

Capítulo 5

Integracion Dron Tello en Kibotics

En este capítulo se describe el desarrollo realizado para poder programar el dron Tello. A diferencia del Mbot, donde el proceso era similar para cualquier sistema operativo, en este caso el proceso va a variar.

5.1. Características del dron Tello

Tello es un mini dron distribuido por la empresa Ryze Technology (Figura 5.1). Debido a la facilidad de su uso, está destinado tanto a niños como a adultos que quieran aprender a utilizar drones. Puede alcanzar una velocidad de hasta 28 km/h, su radio de control abarca hasta los 100 metros y su señal de vídeo se transmite a una frecuencia de 2,4Ghz. Posee una unidad procesadora de Intel, un barómetro para controlar de altura, motores de tipo escobilla y una cámara de 720p. Además, sus hélices están protegidas y es resistente a las caídas.

Entre sus puntos fuertes destacan su gran estabilidad durante el vuelo y que utiliza un sistema de posicionamiento por visión (VPS). El sistema VPS utiliza un mínimo de dos cámaras para medir la distancia al suelo y la posición a la que se encuentra, compensando así los cambios en la posición que puedan darse (Castro, 2019).

5.2. Requisitos previos para su uso

Para el Mbot no se requería ninguna instalación y, además, el proceso de envío del programa era el mismo para los diferentes sistemas operativos. En el caso del Tello, existen peculiaridades para los diferentes sistemas operativos al no tener control sobre la controladora que posee el robot y al necesitar un proxy intermediario

Para MacOS y Windows, es indispensable que el usuario tenga instalado Python en su versión 2.7 y los drivers necesarios para el uso del Tello. La empresa Dji-sdk posee un repositorio en GitHub (<https://github.com/dji-sdk/Tello-Python>) en el que se facilitan instaladores para su uso

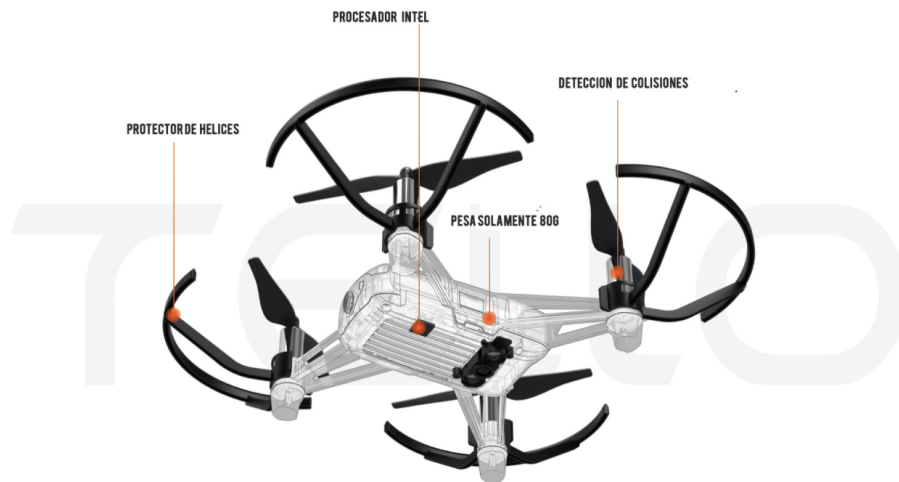


Figura 5.1: Esquema del dron Tello

en los diferentes sistemas operativos.

En Linux, sin embargo, no es necesaria ninguna instalación previa para poder utilizar este dron. Esto es gracias a PyInstaller, una librería de Python, y a que el servidor de Kibotics se encuentra en una máquina Linux. En las próximas secciones se profundizará en este aspecto.

5.3. Proceso de envío desde la plataforma Kibotics

Al igual que el Mbot, en el dron Tello también existe la posibilidad de seleccionar la unidad dedicada a su programación en Python o en Blockly.

Una vez que el usuario ha desarrollado el programa, deberá pulsar el botón azul llamado “Ejecutar en Tello” (Figura 5.2) para iniciar el proceso de envío, que variará dependiendo del sistema operativo utilizado.

5.3.1. Proceso de envío para Linux

1. Se descargará un fichero ejecutable.
2. Encender el dron Tello y conectarse a su punto de acceso WiFi (Figura 5.3).

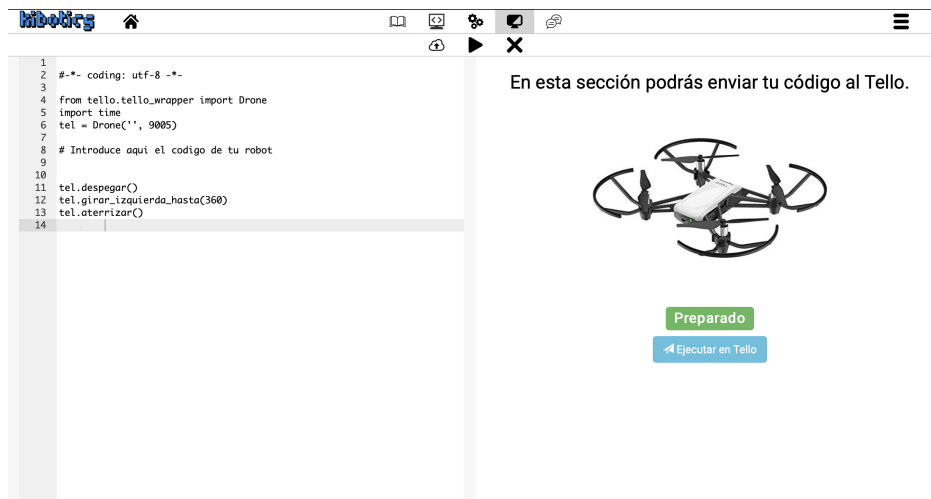


Figura 5.2: Editor para programar el dron Tello en Python



Figura 5.3: Conectarse al punto de acceso WiFi emitido por el dron

3. Dirigirse al directorio donde se descargó el ejecutable, dar los permisos de ejecución y ejecutarlo con los comandos que se ven en el fragmento 5.1.

```

chmod +x tello_code &&
./tello_code

```

Fragmento 5.1: Comandos para ejecución del envío en Linux

4. Una vez ejecutado, el programa se enviará al dron Tello.

5.3.2. Proceso de envío para Windows

1. Se descargará un fichero comprimido, que debe ser extraído.
2. Encender el dron Tello y conectarse a su punto de acceso WiFi (Figura 5.3)
3. Dirigirse al directorio donde se descomprimió el fichero (Figura 5.4)

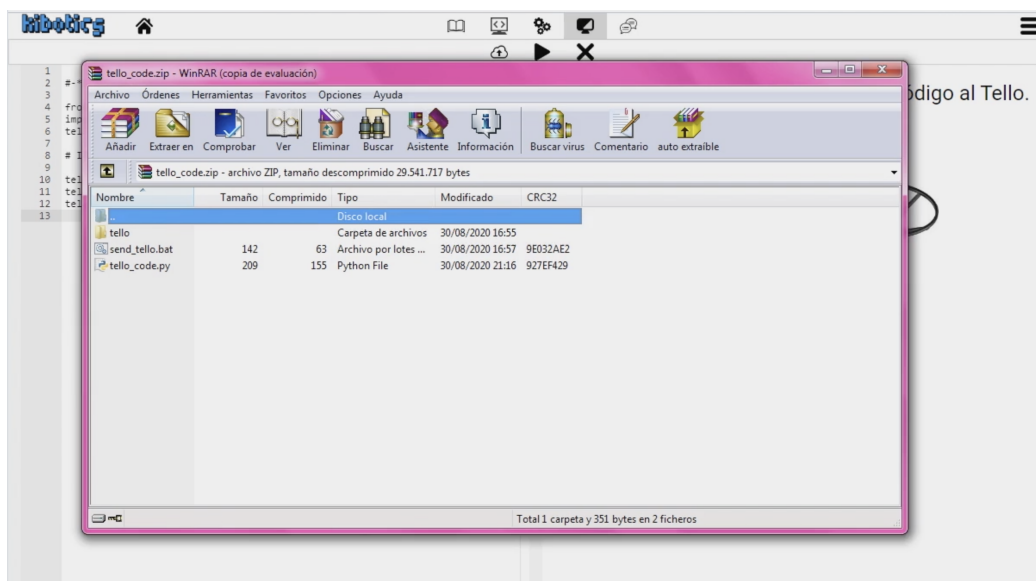


Figura 5.4: Descompresión en Windows

4. Haciendo doble “click” en el ejecutable send_tello.bat, el programa será enviado al Tello.

5.3.3. Proceso de envío para MacOS

1. Se descargará un fichero comprimido, que debe ser extraído.
2. Encender el dron Tello y conectarse a su punto de acceso WiFi (Figura 5.3).
3. Dirigirse al directorio donde se descomprimió el fichero.
4. Dentro del directorio se encuentra un archivo ejecutable llamado send_code.sh, al que tendremos que dar permisos de ejecución y, posteriormente, ejecutarlo para enviar el programa

al dron.

```
chmod +x send_code.sh &&  
./send_code.sh
```

Fragmento 5.2: Comandos para ejecución del envío en MacOS

5.4. Desarrollo realizado para la integración

En la Figura 5.5, se muestra cuál es la arquitectura necesaria para conseguir el envío del programa al dron Tello.



Figura 5.5: Arquitectura del desarrollo para el Tello

A continuación hablaremos en detalle sobre los pasos necesarios para conseguir la integración del Tello en Kibotics en los sistemas operativos Linux, Windows y MacOS.

5.4.1. Envío del programa al servidor

Este paso es similar al del Mbot. Una vez que el usuario ha realizado el programa y ha pulsado en el botón “Ejecutar en Tello” (Figura 5.2), se extrae del editor el código escrito y se envía desde el lado del navegador una petición al servidor de Kibotics con el programa realizado.

```
function send_code_tello() {
```

```

var editor = ace.edit("ace");
let code = editor.getValue();
console.log(code);
const message = {
  method: "GET"
};
var url = '/get_code_to_tello?python_code=' + JSON.stringify(
  code);
fetch(url, message)
  . . . . .
}

```

Fragmento 5.3: Envío del programa desde el navegador al servidor

Si el programa está escrito en Blockly, también es necesario convertirlo a lenguaje Python.

```

var pythoncode = Blockly.Python.workspaceToCode(editor.ui);

```

Fragmento 5.4: Convertir de Blockly a Python

El servidor se encarga de extraer el programa enviado en la petición.

```

def get_code_to_tello(request):
    . . . . .
    python_code = json.loads(request.GET.get('python_code', None))
    . . . . .

```

Fragmento 5.5: Extracción programa en el servidor

5.4.2. Empaquetado del programa

El dron Tello entiende el lenguaje Python, de manera que no es necesario convertirlo a otro lenguaje. Por lo tanto, una vez extraído el código se procede a su empaquetado, que será diferente en función del sistema operativo utilizado por el usuario.

- **Linux.** Una vez extraído el código, se procede a realizar un ejecutable. Esto es posible gracias a una librería de Python llamada PyInstaller, un módulo que permite empaquetar ficheros Python, generando así un ejecutable e incluyendo dentro del propio ejecutable el intérprete de Python. PyInstaller no permite una compilación cruzada, es decir, si este empaquetado se realiza en una máquina Linux, el ejecutable solo podrá utilizarse en una máquina Linux. Debido a esto, la integración para el dron Tello varía según el sistema operativo; como el servidor de Kibotics está alojado en una máquina Linux, solo los usuarios de Linux se beneficiarán de esta utilidad.

```

def create_executable_with_pyinstaller(exercise_path):

```

```

call(". kibotics-drivers/tello/tello-env/bin/activate;
pyinstaller -F --distpath " +
exercice_path + "dist --workpath " + exercice_path + "
build --specpath " + exercice_path + " --clean " +
exercice_path +
"tello_code.py", shell=True)
. . .

```

Fragmento 5.6: Creación ejecutable con PyInstaller

Tras generar el ejecutable, se introduce en la respuesta que se enviará al navegador.

```

create_executable_with_pyinstaller(exercice_path)
response = FileResponse(open(exercice_path + "dist/tello_code",
'rb'), content_type='application/octet-stream')
shutil.rmtree(exercice_dir + "dist/")
shutil.rmtree(exercice_dir + "build/")
os.remove(exercice_dir + "tello_code.spec")

```

Fragmento 5.7: Respuesta a la petición para Linux

- **Windows y MacOS.** El proceso es similar en ambos, pero diferente al caso anterior. Se empaqueta el programa con el directorio que engloba todas las librerías y las dependencias necesarias y que, además, contiene el ejecutable .bat (en Windows) o .sh (en MacOS) que se encargará del envío. La única diferencia entre ambos, es que al necesitar dependencias específicas por sistema operativo, se utilizan directorios diferentes en el empaquetado.

Una vez realizado el empaquetado, se comprime en formato Zip y se introduce en la respuesta de la petición.

```

if operative_system == 'Mac':
    shutil.move(exercice_dir + "tello_code.py", os.getcwd() +
"/kibotics-drivers/tello/MacOs")
    shutil.make_archive('output-tello-mac', 'zip', os.getcwd()
+ "/kibotics-drivers/tello/MacOs")
    os.remove(os.getcwd() + "/kibotics-drivers/tello/MacOs/
tello_code.py")
    with open('output-tello-mac.zip', 'rb') as f:
        response = HttpResponse(f, content_type=guess_type('
output-tello-mac.zip')[0])
        response['Content-Length'] = len(response.content)
        os.remove('output-tello-mac.zip')

elif operative_system == 'Windows':
    shutil.move(exercice_dir + "tello_code.py", os.getcwd() +
"/kibotics-drivers/tello/Windows")

```

```

shutil.make_archive('output-tello-windows', 'zip', os.
    getcwd() + "/kibotics-drivers/tello/Windows")
os.remove(os.getcwd() + "/kibotics-drivers/tello/Windows/
    tello_code.py")
with open('output-tello-windows.zip', 'rb') as f:
    response = HttpResponse(f, content_type=guess_type('
        output-tello-windows.zip')[0])
    response['Content-Length'] = len(response.content)
    os.remove('output-tello-windows.zip')

```

Fragmento 5.8: Creación empaquetado y respuesta en Windows y MacOS

5.4.3. Descarga del empaquetado y envío al Tello

El navegador recibe la respuesta de la que se extraerá el ejecutable, en el caso de que se use una máquina Linux, o el comprimido, en el caso de usar Windows o MacOS. Posteriormente, se descarga en el ordenador.

```

. . .

var = url = '/get_code_to_tello?python_code=' + JSON.stringify(code)
;
fetch(url, message)
    .then(response => {
        ld.style.display = "none";
        if (response.ok) {
            download_executable(response);
        } else {
            console.error("Bad Response");
        }
    })
    .catch(err => console.error(err));
}

function downloadExecutable(response) {
    response.blob().then(blob => {
        var down = document.createElement("down");
        document.body.appendChild(down);
        down.style = "display: none";
        url = window.URL.createObjectURL(blob);
        down.href = url;
        down.download = 'tello_code';
        down.click();
        window.URL.revokeObjectURL(url)
    })
}

```



```
}
```

Fragmento 5.9: Extracción empaquetado y descarga

Una vez realizada la descarga, se puede realizar el envío al dron Tello, tal y como se explica en la sección 5.3.

El ejecutable que se utiliza en MacOS para el envío (send_code.sh), con el objetivo de facilitar las instalaciones requeridas para el Tello, instalará todas las dependencias necesarias en el caso de que no se tengan en el ordenador.

```
#!/bin/bash

##### Requirements #####

env=~/.virtualenvs/tello-env
if [ -d $env ];
then
    source ~/.virtualenvs/tello-env/bin/activate
else
    if ! type "pip" > /dev/null; then
        sudo easy_install pip
    fi
    if ! type "brew" > /dev/null; then
        /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
        brew update
    fi
    sudo brew install cmake
    sudo brew install boost
    sudo brew install boost-python
    sudo brew install ffmpeg
    sudo brew install tcl-tk

    mkdir ~/.virtualenvs
    pip install virtualenv
    virtualenv ~/.virtualenvs/tello-env --python=python2.7
    source ~/.virtualenvs/tello-env/bin/activate
    pip install SimpleWebSocketServer
    pip install numpy
    pip install matplotlib
    pip install opencv-python==3.1.0.1
fi

echo "#####"
echo "### Sending program to tello ###"
```

```
echo "#####"  
python tello_code.py  
echo "Program finished"  
deactivate
```

Fragmento 5.10: Ejecutable de envío para MacOS

5.5. Resolución y experimentación

La tecnología utilizada en este caso es distinta a la del Mbot debido a las diferencias en las controladoras del robot; con Arduino, el control sobre ella es mayor, mientras que en la que utiliza el Tello no tenemos ningún control.

Otro aspecto importante es que para el envío del programa al Tello se necesita una pasarela, es decir, para poder comunicarnos con él necesitamos conectarnos al punto de acceso WiFi que emite el dron, por lo que no es posible un envío directo desde el navegador, algo que sí se podía hacer en el Mbot con Web Serial.

Por otro lado, la tecnología PyInstaller permite que el usuario interactúe con el dron sin necesidad de instalar previamente nada, y también simplifica el envío al lanzar el ejecutable descargado. El problema que surge es que, con la restricción de compilación cruzada, este proceso solo puede ser factible para Linux, ya que el servidor de Kibotics, que es el encargado de preparar el ejecutable, está alojado en una máquina Linux.

Para Windows y MacOS la situación cambia completamente. El usuario sí necesita tener instalado en su ordenador el intérprete Python2.7 y los drivers para poder usar el Tello, aunque en MacOS el ejecutable de envío será el que lo instale si el usuario no lo tiene, facilitando, por tanto, su uso. Además, el envío del programa no es tan directo, ya que en primer lugar hay que descomprimir el fichero descargado, que tiene las dependencias necesarias para el envío y, después, se lanza el ejecutable.

Para su experimentación se ha probado en diferentes sistemas operativos y ordenadores, verificando así el correcto funcionamiento de la integración. Entre los que han sido probados se encuentran:

- *Ubuntu 16.04*
- *Ubuntu 18.04.*
- *Windows 10.*
- *Windows 7.*
- *MacOs Catalina Versión 10.15.6.*