

Lección 6:

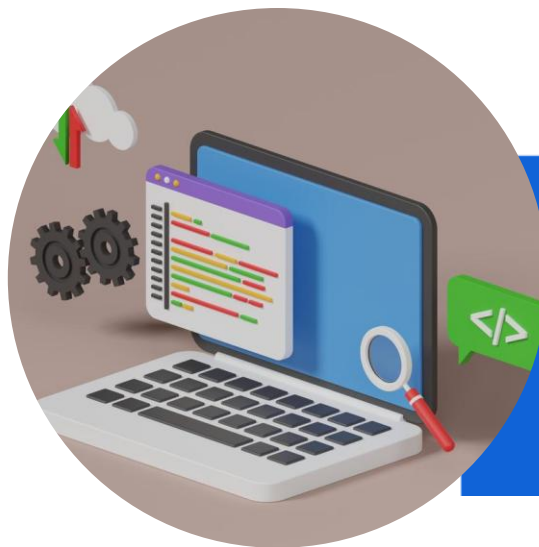
Recopilación de datos con drones

Agricultura inteligente con drones



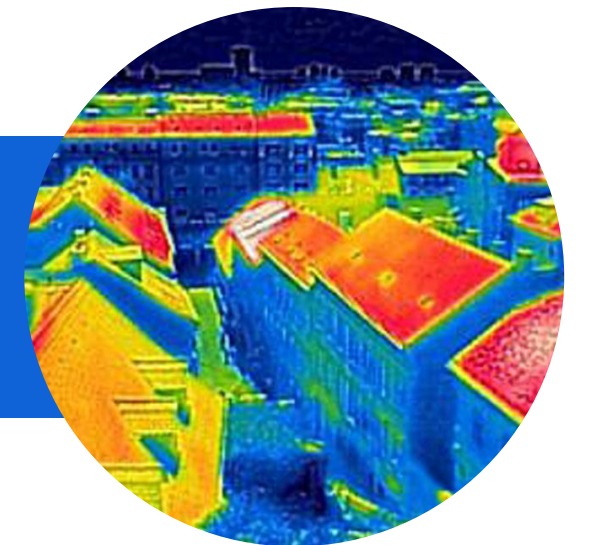
Objetivos de la lección

Comprenda cómo los drones recopilan datos
en la agricultura



Aprende a usar Python para hacer una foto

Descubra cómo las imágenes aéreas ayudan a los



TEMAS

¿Qué es la recopilación de datos en la agricultura inteligente?

Sensores, imágenes aéreas y automatización

Los drones como recopiladores de datos

Debate y preguntas

y respuestas

¿Qué es la recopilación de datos en la agricultura

La agricultura inteligente utiliza tecnologías avanzadas, como drones y sensores, para recopilar datos en tiempo real del campo. Estos datos ayudan a los agricultores a tomar mejores decisiones sobre la salud de los cultivos, el riego, la fertilización y el control de plagas.

- Los drones actúan como «exploradores voladores» que observan rápidamente grandes extensiones de terreno.
- Las cámaras y los sensores de los drones recopilan imágenes y datos ambientales.
- La recopilación de datos ayuda a detectar problemas de forma temprana, antes de que se propaguen.

Objetivo de la misión

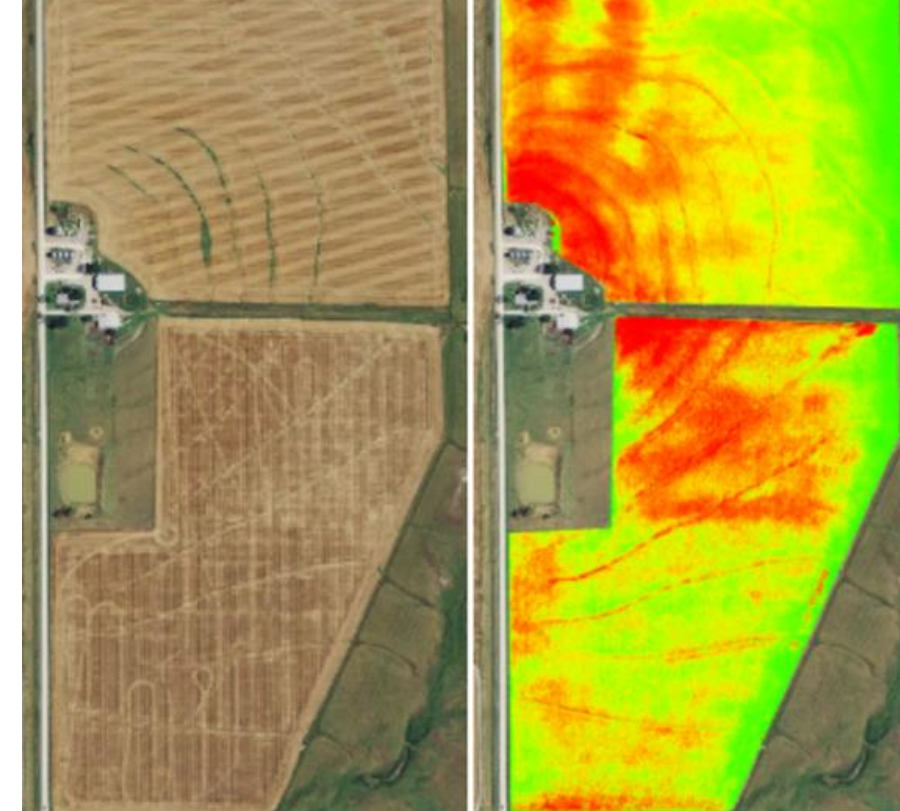
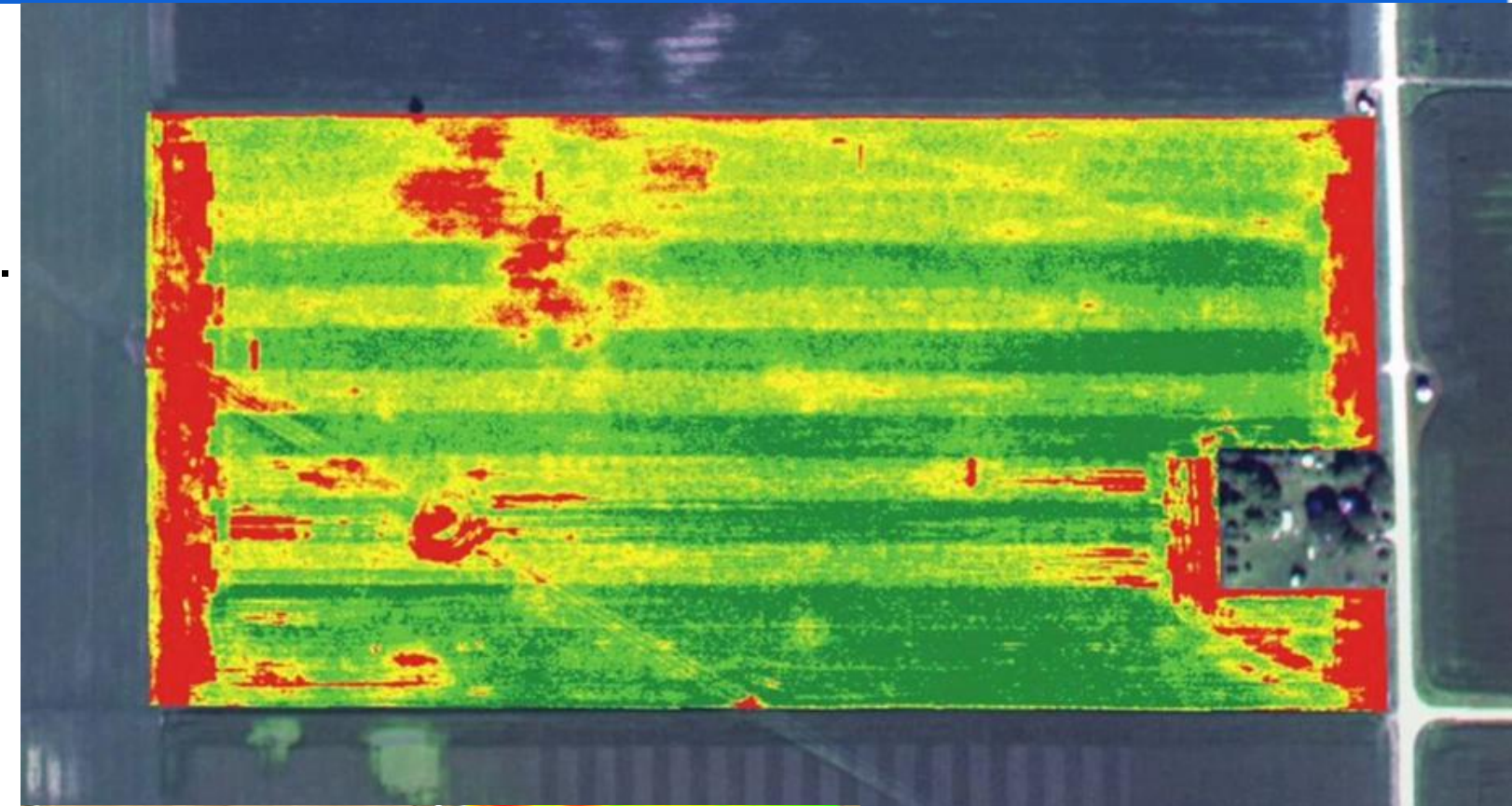
En esta lección, la misión de nuestro dron es encontrar un área problemática específica en el campo y tomar una foto de ella.

- El dron volará siguiendo un patrón predefinido para escanear el campo.
- El código QR se coloca en el área del mapa del suelo donde se encuentran los cultivos enfermos.
- Una vez detectado el QR, el dron se detendrá y tomará una foto para su posterior análisis y volverá a aterrizar.
- Este proceso simula cómo los agricultores pueden identificar y actuar rápidamente ante cualquier problema.



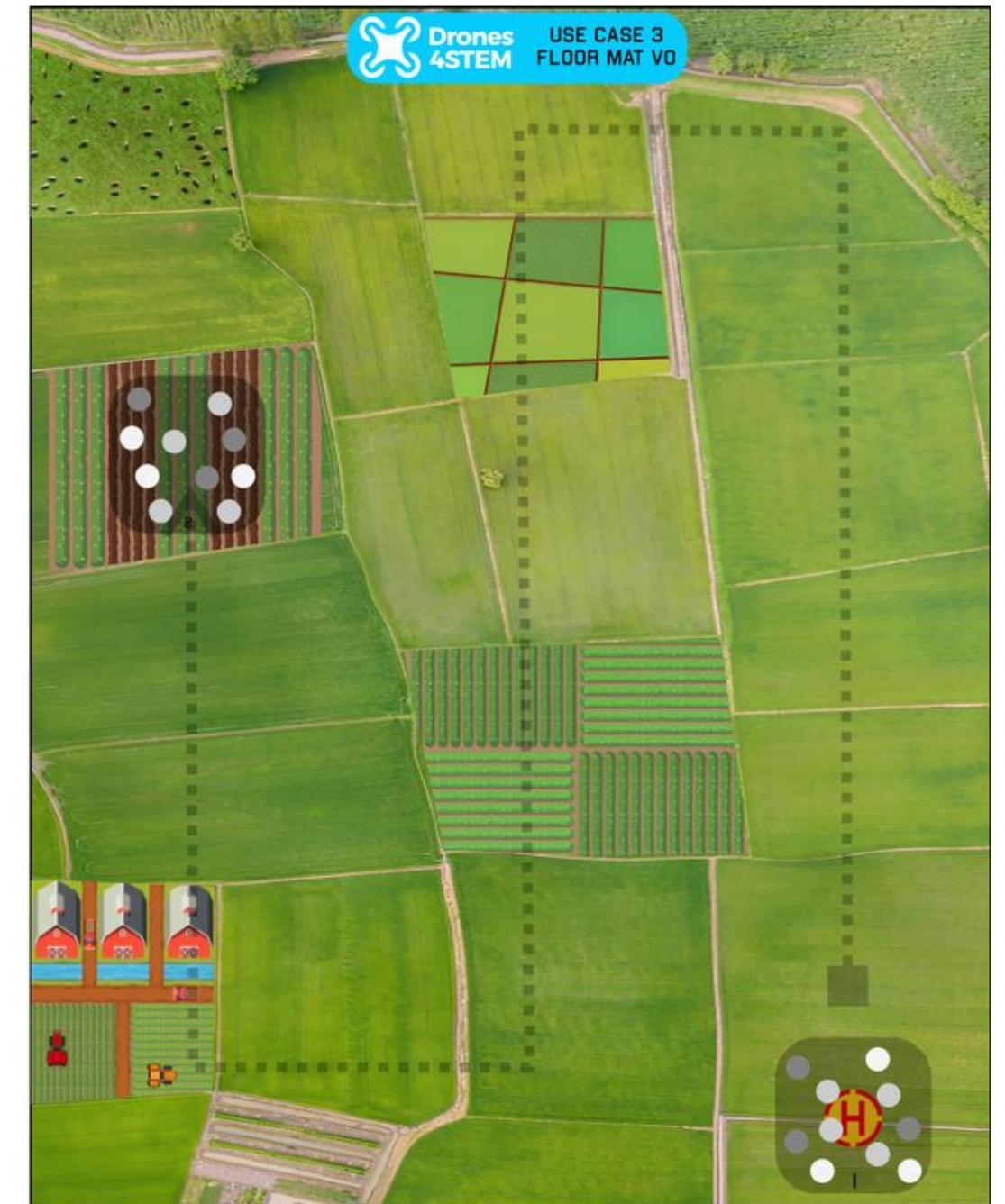
Trayectoria de vuelo del dron

- Para escanear el campo, el dron sigue una trayectoria de vuelo sistemática denominada «cortacésped» o patrón de cuadrícula.
- Esto garantiza una cobertura completa del área sin dejar ningún punto sin explorar. El dron vuela en línea recta, de un lado a otro del campo. La altitud, la velocidad y la distancia entre líneas se establecen previamente en el plan de vuelo.
- Este patrón ayuda al dron a recopilar datos coherentes y precisos en todo el campo.

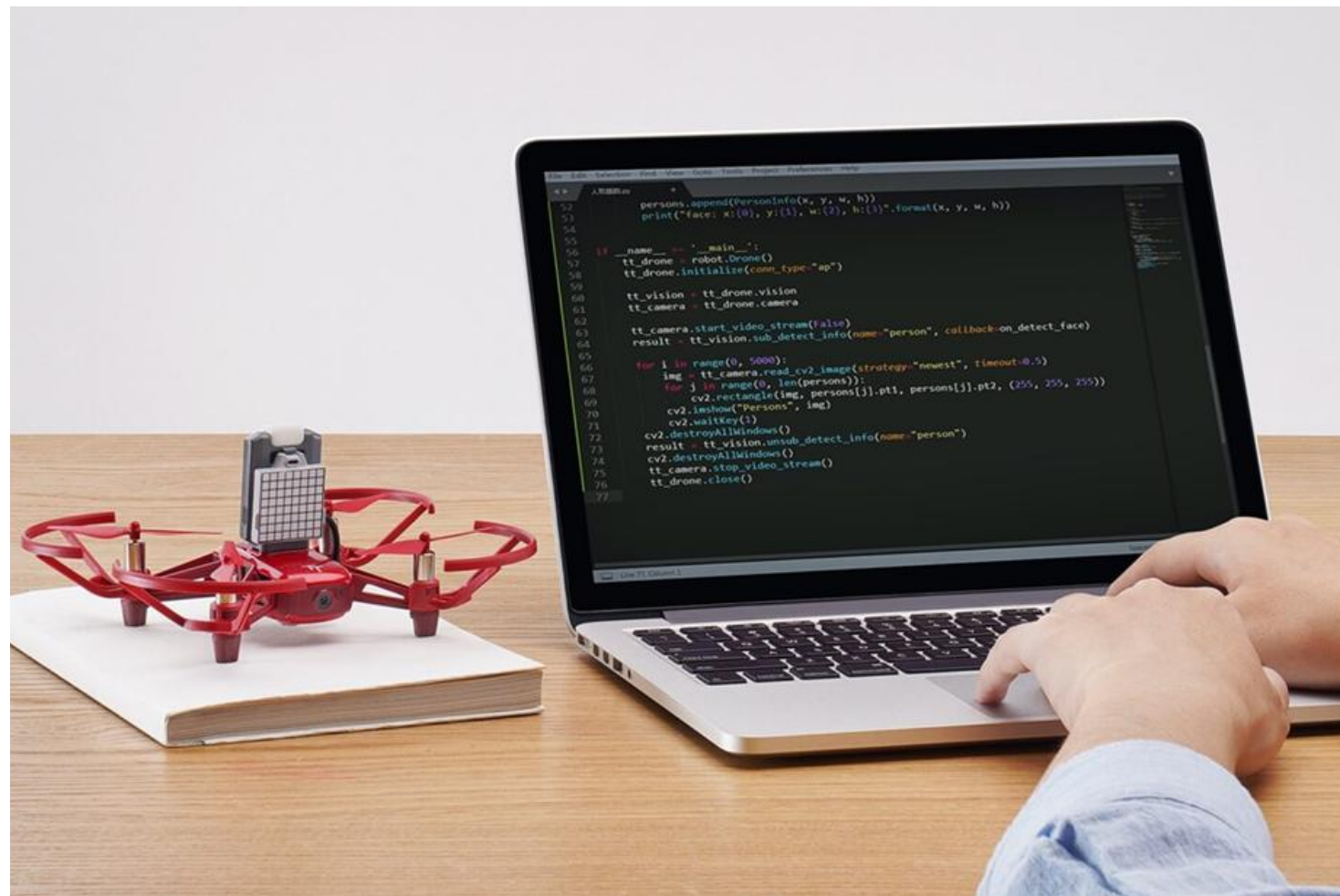


Actividad: Programación de captura de imágenes

- Despegar.
- Realizar un patrón de búsqueda sobre un área predefinida.
- Encuentre cultivos defectuosos
- Tomar una foto y aterrizar.
- Regresar al punto de partida y aterrizar.



Materiales necesarios



HARDWARE

Dron DJI Robomaster TT

Alfombrilla Drones4STEM

PC

SOFTWARE

Plataforma GUI MIND+ Scratch (MÓDULO PYTHON)

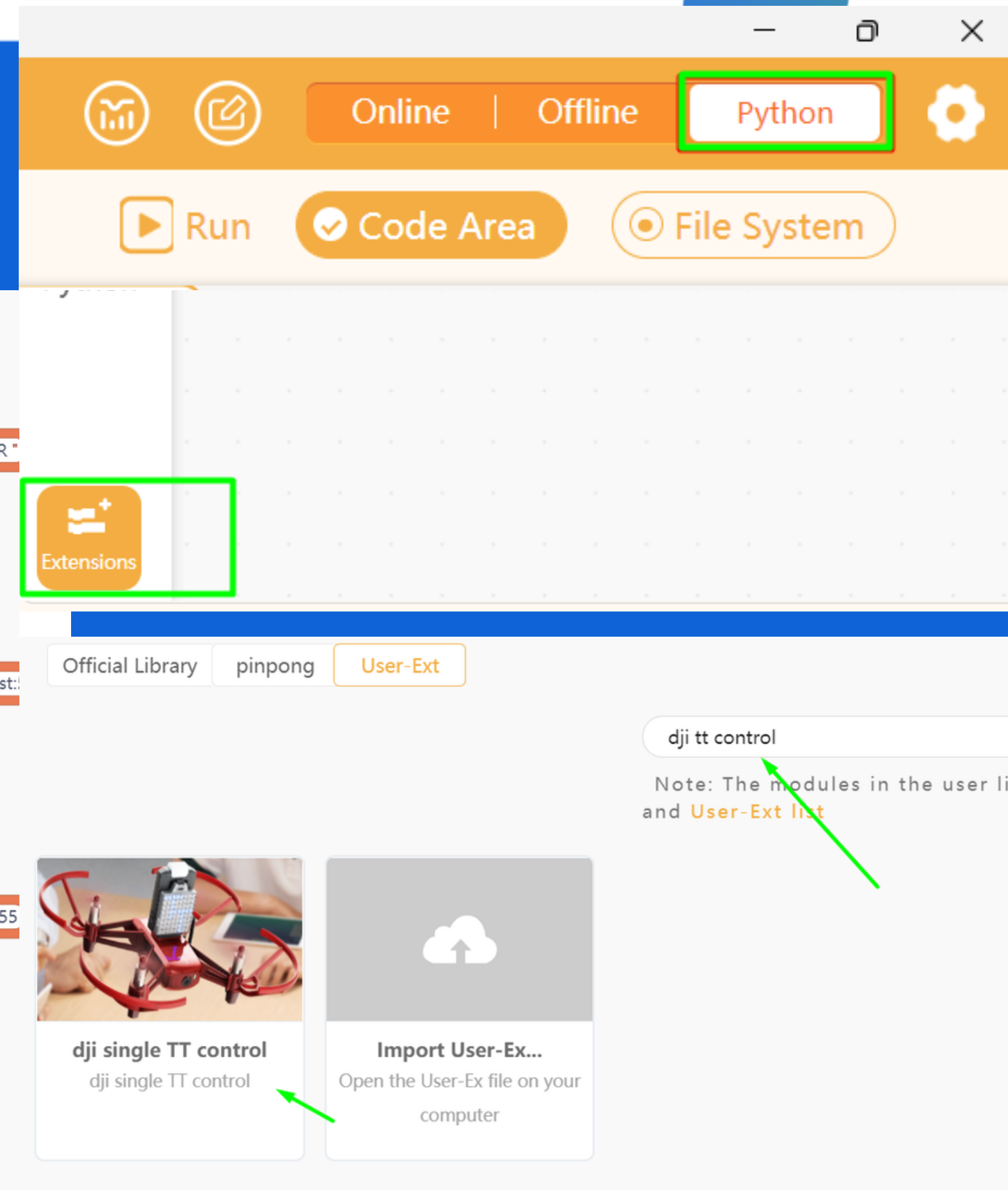
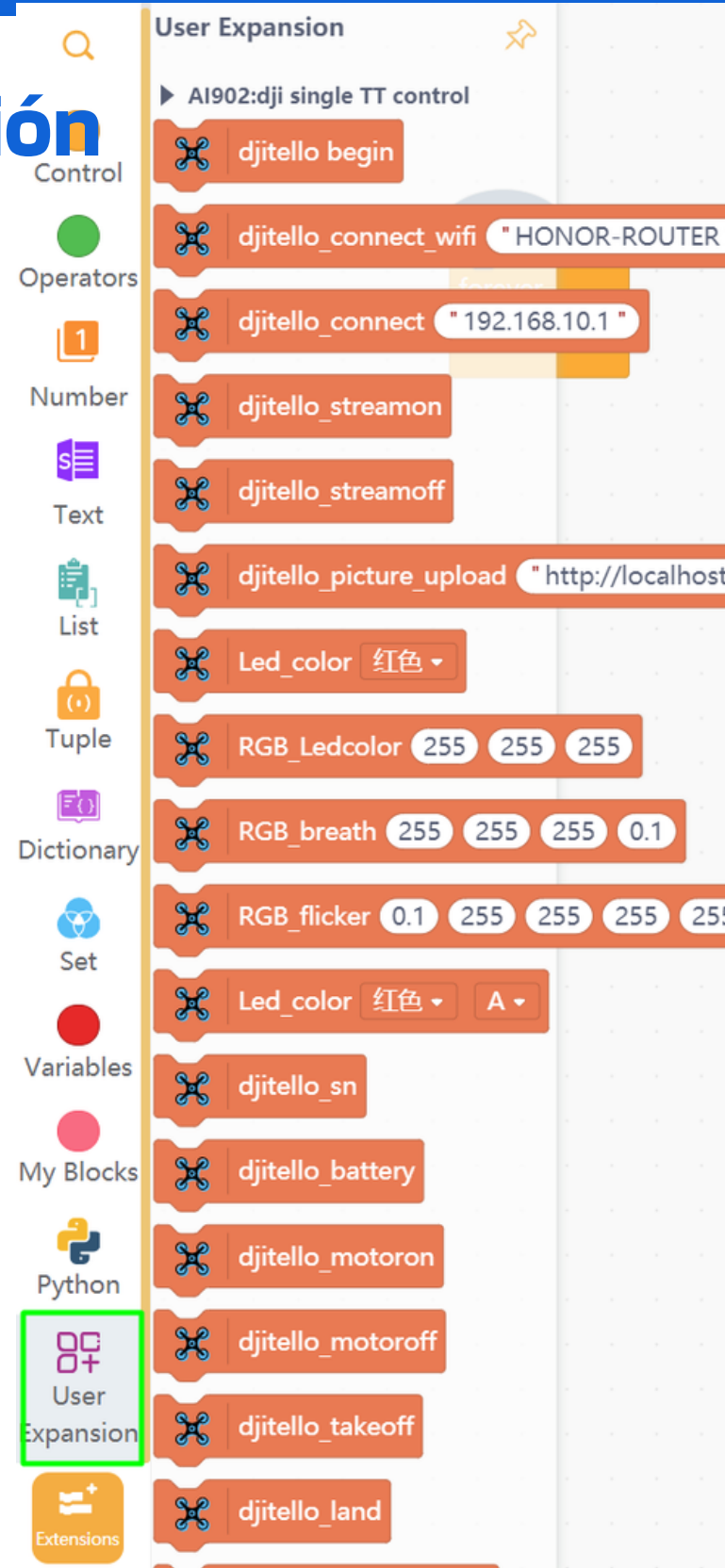
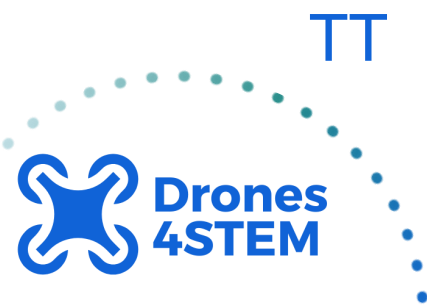
O

Aplicación Drone4Stem

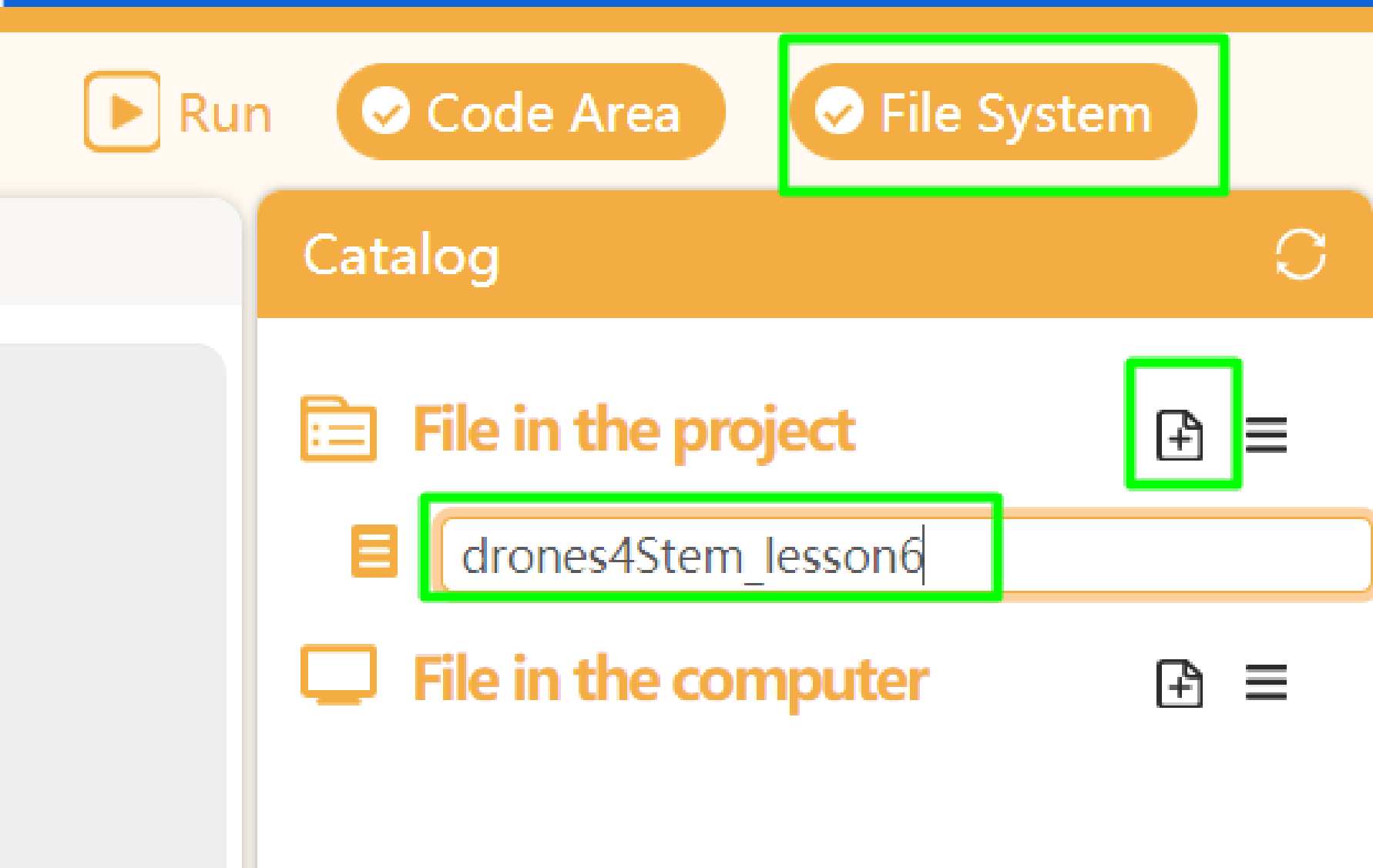
MIND+ Python Instalar

Lista de comprobación de la instalación de Python

- Seleccionar módulo Python
- Seleccione EXTENSIONES-
>USUARIO-EXT y busque: DJI TT
CONTROL y seleccione la biblioteca
- bloque USER EXPANSION
 - Funciones específicas de TELLO



Iniciar un programa Python



- Haga clic en Sistema de archivos
- Añada un archivo (Archivo en el proyecto)
- Escriba un nombre para el archivo

Bloques para la generación de código Python

Mind+ convierte bloques en código Python

Podemos copiar y mejorar eso en el modo CODE.

The screenshot displays the Mind+ IDE interface. On the left, a sidebar contains various tool categories: Control, Operators, Number, Text, List, Tuple, and Dictionary. The main workspace is divided into two panes. The left pane, titled 'Blocks', shows a script starting with 'Python main program start', followed by 'djitello begin', 'djitello_connect' with the IP address '192.168.10.1', 'djitello_streamon', 'djitello_takeoff', a 'forever' loop containing 'djitello_challenge' and 'djitello_moveforward' with the value '20'. The right pane, titled 'Auto Generate', shows the corresponding Python code. A green box highlights the 'Auto Generate' button and the generated code. A green arrow points from the 'djitello_streamon' block in the script to the 'tello.streamon()' line in the code.

Blocks Code Run Code Area File System

Auto Generate

```
1 # -*- coding: UTF-8 -*-
2
3 # MindPlus
4 # Python
5 import requests
6 from PIL import Image
7 import io
8 from djitellopy import Tello
9 import cv2
10 from djitellopy import TelloSwarm
11
12
13 tello = Tello(host="192.168.10.1")
14 tello.connect()
15 tello.streamon()
16 tello.takeoff()
17
18 while True:
19     tello.enable_mission_pads()
20     tello.set_mission_pad_detection_direction(0)
21     tello.move_forward(20)
22
```

Catalog

- File in the project
 - drones4stem_lesson6.py
- File in the computer

Python main program start

- djitello begin
- djitello_connect "192.168.10.1"
- djitello_streamon
- djitello_takeoff
- forever
 - djitello_challenge
 - djitello_moveforward 20

Drones 4STEM

Bloques para la generación de código Python

The screenshot displays the Mind+ V1.8.0 RC3.1 interface. The top bar includes the Mind+ logo, 'Project' and 'Learning' dropdowns, and buttons for 'Online', 'Offline', and 'Python'. Below this, a 'Blocks' tab is visible next to a 'Code' tab, which is highlighted with a green box. The 'Code' tab shows a Python script for controlling a Tello drone. The script includes imports for 'requests', 'PIL Image', 'io', 'Tello', 'cv2', and 'TelloSwarm'. It sets up a connection to a Tello drone at IP '192.168.10.1', streams video, and takes off. A 'while True' loop enables mission pads, sets the detection direction to 0, and moves forward by 20 units. On the right, a 'Catalog' panel shows 'File in the project' and 'File in the computer' sections. The 'File in the project' section has a green box around 'drones4Stem_lesson6.py'. The bottom panel is a 'Terminal' window showing the Python 3.8.5 prompt and the command prompt 'Type "help", "copyright", "credits" or "license" for more information.'.

```
1  # -*- coding: UTF-8 -*-
2
3  # MindPlus
4  # Python
5  import requests
6  from PIL import Image
7  import io
8  from djitellopy import Tello
9  import cv2
10 from djitellopy import TelloSwarm
11
12
13 tello = Tello(host="192.168.10.1")
14 tello.connect()
15 tello.streamon()
16 tello.takeoff()
17
18 while True:
19     tello.enable_mission_pads()
20     tello.set_mission_pad_detection_direction(0)
21     tello.move_forward(20)
22
```

Terminal

Python 3.8.5 [MSC v.1924 64 bit AMD64] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _

Especificaciones de la alfombra Drones4Stem



- Cada cuadrícula mide 40 x 40 cm
- Movimiento hacia delante: 100 cm
- Cambio de carril: deslizamiento de 40 cm
- El código QR 2 indica un riego deficiente
- El código QR 4 indica cultivos defectuosos

Inicialización y despegue

Inicializar

```
tello = Tello()
tello.connect()
print(f"Batería: {tello.get_battery()}%")
```

Iniciar transmisión de vídeo

```
tello.streamon()
lectura_fotogramas = tello.get_frame_read()
```

Habilitar Mission Pads

```
tello.enable_mission_pads()
tello.set_mission_pad_detection_direction(0) # Detección hacia abajo
```

Despegar

```
tello.takeoff()
time.sleep(2)
```

Comprueba el funcionamiento de las almohadillas de misión, foto, regreso a

```
# Define helper to check for Pad 4
def check_mission_pad():
    pad = tello.get_mission_pad_id()
    print(f"Detected Pad: {pad}")

    if pad == 4:
        print("Pad 4 detected - taking photo and returning to Pad 1")

        # Take photo
        frame = frame_read.frame
        cv2.imwrite("pad4_photo.jpg", frame)
        print("Photo saved as pad4_photo.jpg")

        # Move to return path
        tello.move_right(80)
        tello.move_back(80)
```

```
# Try to find Pad 1
print("Looking for Pad 1 to land...")
for _ in range(10): # Retry up to 10 times
    pad1 = tello.get_mission_pad_id()
    print(f"Pad ID: {pad1}")
    if pad1 == 1:
        print("Pad 1 found - landing.")
        tello.land()
        tello.streamoff()
        exit()
    time.sleep(1)

print("Pad 1 not found. Landing anyway.")
tello.land()
tello.streamoff()
exit()
```

Patrón de cortacésped

```
# Flight pattern with Pad 4 check
tello.move_forward(120)
check_mission_pad()

tello.move_left(40)
check_mission_pad()

tello.move_back(120)
check_mission_pad()

tello.move_left(40)
check_mission_pad()

tello.move_forward(120)
check_mission_pad()

# End of pattern
print("Pattern complete. Mission Pad 4 not found. Landing.")
tello.land()
tello.streamoff()
```

Normas de seguridad para volar drones en interiores

1 Antes del vuelo:

- Inspeccione el dron
 - Compruebe si hay daños o hélices sueltas.
 - Asegúrese de que la batería esté completamente cargada.
- Prepare el área
 - Retire obstáculos como sillas, escritorios y otros objetos.
 - Asegúrese de que el entorno sea amplio y abierto.
 - Cierre las ventanas o las fuentes de corrientes de aire.
- Informe a los participantes
 - Informe a todos sobre los protocolos de seguridad.
 - Mantenga una distancia de seguridad de al menos 2 metros con respecto al dron.

Normas de seguridad para volar drones en interiores

2 Durante el vuelo:

- Despegue y aterrizaje controlados
 - Despegue y aterrice en una superficie plana y estable.
 - Evite volar cerca de personas u objetos.
- Límites de vuelo
 - Mantenga el dron dentro del campo de visión.
 - Vuela a baja altura para evitar chocar con el techo.
 - Aterrice cuando la batería esté por debajo del 20 %.
- Protocolo de emergencia
 - Detenga el dron inmediatamente si se sale de control. (SPACE inicia el ATERRIZAJE)
 - Utilice el botón de «parada de emergencia» de la aplicación si es necesario. («e» detiene los motores).

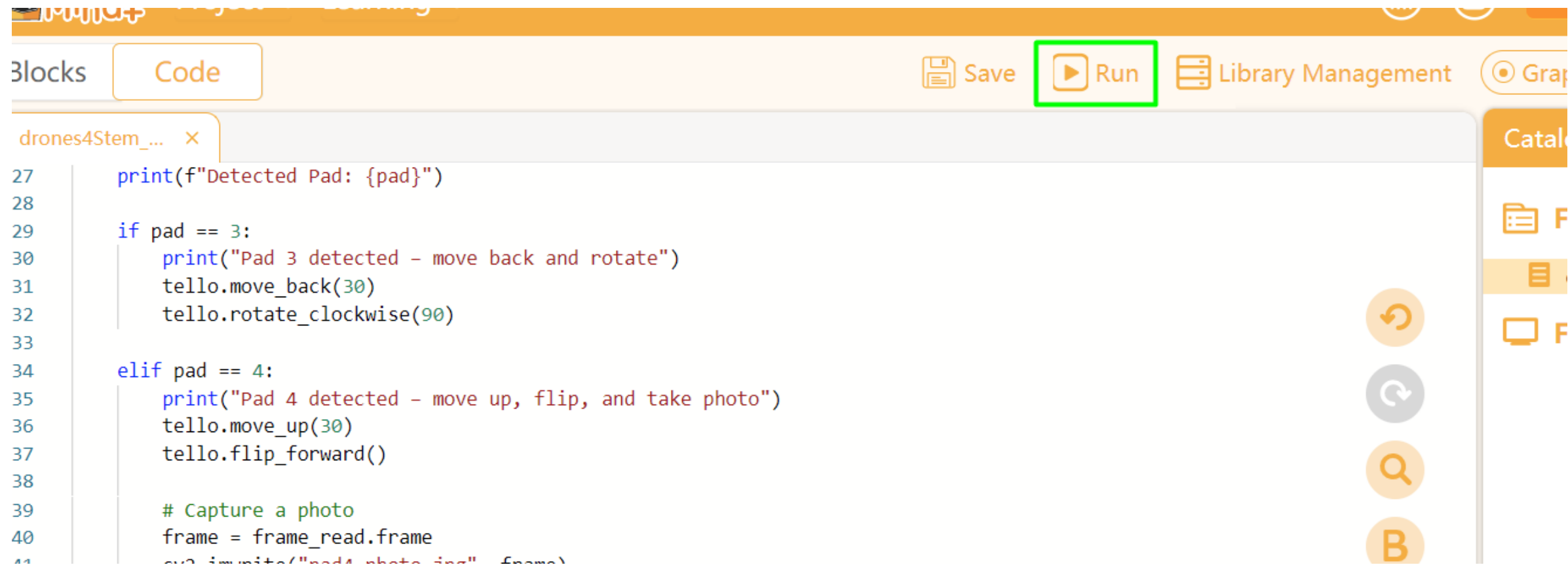
Normas de seguridad para volar drones en interiores

3 Después del vuelo:

- Apagar de forma segura
 - Apague el dron y desconecte la batería.
- Compruebe si hay daños
 - Compruebe si hay algún problema después del vuelo.
- Guárdelo de forma segura
 - Guarde el dron en un lugar seguro y seco.

MANTENGA SIEMPRE LA CALMA Y LA CONCENTRACIÓN

Prueba el programa



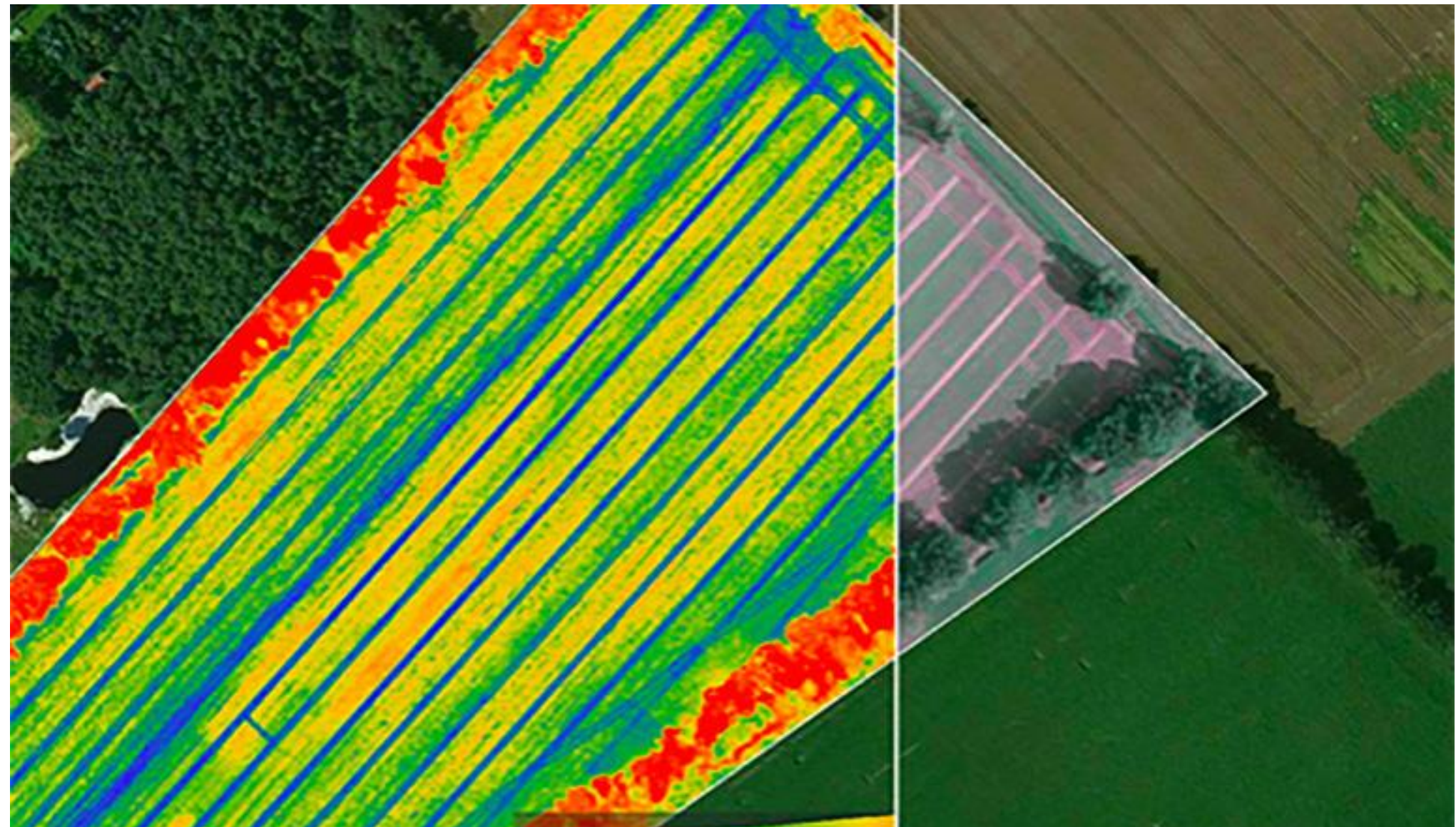
The screenshot shows the Tello EDU programming environment. The 'Code' tab is selected, displaying a Python script for controlling a Tello drone. The script includes comments and function calls for movement and photography. The 'Run' button, represented by a play icon, is highlighted with a green rectangular box. Other interface elements include a 'Save' button, 'Library Management', and a 'Graph' button. A sidebar on the right contains a 'Catalog' and several icons for file management and execution.

```
27 print(f"Detected Pad: {pad}")
28
29 if pad == 3:
30     print("Pad 3 detected - move back and rotate")
31     tello.move_back(30)
32     tello.rotate_clockwise(90)
33
34 elif pad == 4:
35     print("Pad 4 detected - move up, flip, and take photo")
36     tello.move_up(30)
37     tello.flip_forward()
38
39     # Capture a photo
40     frame = frame_read.frame
41     save_image(f"pad4_photo.jpg", frame)
```



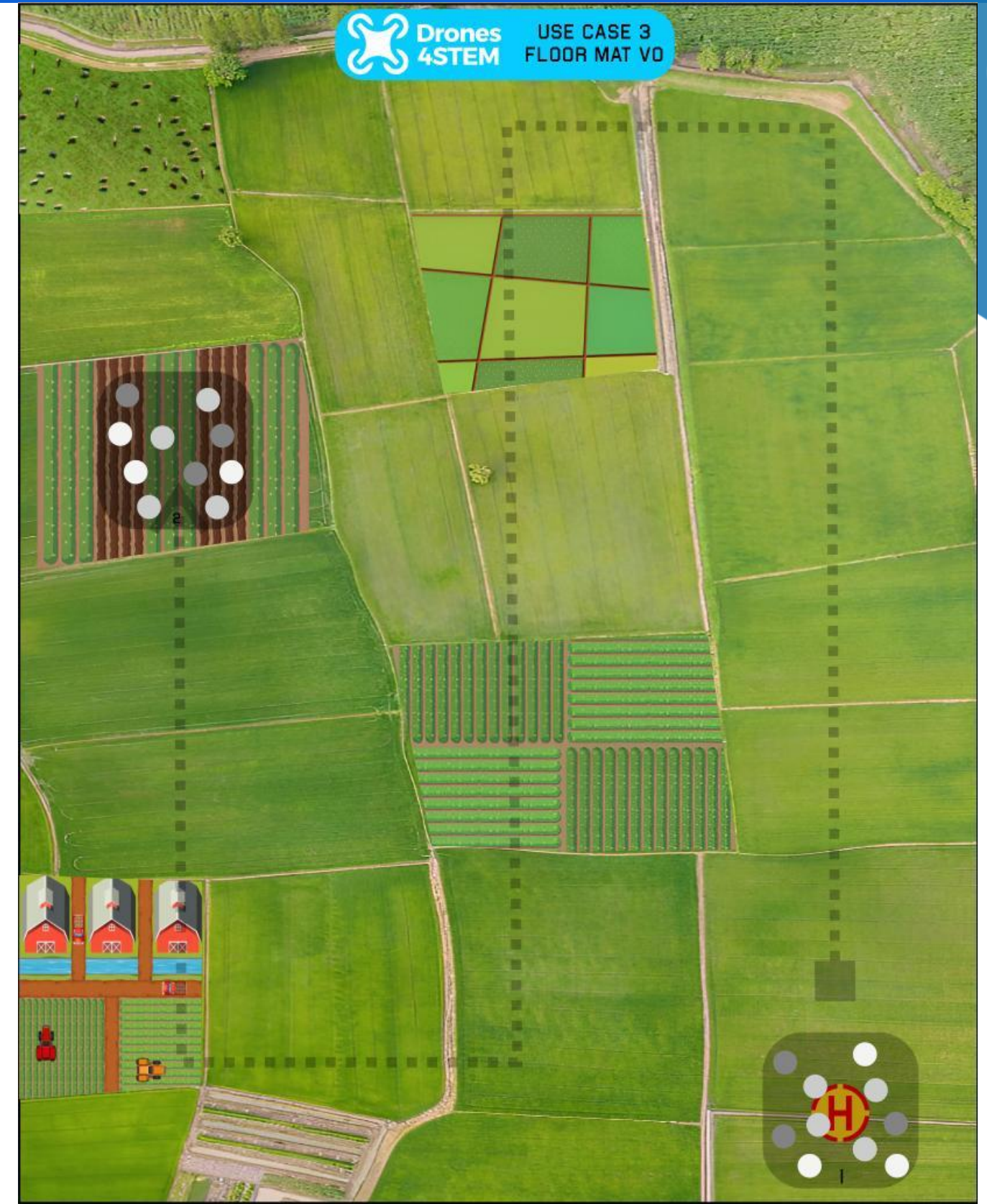
Análisis básico de imágenes

- Cultivos sanos: colores verdes brillantes.
- Áreas problemáticas: manchas amarillas o marrones, hojas marchitas.
- Problemas del suelo: suelo seco o agrietado.



Mejoras y limitaciones

- ¿Qué pasará si no se reconoce la Mission Pad?
- ¿Se puede añadir algún patrón LED (por ejemplo, rojo para cultivos defectuosos)?
- ¿Qué ocurre si la batería se está agotando?
- ¿Por qué el dron aumenta la altitud antes de tomar la foto?



RESUMEN

- Lo que hemos aprendido:
 - Por qué los drones son importantes en la agricultura inteligente.
 - Cómo instalar el módulo Python.
 - Cuáles son los comandos básicos para inicializar DJI Tello con Python.
 - Cómo crear una función para leer las plataformas de mi
 - Cómo crear una ruta para cortacéspedes.
- La seguridad es lo primero:
 - Importancia de despejar la zona de vuelo.
 - Mantener un entorno seguro para todos.



Recursos adicionales

Lecturas recomendadas

- <https://djitellopy.readthedocs.io/en/latest/tello/>

Vídeos

[Cómo instalar el módulo Python](#)



¡Gracias por su atención!



Co-funded by
the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.