -D WITH_V4L=OFF -D  
WITH_VTK=OFF -D BUILD_TESTS=OFF -D BUILD_PERF_TESTS=OFF -  
D CMAKE_BUILD_TYPE=RELEASE -D  
CMAKE_INSTALL_PREFIX=/usr/local -D  
OPENCV_EXTRA_MODULES_PATH=/opt/opencv_contrib/modules  
/opt/opencv/
```

```
/opt/opencv/release$ make -j4
```

```
/opt/opencv/release$ make install
```

```
/opt/opencv/release$ ldconfig
```

```
/opt/opencv/release$ exit
```

```
/opt/opencv/release$ cd ~
```

Para comprobar que todo ha ido bien haremos lo siguiente:

```
pkg-config --modversion opencv
```

Esto debería devolvernos el número de versión de la librería opencv instalada.

## Código del sistema

Pondremos en nuestro servidor el código necesario para que el sistema funcione correctamente. Para esto deberemos descargarlo del siguiente repositorio:

```
git clone https://github.com/Shooter9/TFG.git
```

Una vez hecho esto el Sistema estará listo para ser utilizado.

