

Paso 1:

Para instalar Docker en Windows, tenemos que ir al siguiente enlace:

<https://docs.docker.com/desktop/setup/install/windows-install/>

Descargamos esta versión:

Docker Desktop for Windows - x86_64

Ejecutamos el exe. Si en algún momento nos da a elegir, elegiremos WSL 2 en vez de Hyper-V, pero aparentemente por defecto ya siempre funciona con WSL 2. Nos hará hacer log out y log in cuando termine la instalación.

Al volver a iniciar sesión nos saltará con la ventana de Docker.

Aceptamos los términos y condiciones.

No hace falta que pongamos correo, le damos a skip.

Paso 2:

Nos aseguramos de que Docker está ejecutándose en segundo plano. Podemos ver los programas ejecutándose en segundo plano en la barra de tareas, abajo del todo a la derecha. Pulsamos la flechita desplegable hacia arriba y nos saldrá el icono de Docker (una ballena).

Paso 3:

Aquí estoy dando la versión sencilla usando archivos bat. Si esto no funcionase, habrá que recurrir al archivo extenso de instalación. Los archivos bat están asumiendo que existe carpeta de usuario, imagino que, aunque sean ordenadores de la universidad, tendrán la típica estructura de carpetas con usuario (por ejemplo, C:\Users\yanco). Entonces la construcción de las carpetas se genera ahí, en la carpeta de usuario, automáticamente.

Lo que vamos a hacer ahora es doble clic sobre el archivo bat llamado **1_label-studio.bat**. Nos abrirá una ventana de la PowerShell que tenemos que dejar abierta. La podemos minimizar tras ver que no ha dado error. Esto hará *pull* de la imagen de Label Studio, luego generará una serie de carpetas dentro del usuario/mydata (para la persistencia de datos); y, por último, dejará el contenedor/programa ejecutándose. Para poder ver el programa, hay que abrir el navegador y meterse en la siguiente dirección localhost (cualquiera de las dos debería funcionar):

<http://0.0.0.0:8080/>

`http://localhost:8080/`

Ahora, dentro de la aplicación, nos hará iniciar sesión o crearnos una. Vamos a crear una nueva cuenta con la siguiente información:

Correo: taller@gmail.com

Contraseña: 1234abcd

Estas credenciales son para el taller, pero luego la gente en su casa que se ponga las credenciales que quiera. Está todo gestionado localmente, esa cuenta no va online a nadie, y solo sirve para gestionar el usuario dentro de la aplicación.

Cuando comprobemos que funciona, podemos pasar al siguiente paso, no hace falta cerrar esa terminal de PowerShell aún.

Paso 4:

Lo primero, necesitamos tener instalado git en el sistema. Para ello, nos iremos a la siguiente dirección:

<https://git-scm.com/install/windows>

Nos descargamos el primero que sale, “Click here to download”, en la pestaña de Windows. Le daremos al archivo de instalación. Todas las opciones por defecto están correctas, le damos a todo que sí. Ya tenemos git en nuestro sistema.

Paso 5:

Tenemos que ejecutar el archivo **2_ML-backend.bat**. Esto descargará en la carpeta del usuario la carpeta llamada **label-studio-ml-backend**. Cuando termine, lo podemos correr una segunda vez y nos confirmará que ya está instalado y nos dirá dónde.

Paso 6:

Ahora, tenemos que ir Label Studio y clicar a las rayitas de arriba a la izquierda. Vamos a **Organization**. Aquí, arriba a la derecha está la opción **API Tokens Settings**. Desactivamos Personal **Access Tokens**, y activamos **Legacy Tokens**.

A continuación, pulsamos arriba a la derecha, en nuestro perfil, y le pulsamos en **Account & Settings**. Clicamos en Legacy Token, y copiamos el Access Token (un código de 40 caracteres). Tiene este aspecto:

2e83ffff1a522a73db30695affd8ec907cce6a05

Lo que tenemos que hacer ahora es copiar ese código de 40 caracteres, abrir el archivo `env.txt` que adjunto, y pegar el código en las variables, donde pone “PEGARAQUÍ”, son dos sitios. Entonces tendrá este aspecto:

```
LABEL_STUDIO_HOST=http://host.docker.internal:8080
```

```
LABEL_STUDIO_API_KEY=2e83ffff1a522a73db30695affd8ec907cce6a05
```

```
LABEL_STUDIO_ACCESS_TOKEN=2e83ffff1a522a73db30695affd8ec907cce6a05
```

Cuando este archivo ya tenga el código pegado en ambos sitios, activamos con doble clic el archivo **3_tesseract-env.bat**. Este archivo llevará el archivo **env.txt** (convertido a **.env** automáticamente) y el archivo **docker-compose.yml** a la ubicación en que tiene que estar (`C:\Users\yanco\label-studio-ml-backend\label_studio_ml\examples\tesseract`).

Paso 7:

Tenemos que seguir con la ballena en segundo plano (Docker). Ahora podemos hacer doble clic en **4_tesseract.bat**. Esto abrirá una nueva terminal donde se ejecutará tesseract. Hay que mantenerla abierta porque está conectando en el puerto 9090 el modelo para que Label Studio, que es el puerto 8080, lo lea (ambos sirven).

```
http://0.0.0.0:9090/
```

```
http://localhost:9090/
```

Nos metemos con el navegador en esa dirección. Si sale esto:

```
model_class"BBOXOCR"
```

```
status      "UP"
```

Es que todo está correcto. Los logs que salgan en la terminal de Tesseract también habrán dicho que está todo correcto. Ya podemos detener todas las terminales con **control c** (spam hasta que se corte el proceso), y luego ya se pueden cerrar si no se han cerrado solas. Ya hemos comprobado que todo funciona. Cuando ya se ha hecho todo esto al menos una vez, las siguientes veces basta con doble clic sobre el archivo **1_label-studio.bat** para iniciar la aplicación. Si queremos habilitar el *backend* con Tesseract, haremos doble clic en **4_tesseract.bat**.

Para el taller, recomiendo conservar todos los archivos adjuntos en cualquier carpeta, por ejemplo, en la carpeta de usuario, dentro de una carpeta llamada “Taller_Label_Studio” (o en el escritorio, donde sea más cómodo).