

---

# **Annotation Tool Documentation**

***Release 0.1***

**Juan Marín**

May 26, 2016



CONTENTS

<b>1</b>	<b>Contenidos</b>	<b>3</b>
1.1	Instalación . . . . .	3
1.2	Ejecutando la aplicación . . . . .	4
1.3	Importando imágenes . . . . .	6
<b>2</b>	<b>Indices and tables</b>	<b>9</b>



Annotation tool es una aplicación que esta pensada para etiquetar imagenes de forma rapida. La información de las imagenes y sus etiquetas estan guardadas en una base de datos [SQLite](#) . La aplicación incluye scripts para exportar las etiquetas y los datasets. Además, esta diseñada para trabajar con imagenes de la camara [Narrative Clip](#) con lo cual incluye una serie de scripts para ser usados con [Matlab](#) que son ejecutados desde la misma consola de linux en la que esta el servidor funcionando.



## CONTENIDOS

## 1.1 Instalación

### 1.1.1 Instalando la aplicación desde Github

La aplicación esta alojada en [Github](https://github.com/hermetico/TFG.git) con lo cual solo necesitarás clonar el repositorio haciendo:

```
$ git clone http://github.com/hermetico/TFG.git
```

Esto te deberia crear una estructura similar a la siguiente:

```
TFG/  
  |- annotation-tool/  
    |- app/  
    |- manage.py  
    ...  
  |- matlab-scripts/  
  |- sphinx/  
  |- docs/  
  |- log/  
  ...
```

Esto automaticamente te creara una carpeta llamada TFG con el contenido de la web dentro, incluida la documentacion. Si no tienes git instalado tan solo tienes que lanzar el siguiente comando:

```
$ sudo apt-get install git
```

Una vez descargado tendras que instalar las dependencias. Es recomendable que utilices virtualenv. Puedes instalar virtualenv con el siguiente comando:

```
$ pip install virtualenv
```

Ahora nos vamos a la raiz de la web e iniciamos un virtualenv para instalara ahi todos los modulos python que necesitamos en sus correspondientes versiones:

```
$ cd TFG/annotation-tool  
$ virtualenv venv  
$ source venv/bin/activate
```

Esto te creara el virtualenv en TFG/annotation-tool/venv aunque no es necesario que lo pongas ahi, lo puedes poner directamente en la raiz del proyecto.

**Note:** Si no utilizas virtualenv, los paquetes se instalaran en el python path que tengas por defecto pudiendo causar conflicto de paquetes y versiones

Con esto deberíamos ver `(venv)` en la consola, justo en la línea que escribimos los comandos. Para desactivar el `enviroment` basta con ejecutar el comando `deactivate`.

```
(venv)hermetico@socrates:~$
```

Ahora podemos instalar los paquetes que se encuentran listados en el archivo `requirements.txt` con el siguiente comando:

```
$ pip install -r requirements.txt
```

Para instalar un paquete cualquiera bastaría con hacer `pip install Flask` y con esto nos instalaba el paquete `Flask` sin necesidad de permisos de super usuario. Estos paquetes se instalan dentro del `virtualenv` que tenemos activado. Si en ese momento no está activado el `virtualenv`, `pip` lo instalara en la ubicación principal de `python`. Utilizando `-r requirements.txt` estamos diciendo a `pip` que nos instale los paquetes especificados en el archivo `requirements.txt`, que además incluye las versiones de cada paquete con lo cual el `enviroment` generado será exactamente el mismo.

**Note:** Si en algún momento necesitas actualizar el archivo `requirements.txt` bastará con ejecutar el siguiente comando:

```
$ pip freeze > requirements.txt
```

Más información de `virtualenv` y de `pip`

### 1.1.2 Instalando a partir de una copia existente

Si ya tenías la aplicación, puedes optar por realizar una copia de la misma a partir del directorio raíz. A partir de aquí, elimina la carpeta del `virtualenv` y vuelve a realizar los pasos de configuración del `virtualenv`.

## 1.2 Ejecutando la aplicación

### 1.2.1 Probando la instalación

Ejecutaremos el servidor de pruebas para ver que todo funciona correctamente, para ello solo tienes que ir a la raíz de la aplicación (`TFG/annotation-tool`) y ejecutar el servidor con los siguientes comandos:

```
$ python manage.py runserver
```

Antes de ejecutarlo asegúrate de que tienes el `venv` activado. Luego deberás ver el siguiente mensaje:

```
Importing environment from .env...
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Si sale un mensaje parecido es que la instalación está bien, por ahora no hace falta que abras el navegador, antes debemos realizar una serie de configuraciones.



**Warning:** Si te sale un error como el siguiente:

```
import hmac File "...../anaconda/lib/python2.7/hmac.py", line 8, in <module>
from operator import _compare_digest as compare_digestImportError: cannot import name _compare_dige
```

Es que probablemente estes usando python 2.7.6 que viene con anaconda. En la version 2.7.9 de python este problema no aparece, y el unico cambio que contiene el modulo es que esa linea ya no esta incluida. La manera rapida de solucionar este error sin necesidad de actualizar la version de python, que podría afectar otros programas que tengas en el ordenador, es copiar el modulo de anaconda a nuestro virtualenv y eliminar la linea afectada, asi al cargar el modulo nos cargara nuestra version modificada:

```
$ cp ~/anaconda/lib/python2.7/hmac.py venv/lib/python2.7/hmac.py
```

Y con nuestro editor de texto eliminamos la linea 8 que contiene:

```
from operator import _compare_digest as compare_digestImportError
```

**Note:** Si partes de una copia, en este paso abriendo el navegador tendrias una copia exacta y funcional de la version de la cual copiaste.

Ahora puedes pulsar CTRL+C para parar el servidor y configurarlo.

## 1.2.2 Puesta a punto

Tanto si descargas la aplicacion desde github como si haces una copia de una versión existente y quieres trabajar desde cero con tus imagenes, es preferible eliminar las bases de datos que puedan haber y vaciar las carpetas de imagenes.

Las bases de datos estan en la raiz de la y tienen la extension `.sqlite`. La base de datos de producción se llama `data.sqlite` (duh). Con que elimines esa ya será suficiente. Además, sera necesario vaciar la carpeta de imagenes.:

```
$ rm data.sqlite
$ rm -r app/static/media
```

**Note:** En este paso se pedirán datos para que la aplicación funcione correctamente, esto es roles: administrador y usuario. Se creará un usuario administrador y a parte se pedirá dar nombre a una etiqueta por defecto, que sera la que se asignara a todas las imagenes importadas.

El ultimo paso será opcional, se pedirá si se quiere crear un conjunto de etiquetas, definidas en el archivo `tools/deploy.py`. En la parte superior del archivo encontraras un a lista:

```
# modifica esta lista de etiquetas a tu gusto para realizar el deploy
DEFAULT_LABELS = ['etiqueta1, etiqueta2, blablabla]
```

Puedes modificar esa lista para tener tus etiquetas a la hora de hacer el deploy.

Piensa que puedes añadir mas adelante desde la propia aplicación.

Para realizar las configuraciones iniciales para la aplicación tan solo tienes que ejecutar el siguiente comando en la raiz de la aplicación (recuerda: `TFG/annotation-tool/`):

```
$ python manage.py deploy
```

Sigue los pasos y ya tendras la aplicación *casi* lista para ser completamente funcional. Quedará importar imagenes. De todas formas, ejecuta el servidor y ya podras usar la web con el nombre de usuario y contraseña que indicaste en el deploy.

## 1.3 Importando imágenes

### 1.3.1 Pasos previos

Para importar imagenes primero has de crear un usuario a nombre del cual iran asignadas dentro de la base de datos. Si solo vas a importar imagenes de una persona, puedes hacerlo sobre el usuario administrador.

### 1.3.2 Importar imagenes

Las imagenes que van a ser importadas han de estar en la carpeta TFG/annotation-tool/import-bucket y puesto que originalmente es para trabajar con la imagenes de la camara Clip narrative, las imagenes han de mantener la siguiente estructura:

```
id/year/month/day/time.jpg
```

Como las imagenes de la camara clip narrative ya nos dan la estructura de year/month/day/time.jpg lo unico que tenemos que hacer es copiar las imagenes que queremos importar dentro de la carpeta import-bucket/<id> donde id sera la id del usuario al cual asociaremos las imagenes.

Una vez hemos copiado las imagenes que queremos importar, lanzaremos el script de importacion:

```
$ cd TFG/annotation-tool
$ python manage.py import_pictures
```

Este script ira añadiendo las imagenes a la base de datos, les asociará el id especificado en la carpeta y además les asociará la etiqueta por defecto especificada en el deploy.

Cuando el script termine verás que ya no esta la carpeta que creaste, las imagenes se habran movido a la carpeta TFG/annotation-tool/appstatic/media/

### Metodo alternativo de importación

Para utilizar este metodo la estructura de la carpeta a importar debe de obedecer al siguiente patron:

```
id/fecha/nombre.jpg
```

Las imagenes se deben encontrar en la misma carpeta que con el método anterior. Para lanzar el script de importación utiliza el siguiente comando:

```
$ python manage.py import_pictures_simplified
```

**Note:** En este caso no se utilizara hora para almacenar las imagenes. La fecha debe estar en formato iso 8601, que se representa de la siguiente manera YYYY-MM-DD. Lo cual quiere decir que primero va representado el año con 4 digitos, luego el mes con dos digitos y finalmente el día también con 2 digitos. La siguiente es una muestra de una ruta valida:

```
1/2016-06-01/picture.jpg
```

De esta forma tenemos una estructura similar a la anterior pero mas sencilla de mantener.

---

### 1.3.3 Tratamiento de imagenes previo a la importación

Es posible que quieras tratar las imagenes antes de importarlas al sistema, para ello se han incluido dos scripts, uno que hace practicamente todos los pasos necesarios : rotar, cropar y redimensionar. Y un segundo script que tan sólo

redimensiona las imagenes.

## Rotando, cropando y redimensionando

**Warning:** Para importar imagenes de la camara clip narrative, se ha incluido un script que hace uso de librerias matlab. Asegurate de tener instalado matlab antes de ejecutar el script.

Ves a la carpeta `TFG/matlab-scripts` en la cual encontraras un archivo llamado `main.sh` el cual funciona de la siguiente manera:

```
$ ./main.sh <path-carpeta-origen> <path-carpeta-destino>
```

Este script hará un rotate, crop y un resize de las imagenes que contenga la carpeta origen y guardará el resultado en la carpeta destino manteniendo su estructura interna. Además, te ira informando por pantalla de los comandos que va ejecutando.

**Note:** La camara narrative clip mantiene la siguiente estructura de carpetas `year/month/day/time.jpg`. Este script espera encontrar la estructura `name/year/month/day/time.jpg` dentro de la carpeta origen indicada.

Dentro de la carpeta `TFG/matlab-scripts/matlab/` se guardarán los logs de cada proceso matlab ejecutado.

Puedes indicar que la carpeta destino sea la propia `import-bucket` así luego tan solo tendras que modificar la id del usuario

---

## Solo redimensionando

El segundo script se encuentra en la carpeta `TFG/scripts` y se llama `resize-images.sh`. Este script tan solo redimensiona las imagenes a un tamaño especificado, para ello hace uso de la libreria `imagemagick`, en concreto de un script llamado `mogrify` ejecuta `$ man mogrify` para asegurarte de que lo tienes instalado. El script funciona de la siguiente manera:

```
$ resize-images.sh <path-carpeta>
```

Como en el caso anterior, ira mostrando por pantalla los comandos que se van ejecutando.

**Note:** Al igual que el script anterior, espera una estructura de carpetas del tipo `name/year/month/day/time.jpg`

En este caso, puedes ejecutar el script sobre la propia carpeta `import-bucket` para redimensionar todas las imagenes que haya dentro.

Actualmente el tamaño especificado es de `256x256` pero si quieres lo puedes modificar sobre el propio script en la variable `SIZE="256x256"`

---



## INDICES AND TABLES

- `genindex`
- `search`