# CIRCUITO DEL DRON CON FOTO DEL AULA ATECA

Natalia Casillas, Víctor Carrillo, José María Girón





#### <u>1º Entorno de desarrollo</u>





### <u>2° Explicar el código</u>

#### <u>3º Vídeo del recorrido</u>

## Entorno de Desarrollo Pasos para descargar el entorno



## Primer paso:

Tienes que buscar la aplicación VirtualBox y cuando la tengas la descargas.
 Primero una breve definición de VirtualBox

VirtualBox es un software para virtualización, también conocido como hipervisor de tipo 2, que se utilizar para virtualizar sistemas operativos dentro de nuestro ordenador existente

## Segundo Paso:

A continuación descárgate la ISO del CD de instalación de **Ubuntu Server** 22.04 LTS

**Definición:** Este sistema operativo dispone de un gran rendimiento para servidores y funcionalidades de virtualización con Docker entre otras.











### Tercer Paso:

El siguiente paso es crear una máquina virtual para Ubuntu Server 22.04 e indicarle dónde está la imagen ISO con la instalación.





--//////\\\\\\\\ ---//////\\\ \_\_\_////// **```** ----////// 111----////// 1111111 · / | | KStop stop-color="#86181F offset / ///||\\\\ (stop stop-color="#103848" offset+"#\*\*/> «/linearGradient> c/defs> //// | \ \ crect width="800" height="450" rx="8" fill="willes NO. COMPANY ////// Ksvg width-"96" height="96" viewBox-"0 0 % >= animal -----<linearGradient x1="87.5651" y1="15.000" #</pre> «defs> <stop stop-color-"#FFF" offset-" --//// <feGaussianBlur stdDeviation Im torNatrix values \*\*\* \* \* \* 

 $\prime$   $\prime$   $\prime$   $\prime$ 

Una vez iniciada la máquina y terminada la instalación hay que proceder a instalar las librerías

## Cuarto paso:

# Quinto paso:

Crea el programa Python que aparece como ejemplo y, aunque no tengas acceso al drone, comprueba que se puede ejecutar





# Sexto y último paso:

Busca información sobre qué paquete o paquetes es necesario instalar y procede a su instalación.

	31 32 33 34 35 36	<pre>self.file = self. self.fingerprints = self. self.logdupes = logsing self.logger = logsing</pre>
	37 38 39 40	<pre>if path: self.file = cont(05.0000.000000000000000000000000000000</pre>
	41 42 43 44 45	<pre>@classmethod def from_settings(cls, settings) debug = settings.getbeel("Bureral We amount return cls(job_dir(settings))</pre>
	40 47 48 49 50	<pre>def request_seen(setr. request_ fp = self.request_fingerprints if fp in self.fingerprints return True self.fingerprints.add(fp)</pre>
	51 52 53 54 55 55 56	<pre>if self.file: self.file.write(fp + of the self self.file.write(fp + of the self)) def request_fingerprint(self.request) return request_fingerprint(request)</pre>



Eavoritos					
	a18gipajo [Corriendo] - Oracle	e VM VirtualBox		• • •	
GNU nano 6.2 import cv2_	dron_circ	uito.py		i =	1
from djitellopy import Tello				whil	e i < 4: tello move forwary
tello = Tello() tello.connect() tello.takeoff()					tello.move_rorward tello.move_up(30) print(tello.get_h i += 1
i = 1 while i < 4:				if t	h = tello.get_heig ello.get_height(); ppipt('Estamos.us
tello.move_torWard(50) tello.move_up(30)				1	tello.move_down(20
<pre>print(tello.get_height()) i += 1 h = tello.get height()</pre>				else	: print('El dron es
<pre>if tello.get_height()&gt; 190: print('Estamos volando demasi tello.move_down(20)</pre>	ado alto, vamos a ba	jar 20 centimet	ros')	tell	o.rotate_counter_(
else: print('El dron esta a una alt	ura segura, sigamos v	volando')		tell fram cv2.	o.streamon() e_read = tello.ge imwrite("picture.
tello.rotate_counter_clockwise(18	0)			tell	o.streamoff()
tello.streamon() frame_read = tello.get_frame_read	0			j =	1
cv2.imwrite("picture.png", frame_ tello.streamoff() j = 1	read.frame)			whil	e j < 3: tello.move_forwar( tello.move_down(3) i += 1
while j < 3: tello.move_forward(50) tello.move_down(30)				tell tell	o.rotate_clockwise
tello.rotate_clockwise(270)	[ Concelle	4 1			
<mark>^G</mark> Help <mark>^O</mark> Write Out   ₩ Wh <mark>^X</mark> Exit <mark>^R</mark> Read File <mark>`\</mark> Re	ere Is	, J ∩T Execute ^J Justify	<mark>^C</mark> Location <mark>M−U</mark> Und <mark>^−</mark> Go To Line <mark>M−E</mark> Red	io tell	o.land()

## Explicación del código

^G Help

```
d(50)
          eight())
          ght()
           190:
           a<u>n</u>do demasiado alto, vamos a bajar 20 centimetros')
          ))
           ) a una altura segura, sigamos volando')
          clockwise(180)
           _frame_read()
           ng", frame_read.frame)
          1(50)
          e(270)
          ),0,100,90)
<sup>^</sup>O Write Out <sup>^</sup>₩ Where Is <sup>^</sup>K Cut
                                                    <sup>ˆ</sup>⊤ Execute
                                                                     ℃ Location M-
```

```
[INFO] tello.py - 461 - Response forward 50: 'ok'
[INFO] tello.py - 437 - Send command: 'up 30'
[INFO] tello.py - 461 - Response up 30: 'ok'
90
[INFO] tello.py - 437 - Send command: 'forward 50'
[INFO] tello.py - 461 - Response forward 50: 'ok'
[INFO] tello.py - 437 - Send command: 'up 30'
[INFO] tello.py - 461 - Response up 30: 'ok'
120
[INFO] tello.py – 437 – Send command: 'forward 50'
[INFO] tello.py - 461 - Response forward 50: 'ok'
[INFO] tello.py - 437 - Send command: 'up 30'
[INFO] tello.py - 461 - Response up 30: 'ok'
150
El dron esta a una altura segura, sigamos volando
[INFO] tello.py - 437 - Send command: 'ccw 180'
[INFO] tello.py - 461 - Response ccw 180: 'ok'
[INFO] tello.py – 437 – Send command: 'streamon'
[INFO] tello.py - 461 - Response streamon: 'ok'
[INFO] tello.py - 437 - Send command: 'streamoff'
[INFO] tello.py - 461 - Response streamoff: 'ok'
[INFO] tello.py - 437 - Send command: 'forward 50'
[INFO] tello.py - 461 - Response forward 50: 'ok'
[INFO] tello.py - 437 - Send command: 'down 30'
[INFO] tello.py - 461 - Response down 30: 'ok'
[INFO] tello.py - 437 - Send command: 'forward 50'
[INFO] tello.py - 461 - Response forward 50: 'ok
[INFO] tello.py - 437 - Send command: 'down 30'
[INFO] tello.py - 461 - Response down 30: 'ok'
[INFO] tello.py - 437 - Send command: 'cw 270'
[INFO] tello.py – 461 – Response cw 270: 'ok'
[INFO] tello.py - 437 - Send command: 'go 400 0 100 90'
[INFO] tello.py - 461 - Response go 400 0 100 90: 'ok'
[INFO] tello.py - 437 - Send command: 'land'
[INFO] tello.py - 461 - Response land: 'ok'
Segmentation fault (core dumped)
a18gipajo@a18gipajo:~$ python3 dron_circuito.py
```

B 🖉 🗢 🖬 🤉 94% C O O

The survey of the

Concernent in



---/

////

-----

----

1111







Muchay

Gracias



