

# Taller de drones: Guía del estudiante

En este taller vas a aprender algunas cosas básicas sobre el funcionamiento de los drones y sobre el software para controlarlos desde tierra. Para ello te vamos a proporcionar algunas herramientas sencillas que tendrás que instalar en tu ordenador y que te van a permitir experimentar un montón de cosas. Creemos que al acabar el taller te quedarás con ganas de experimentar más (las herramientas te lo permitirán) y, sobre todo, con ganas de aprender más sobre el tema.

El taller está organizado en tres partes. En la primera (Aspectos básicos) veremos cuáles son los componentes típicos de un dron y también algunas cosas básicas sobre software para controlar el dron (lo que llamamos habitualmente estación de tierra).

La segunda parte (Paseo por Mission Planner) consiste en una colección de vídeos cortos que muestran el funcionamiento del programa de estación de tierra más usado en ordenadores Windows, que es Mission Planner. Este programa puede trabajar con un simulador del dron, de manera que instalando unas pocas herramientas en nuestro ordenador podemos hacer un montón de cosas interesantes.

Una de esas cosas interesantes es desarrollar nuestra propia estación de tierra desde la que enviar órdenes al dron (ya sea el dron real o el simulador). En la tercera parte (Pequeña estación de tierra) veremos cómo hacer esto.

## Parte 1: Aspectos básicos

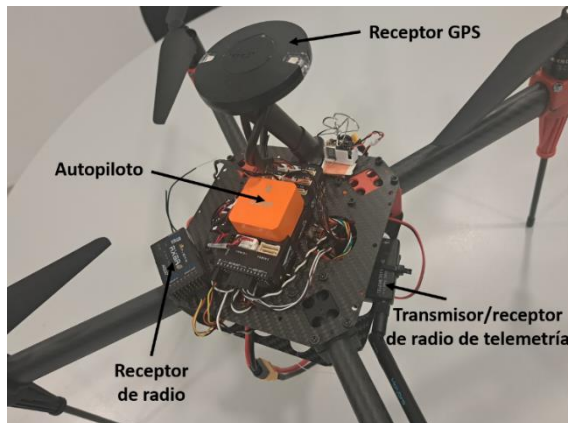
### El dron

El dron que se usa en el taller es un multi-rotor como el mostrado en la figura 1. En concreto se trata de un cuatrimotor (también los hay de seis motores o de ocho). Los multi-rotores tienen algunas ventajas respecto a los drones de ala fija. Por ejemplo, no necesitan una pista de despegue o de aterrizaje y pueden realizar operaciones de precisión como, por ejemplo, detenerse en un punto determinado para hacer una foto o para dejar caer en ese punto un pequeño equipo de primeros auxilios. También tienen sus inconvenientes. Por ejemplo, tienen menos autonomía de vuelo que los drones de ala fija ya que los multi-rotores consumen mucha energía eléctrica para mover los motores mientras que los de ala fija pueden beneficiarse de las ráfagas de viento y de las fuerzas aerodinámicas que generan (por ejemplo, pueden planear con poco gasto de energía).



Figura 1: El dron

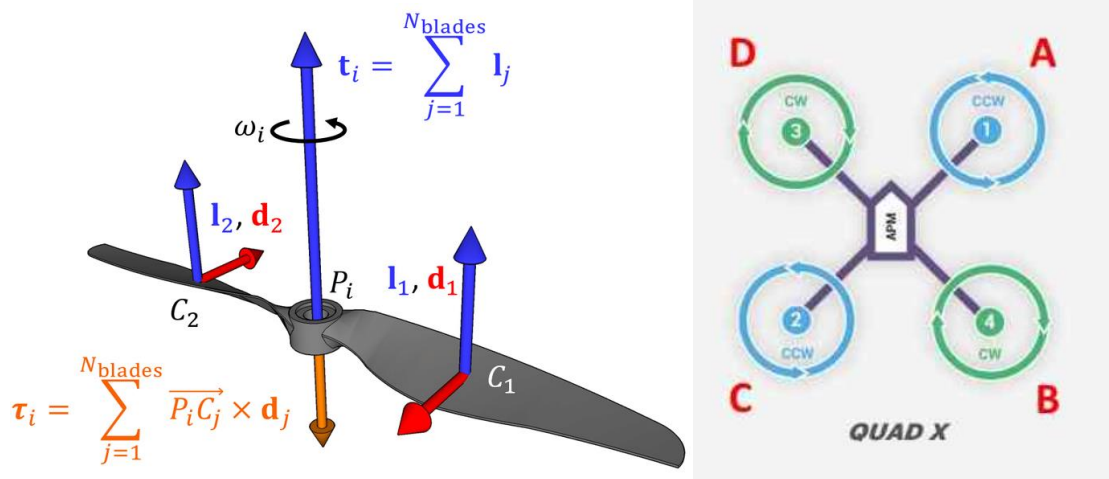
Los componentes principales de nuestro cuatrimotor se muestran en la figura 2.



**Figura 2:** Componentes principales del dron

### Motores y hélices

El giro de los motores y la forma retorcida de las hélices genera las fuerzas de sustentación para que el dron se eleve o se mueva en la dirección deseada. En la figura 3 puede verse que si la hélice gira en sentido anti-horario el aire es impulsado hacia abajo y el dron ascenderá. En un cuatrimotor, el sentido de giro de los 4 motores es el que se indica en la figura, con lo cual queda determinado cómo tienen que ser las hélices que lleva cada motor para que siempre se genere fuerza de sustentación. Si los motores giran muy rápido la fuerza generada será mucha y el dron ascenderá. Si giran más lentamente la fuerza será menor y quedará compensada por el peso del dron, que descenderá.



**Figura 3:** Sentido de giro de las hélices y de los 4 motores del dron

### El autopiloto

Se trata del componente más sofisticado del dron. El autopiloto se encarga de generar las señales adecuadas a los motores para que el dron realice la operación deseada. Por ejemplo, hacer girar los 4 motores a la misma velocidad y rápido (para ascender) o hacer girar más rápido los dos motores traseros para avanzar hacia delante.

El autopiloto tiene integrados varios sensores. Uno de ellos es el barómetro, que le permite estimar la altura a la que se encuentra el dron. También tiene un giroscopio y un magnetómetro que le permiten estimar la orientación que tiene el dron en cada momento. Estos sensores son fundamentales para que el autopiloto pueda hacer una de sus labores más importantes: mantener la estabilidad del dron en el aire. Por ejemplo, gracias a la información proporcionada por el barómetro, el autopiloto puede determinar que la altura del dron ha cambiado sin que así lo haya requerido el piloto (quizá por la acción de una ráfaga de viento) y entonces mover los motores lo necesario para que el dron recupere la altura correcta.

### Receptor de la señal GPS

Gracias a este dispositivo el autopiloto sabe en todo momento en qué posición geográfica (latitud, longitud) está el dron. Esta información también es usada para mantener la estabilidad del dron. Además, permite al autopiloto realizar operaciones muy sofisticadas como, por ejemplo, ejecutar un plan de vuelo especificado en forma de secuencia de puntos a visitar (waypoints), cada uno caracterizado por su latitud y su longitud.

### Receptor de radio

Este dispositivo suministra al autopiloto la información que recibe el dron vía radio desde la emisora de radio que usa el piloto. La figura 4 muestra una emisora de radio típica. La emisora dispone de una variedad de actuadores (palancas, botones, interruptores). Puede configurarse de manera que al manipularse uno de esos actuadores se genere una señal de radio a través de uno de los canales. Cuando el autopiloto recibe esa señal, a través del receptor de radio, puede realizar la operación deseada (por ejemplo, aterrizar el dron) si ha sido configurado para ello.



Figura 4: Configuración típica de la emisora de radio

En una configuración típica, uno de los interruptores se usa para armar/desarmar el dron (una especie de interruptor que hay que accionar para que el dron pueda empezar a funcionar). Las palancas, que pueden moverse en sentido vertical y horizontal, se usan para hacer que el dron ascienda/descienda (aceleración o throttle en inglés), gire sobre sí mismo (giñada), se mueva hacia delante/atrás (cabeceo o pitch) o hacia la izquierda/derecha (alabeo o roll).

### Emisor/receptor de radio de telemetría

Este dispositivo, que trabaja en frecuencias de radio diferentes al anterior, permite conectar al dron con una estación de tierra (por ejemplo, un portátil), que naturalmente también tiene que tener conectado un emisor/receptor. Gracias a este canal de comunicación (que llamamos canal de telemetría), desde la estación de tierra podemos enviar al dron ordenes tales como armar, despegar, desplazarse a un punto dado, etc. También a través de este canal la estación de tierra recibe información del dron, como, por ejemplo, la posición en la que se encuentra o el nivel de batería en todo momento. Este intercambio de información a través del canal de telemetría se realiza de acuerdo al protocolo Mavlink.

En este taller vamos a trabajar con el software de estación de tierra más usado, que se llama Mission Planner. Y también daremos los primeros pasos para la construcción de nuestra propia aplicación de estación de tierra.

## **Mission Planner**

Este software de estación de tierra (desarrollado en lenguaje C#) es gratuito y es el más usado en ordenadores con Windows. La figura 5 muestra una imagen de la pantalla principal.

A través del canal de telemetría, el programa recibe datos del dron que le permiten representar en el mapa la posición del dron en todo momento, el nivel de batería, la altitud, orientación del dron, etc).



**Figura 5:** Pantalla principal de Mission Planner, con diversas opciones

La pestaña de acciones nos presenta una variedad de botones y opciones para enviar instrucciones al dron (como armar, despegar, cambiar velocidad o volar a un punto dado sobre el que se ha clicado en el mapa).

A lo largo de este taller aprenderemos a utilizar esta aplicación para realizar una amplia variedad de operaciones con el dron.

## Simulador SITL

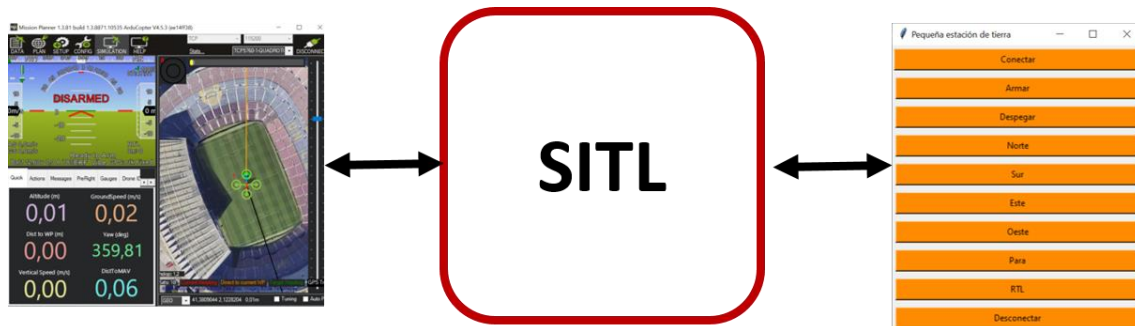
Una de las pestañas de Mission Planner permite poner en marcha un simulador del dron, que llamamos SITL (Software In The Loop). Este simulador es de extraordinaria utilidad porque nos permite verificar el correcto funcionamiento de nuestros desarrollos (por ejemplo, nuestros planes de vuelo) antes de llevarlos a cabo con el dron real. De hecho, todas las actividades de este taller se van a realizar usando este simulador, de forma que el taller se puede hacer íntegramente en casa, para lo cual sólo es necesario instalar en un portátil las (pocas) herramientas que se usarán.

Existe un segundo taller que se realiza de forma presencial en las instalaciones del Campus del Baix Llobregat de la UPC en Castelldefels. En ese taller sí se trabaja con un dron real como el mostrado en la figura 1, con el que se pueden probar algunos de los desarrollos que se hacen en este taller.

## Nuestra pequeña estación de tierra

En la última parte de este taller abordaremos la tarea de programar en Python una sencilla estación de tierra desde la que podamos interactuar con el dron (más concretamente, con el simulador SITL), tal y como muestra la figura 6. Desde nuestra estación de tierra podremos armar el dron, hacerlo despegar y desplazarse. Puesto que Mission Planner también estará conectado al simulador, podremos observar en el mapa los movimientos del dron ordenados desde nuestra pequeña estación de tierra. Esta actividad abrirá las puertas a un gran mundo de posibilidades

para los amantes de la programación (por ejemplo, el uso de inteligencia artificial para controlar el dron o para procesar la información que nos envíe).



**Figura 6:** Interacción entre el simulador SITL, Mission Planner y nuestra pequeña estación de tierra

## Parte 2: Un paseo por Mission Planner

A continuación, tienes una colección de videos cortos, a modo de guía, que te muestran el plan de trabajo que te proponemos para esta segunda parte del taller, en la que vas a aprender a utilizar Mission Planner y el simulador SITL. El material está organizado en 7 pasos que incluyen instalación, vuelos sencillos, misiones e incluso volar un enjambre de drones. También te proponemos algunos retos sencillos (aunque alguno de ellos requerirá un poco de investigación).

### Paso #1: Instalación de Mission Planner

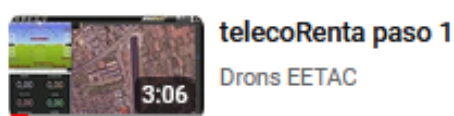
Mission Planner es el software de estación de tierra más utilizado en ordenadores con Windows. Es gratuito (e incluso de código abierto). Puede descargarse desde este enlace:

<https://ardupilot.org/planner/docs/mission-planner-overview.html>

Desde Mission Planner podemos conectarnos al dron real o a un simulador (SITL). Una vez conectados podemos realizar operaciones tales como configurar cualquiera de los múltiples parámetros del autopiloto, dar instrucciones sencillas al dron (como despegar, volar a un punto determinado o aterrizar) o dar instrucciones más complejas (como ejecutar un plan de vuelo especificado en forma de lista de waypoints a visitar).

Mission Planner muestra un mapa de la zona en la que se vuela de manera que puede observarse la respuesta del dron a las órdenes que le enviamos.

En el vídeo siguiente se muestra el proceso de instalación, que es muy sencillo. Es importante no olvidar la instalación de las BETA updates, tal y como muestra el video.



En las versiones más actuales de Mission Planner es posible que esté activa la funcionalidad llamada Altitude Angel, que muestra alertas en el caso de que se intente volar en zonas



restringidas. La alerta se puede mostrar en forma de zona coloreada en rojo, indicando que no es posible volar en esa zona. Obviamente, cuando se trabaja con el simulador conviene desactivar este servicio, cosa que se hace tal y como muestra la figura 7.

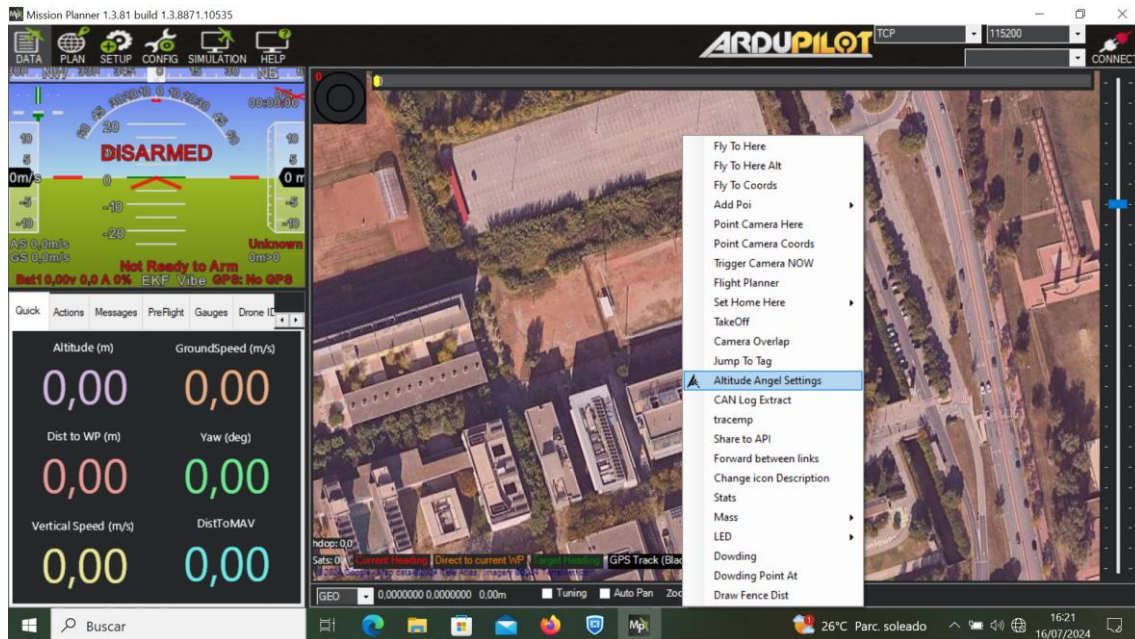


Figura 7: Opción para desactivar el servicio Altitude Angel

## Paso #2: Primer vuelo y un posible problema

En el vídeo siguiente se muestra cómo poner en marcha el simulador SITL desde Mission Planner y como realizar una operación básica con el dron simulado. La operación consiste en armar el dron (equivalente a “encender” los motores), hacerlo despegar y dirigirlo a un punto determinado señalado en el mapa que nos ofrece Mission Planner. El vídeo también muestra un problema que puede producirse, dependiendo de la versión del software y del ordenador en el que se ha instalado. El problema se resuelve en el vídeo del paso siguiente.



telecoRenta paso 2  
Drons EETAC

Retos (Si no has experimentado el problema indicado en el vídeo).

1. El vídeo muestra cómo usar la opción de volar a un punto que marcamos en el mapa. Pero hay una opción para volar a un punto indicándole las coordenadas (lat, lon) de ese punto. Averigua las coordenadas del centro del Anec Blau (famoso centro comercial de Castelldefels) y usa esa opción para volar ahí. Cuando el dron llegue al punto indicado haz que aterrice (tendrás que investigar cómo darle esa orden desde Mission Planner). Después intenta armar de nuevo al dron para volver al DronLab. ¿Por qué no puedes armarlo? ¿Qué debes hacer para poder armarlo?

### Paso #3: Ahora sí: primer vuelo (pero en otro sitio más divertido)

Navegando por el mapa que nos ofrece Mission Planner es posible elegir la zona en la que queremos volar el dron. Eso es lo que se muestra al inicio del vídeo. Después, se muestra cómo resolver el problema indicado en el paso anterior, en el caso de que se haya producido.



telecoRenta paso 3

Drons EETAC

Una vez resuelto el problema, ahora sí que podemos hacer ya el primer vuelo. Para ello, armamos el dron, lo dirigimos a los puntos deseados marcándolos sobre el mapa y finalmente hacemos la operación Return To Launch (RTL). Cuando se le envía esta orden, el dron asciende a una altura prefijada (15 metros en el caso que se muestra en el vídeo) y retorna de manera autónoma al punto del que despegó.

#### Retos

1. Realiza los retos de paso 2 si no pudiste hacerlo en su momento como consecuencia del problema mencionado.
2. Intenta hacer un vuelo en la Plaza del Obradoiro.
3. Imagina que estás volando por una zona donde tienes obstáculos muy altos y necesitas modificar la altura del RTL a 30 metros. Busca dentro de la pestaña CONFIG->Full Parameter List qué parámetro hay que modificar para tener ese comportamiento en tu dron.

### Paso #4: Limitar el área de vuelo

En ocasiones conviene limitar el área de vuelo. Los límites pueden ser físicos, como es el caso de nuestro dronLab, que tiene una red que delimita el volumen del que el dron no va a poder escapar. Pero en otras ocasiones no existen esos límites fijos. Por ejemplo, si volamos en el Nou Camp tenemos que evitar que el dron salga de los límites del terreno de juego y sobrevuele las gradas en las que se encuentra el público. Y ahí no tenemos una red que impida al dron cruzar de una zona a otra.

Afortunadamente, es posible crear una red virtual que denominamos geofence de inclusión, de manera que el autopiloto no va a permitir que el dron salga de la zona delimitada por ese geofence. También es posible definir geofences de exclusión, que son zonas en las que el dron no puede entrar. Los geofences de exclusión son ideales para representar obstáculos en el área de vuelo.

El siguiente vídeo muestra cómo definir esos geofences y cuál es el comportamiento del dron en ese escenario.



telecoRenta paso 4

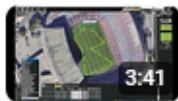
Drons EETAC

## Retos

1. Los geofences que se muestran en el vídeo se definen mediante polígonos. También pueden definirse mediante círculos. Intenta crear un geofence de exclusión que ocupe exactamente el círculo central del Nou Camp.
2. Busca en el mapa de Mission Planner el circuito de Cataluña (en Montmeló). Define un geofence para que el dron no pueda salirse del circuito (necesitarás combinar sabiamente un geofence de inclusión y uno de exclusión). Haz que el dron haga un RTL si intenta salirse del circuito. Intenta dar una vuelta por todo el circuito.

## **Paso #5: Una misión**

El autopiloto es capaz de realizar operaciones muy sofisticadas. Por ejemplo, podemos especificar una misión en forma de secuencia de puntos (waypoints) que el dron tiene que visitar, marcándolos sobre el mapa. Entonces cargamos la misión en el autopiloto y le damos la orden de ejecutar esa misión de forma autónoma. Esa es la operación que se muestra en el vídeo siguiente.



telecoRenta paso 5  
Drons EETAC

## Retos

1. Investiga cómo planificar una misión en la que el dron se detendrá 5 segundos en cada uno de los waypoints del plan de vuelo.
2. Averigua qué efecto tiene el uso del comando `CONDITION_YAW` en uno de los puntos del plan de vuelo.

## **Paso #6: Misión de rastreo**

Con frecuencia la tarea que debe ejecutar el dron consiste en rastrear una determinada zona, por ejemplo, para localizar algún objeto. En ese caso, basta con definir la zona que hay que rastrear y Mission Planner nos propone una misión que rastrea esa zona completamente. El vídeo siguiente te muestra cómo realizar esta operación.



telecoRenta paso 6  
Drons EETAC

## Retos

1. Encuentra una planta fotovoltaica en tu comunidad autónoma y realiza un patrón de rastreo de una parte de las instalaciones fotovoltaica.

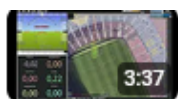


## Paso #7: Un enjambre de drones

Un enjambre de drones no es más que un conjunto de drones que vuelan juntos de manera coordinada (cada vez son más populares los espectáculos basados en enjambres de drones, que muchos llaman fuegos artificiales digitales).

Mission Planner permite trabajar con un enjambre de drones, para lo cual es necesario poner en marcha varias instancias del simulador SITL. Una vez puestos en marcha, Mission Planner nos permite identificar el dron que queremos controlar y dar la orden a ese dron.

El vídeo siguiente muestra cómo trabajar con un enjambre de drones. Naturalmente, lo interesante de esta opción es hacer volar los drones en una misión coordinada. Pero eso ya se escapa del alcance de este taller (aunque te proponemos un reto al respecto). En todo caso, el contenido de la tercera parte del taller, en la que desarrollaremos una pequeña estación de tierra, proporciona las bases para que las personas más interesadas puedan abordar el reto de diseñar misiones para un enjambre de drones.



telecoRenta paso 7  
Drons EETAC

### Retos

1. El video muestra cómo guiar los drones de enjambre un a uno. Mission Planner ofrece la opción de designar a uno de los drones como líder y hacer que todos los demás sigan al líder. Intenta descubrir cómo hacer esto.

## Parte 3: Pequeña estación de tierra

En esta tercera parte vamos a crear nuestra propia (y extremadamente simple) estación de tierra. La idea es sencilla. De igual manera que alguien creo un programa en C# que se llama Mission Planner y que ya conocemos bien, nosotros podemos crear un programa en Python desde el que podamos conectarnos al dron o al simulador y enviarle comandos.

Para poder abordar este reto necesitamos:

- Unos mínimos conocimientos de programación en Python
- Las herramientas necesarias para crear y ejecutar programas en Python
- Una librería con las funciones que nos permitan enviar comandos al dron o al simulador.

### Instalación de nuevas herramientas

Para crear nuestros programas en Python necesitamos instalar un intérprete de Python. Puede descargarse desde esta página (pueden usarse las versiones más actuales):

<https://www.python.org/downloads/>

También necesitamos un entorno de desarrollo. Nuestro favorito es Pycharm, que puede descargarse aquí (asegúrate de instalar la versión denominada Community Edition, que es gratuita y más que suficiente):

<https://www.jetbrains.com/pycharm/>

Finalmente, para descargar el código que vamos a usar necesitas instalarte la herramienta Git, que puedes descargar desde esta página:

<https://git-scm.com/downloads>

## La librería DronLink

Para controlar el dron desde el programa usaremos la librería DronLink. Se trata de una librería desarrollada por el grupo de investigación ICARUS de la UPC, que es muy sencilla de usar. Fíjate en las líneas de código que se necesitan para hacer una operación básica con el dron:

```
# importo la librería
from dronLink.Dron import Dron

# necesito esta librería de sistema
import time

# creo un objeto de la clase Dron para controlar el dron
dron = Dron ()

# me conecto al simulador e indico la velocidad de comunicación
dron.connect ('tcp:127.0.0.1:5763', 115200)

# armo y despego el dron a una altura de 3 metros
dron.arm()
dron.takeOff (3)

# hago volar el dron hacia el norte durante 5 segundos
dron.go ('North')
time.sleep (5)

# hago que el dron regrese al punto de partida y me desconecto
dron.RTL()
dron.disconnect()
```

Recuerda que el simulador es un servidor que acepta hasta 3 conexiones de diferentes programas, en los puertos 5760, 5762 y 5763. Normalmente, Mission Planner se conecta al puerto 5760. El código del ejemplo anterior se conecta al puerto 5763, de manera que el programa puede estar conectado al simulador al mismo tiempo que Mission Planner y podremos ver en el mapa los movimientos del dron, ordenados desde nuestro programa.

Naturalmente la librería tiene muchas más funcionalidades, pero las que se muestran en el ejemplo son suficientes para la construcción de nuestra pequeña estación de tierra. En todo caso, en este enlace puedes encontrar una información más completa sobre las funcionalidades de la librería.

<https://github.com/dronsEETAC/DronLink>

## Interfaz gráfica

La estación de tierra tiene una interfaz gráfica que se muestra en la figura 8 (se muestra superpuesta a la ventana de Mission Planner, en la parte derecha).



Figura 8: Interfaz gráfica de nuestra estación de tierra

No es más que una serie de botones para realizar operaciones básicas. No nos preocupamos aquí de que sea una interfaz visualmente atractiva. Eso queda en tus manos si te interesa la cuestión. Lo importante aquí es que sea funcional.

El código de la estación de tierra es este (disponible en el repositorio del taller):

```
import tkinter as tk
from dronLink.Dron import Dron

dron = Dron()

ventana = tk.Tk()
ventana.geometry('350x400')
ventana.title("Pequeña estación de tierra")

# La interfaz tiene 10 filas y una columna

ventana.rowconfigure(0, weight=1)
ventana.rowconfigure(1, weight=1)
ventana.rowconfigure(2, weight=1)
ventana.rowconfigure(3, weight=1)
ventana.rowconfigure(4, weight=1)
ventana.rowconfigure(5, weight=1)
ventana.rowconfigure(6, weight=1)
ventana.rowconfigure(7, weight=1)
ventana.rowconfigure(8, weight=1)
ventana.rowconfigure(9, weight=1)

ventana.columnconfigure(0, weight=1)

# Disponemos ahora los 9 botones
connectBtn = tk.Button(ventana, text="Conectar", bg="dark orange",

command = lambda: dron.connect('tcp:127.0.0.1:5763', 115200))
connectBtn.grid(row=0, column=0, padx=3, pady=5, sticky=tk.N + tk.S + tk.E + tk.W)
```

```

armBtn = tk.Button(ventana, text="Armar", bg="orange", command=lambda: dron.arm())
armBtn.grid(row=1, column=0, padx=5, pady=5, sticky=tk.N + tk.S + tk.E + tk.W)

takeOffBtn = tk.Button(ventana, text="Despegar", bg="dark", command=lambda: dron.takeOff(3))
takeOffBtn.grid(row=2, column=0, padx=5, pady=5, sticky=tk.N + tk.S + tk.E + tk.W)

NorthBtn = tk.Button(ventana, text="Norte", bg="orange", command=lambda: dron.go('North'))
NorthBtn.grid(row=3, column=0, padx=5, pady=5, sticky=tk.N + tk.S + tk.E + tk.W)

SouthBtn = tk.Button(ventana, text="Sur", bg="orange", command=lambda: dron.go('South'))
SouthBtn.grid(row=4, column=0, padx=5, pady=5, sticky=tk.N + tk.S + tk.E + tk.W)

EastBtn = tk.Button(ventana, text="Este", bg="orange", command=lambda: dron.go('East'))
EastBtn.grid(row=5, column=0, padx=5, pady=5, sticky=tk.N + tk.S + tk.E + tk.W)

WestBtn = tk.Button(ventana, text="Oeste", bg="orange", command=lambda: dron.go('West'))
WestBtn.grid(row=6, column=0, padx=5, pady=5, sticky=tk.N + tk.S + tk.E + tk.W)

StopBtn = tk.Button(ventana, text="Para", bg="orange", command=lambda: dron.go('Stop'))
StopBtn.grid(row=7, column=0, padx=5, pady=5, sticky=tk.N + tk.S + tk.E + tk.W)

RTLBtn = tk.Button(ventana, text="RTL", bg="orange", command=lambda: dron.RTL())
RTLBtn.grid(row=8, column=0, padx=5, pady=5, sticky=tk.N + tk.S + tk.E + tk.W)

disconnectBtn=tk.Button(ventana, text="Desconectar", bg="orange", command=lambda: dron.disconnect())
disconnectBtn.grid(row=9, column=0, padx=5, pady=5, sticky=tk.N + tk.S + tk.E + tk.W)

ventana.mainloop()

```

Para crear la interfaz gráfica se ha usado la librería tkinter. Básicamente se trata de imaginar que la interfaz es una matriz de elementos. En nuestro caso, los elementos son todos botones, que están organizados en una columna de diez filas.

Después se indican las características que tiene cada uno de esos botones. En concreto se indica el texto que debe aparecer en el botón, el color del botón y la función que hay que ejecutar cuando se pulsa el botón. Por ejemplo, cuando se pulsa el botón Conectar se ejecuta la función:

```
dron.connect('tcp:127.0.0.1:5763', 115200))
```

Para cada botón también hay que indicar cómo debe disponerse en la interfaz. En particular, debe especificarse la fila y la columna. También se indica que el botón debe extenderse para ocupar todo el espacio de esa fila.

## Demostración

El código de la estación de tierra y el de la librería DronLink pueden descargarse del repositorio GitHub de este taller. En el vídeo siguiente se muestra cómo poner en marcha el programa, asumiendo que ya se ha instalado Pycharm y el intérprete de Python. Fíjate que es necesario instalar también una librería que se llama pymavlink, que es la que usa DronLink para construir los comandos que se necesitan, en coherencia con el protocolo Mavlink.



**Demo pequeña estación de tierra**

Drons EETAC

## Retos

Sobre la base del código de la estación de tierra puedes ahora plantearte varios retos sencillos. Te proponemos tres, alguno de los cuales requerirá que investigues un poco las posibilidades de las librerías `dronLink` y `tkinter`:

1. Añade un botón para hacer que el dron aterrice (`Land`).
2. Añade los elementos necesarios para que el usuario pueda especificar la altura que debe alcanzar el dron al despegar.
3. Reorganiza los botones con los que se indica la dirección de vuelo para que tengan un aspecto parecido a este:

