



# **Introducción al desarrollo de software para vehículos aéreos no tripulados (UAVs) inteligentes**

## Sección I

Leonardo Camargo Forero, Ph.D

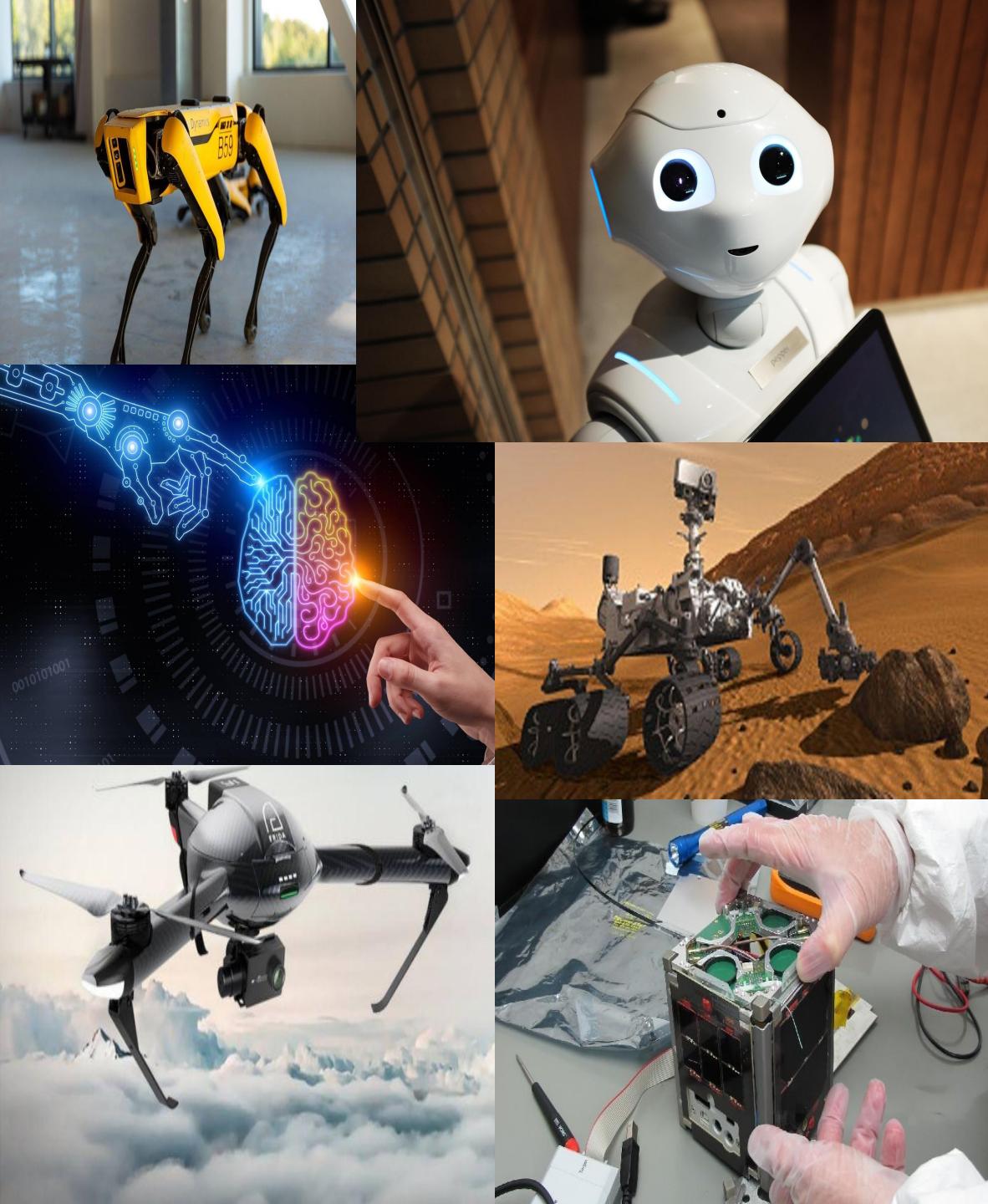
CEO **UbiHPC**  
[www.ubihpc.com](http://www.ubihpc.com)



# Agenda

- Introducción
- Aplicaciones con UAVs
- Emprendimiento
- Tutorial de Git y GitHub
- Tutorial de Python
- Preguntas

# Introducción



# Acerca de mi

## Educación

- Ingeniero de Sistemas – Universidad Industrial de Santander
- Magíster en redes y computación ubicua – Université Nice Sophia Antípolis
- Doctor en ciencia y tecnología aeroespacial – Universitat Politècnica de Catalunya

## Experiencia laboral

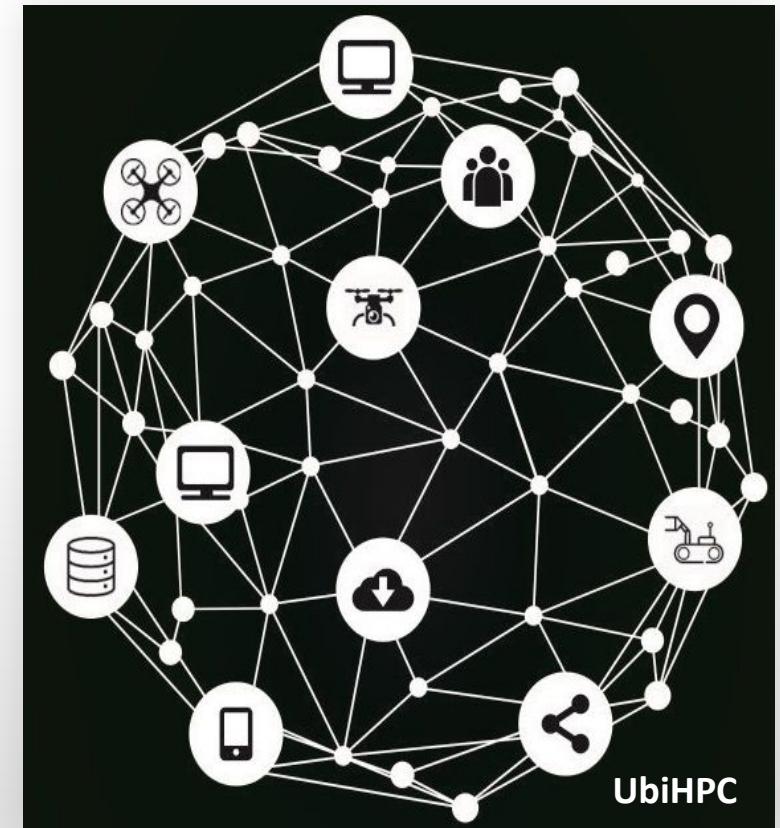


Universidad  
Industrial de  
Santander

## Otras cosas

Escritor de un thriller de ciencia ficción – The Dark Buddha.

[Amazon](#)



# UbiHPC



Desarrollamos software que combina  
supercomputación e Inteligencia Artificial



INCEPTION PROGRAM



Mayo 2020

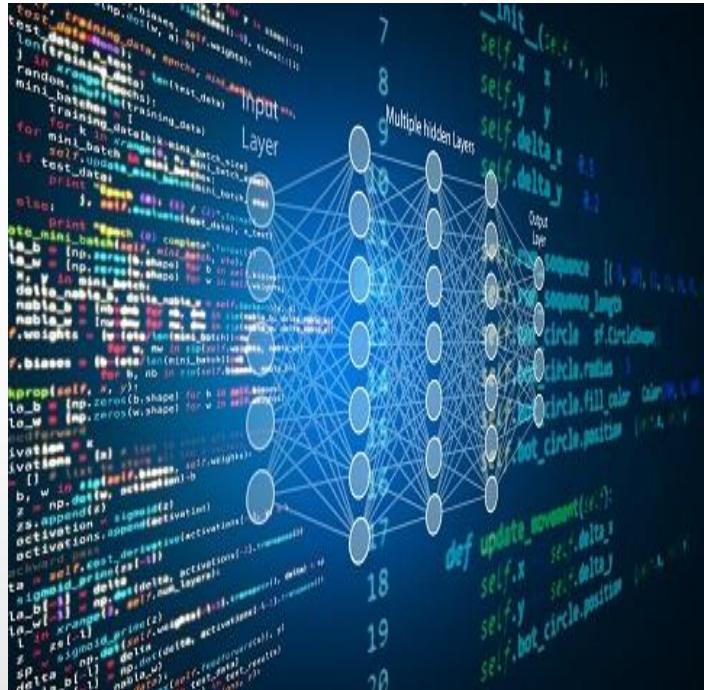


ROYAL  
ACADEMY OF  
ENGINEERING

Leaders in Innovation Fellowship  
60 companies in Latin America

Noviembre 2019

# UbiHPC



Analítica de datos e IA

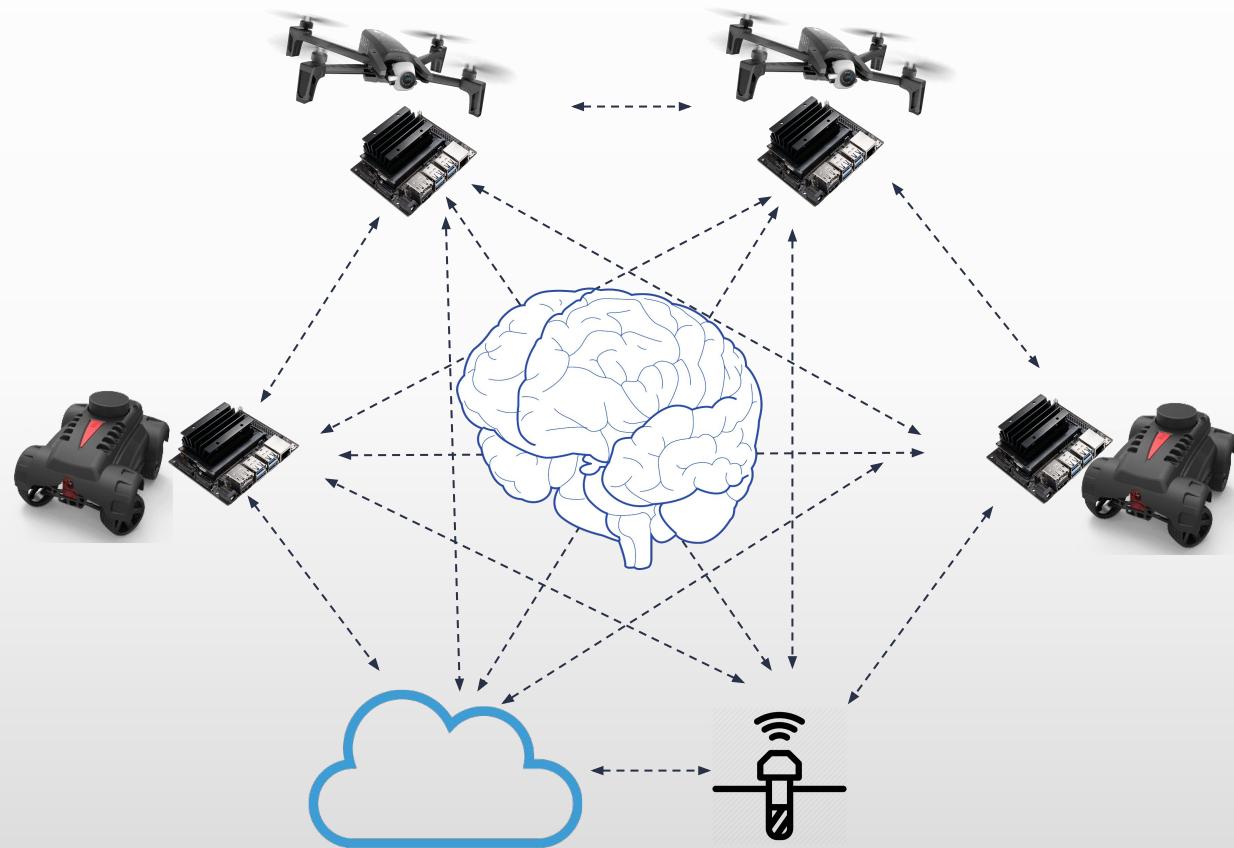


Internet de las Cosas Inteligentes  
(AIoT)



Disquera de música  
genética

# The ARCHADE



# Supercomputing

La siguiente evolución de AoT es crear supercomputadores *in the Edge* y conectarlos con la nube



# The ARCHADE

A distributed AoT collective intelligence to join them all



# The ARCHADE

Algunas aplicaciones (software) que se pueden crear con The ARCHADE



Precision agriculture

Smart homes

Search & Rescue

Wearables

Monitoring / surveillance

Oil & Gas

Space

Drone sports

8



SDK



APIs



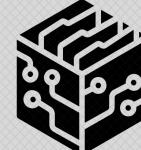
Framework



Templates



Middleware



SimPlat



SaaS



Desktop app



Apps market

The ARCHADE software suite

# Detector y predictor de tormentas eléctricas (The ARCHADE)



50K USD/hora



The ARCHADE SDK - AI (CNN)

Detección de rayos      Predicción de clima



The ARCHADE SDK - AI (RNN)

Predicción de dónde ocurrirán tormentas eléctricas

# NEO: The ethical hacker robot



- Hay un ataque de hacking cada 39 segundos
- 43 % de los ataques cibernéticos ocurren en las PYMES.
- El costo promedio global de un ataque cibernético en PYMES es de \$3.9M USD (SMBs) y \$116M USD en grandes empresas.
- El FBI reporta, que desde que comenzó el COVID-19, la cantidad de ataques cibernéticos ha incrementado 300%

[Fuente](#)

## Problema

- No sabemos cómo defendernos
- Necesitamos protección 24/7/365
- Antivirus no es suficiente y
- Ayuda profesional es muy costosa (15.000 USD en promedio)



## Solución

### NEO (The ARCHADE)

1. Protección automática 24/7/365
2. Aprendizaje continuo usando IA local
3. Aprendizaje continuo mediante cooperación con otros NEOs
4. Mucho más económico que ethical hackers

# GenM



**GenM**  
Music about you



## GenM-ers (músicos genéticos)



Banda



Finalista de la voz teen  
(voz de la ballena)



DJ famoso colombiano  
(Marihuana THC)



5 músicos  
(Diferentes canciones)

## Jingles



Jingle para empresa  
(Marihuana CBD)



Jingle para empresa  
(Aguacate)

## Canciones especiales

Mascotas, personas, etc

# Club 4.0

club40ubihpc@gmail.com



Club 4.0  
Por una Colombia innovadora y competitiva

 python

Introducción al desarrollo de software para vehículos aéreos no tripulados (UAVs) inteligentes



Club 4.0  
Por una Colombia innovadora y competitiva

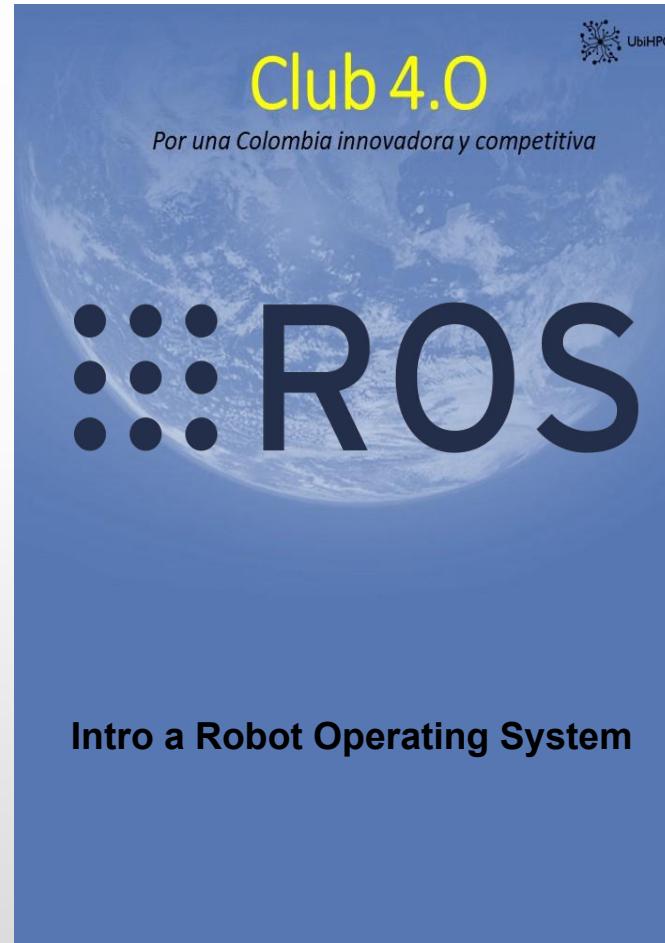
Deep Learning y Visión por Computador



Club 4.0  
Por una Colombia innovadora y competitiva

Inteligencia Artificial y Ciencia de datos

# Club 4.0



# De qué se trata este curso?



Transformar UAVs en mucho más  
que cámaras volando

# Agenda del curso

## Sección 1 (4 horas)

- Revisión de aplicaciones actuales con UAVs y aplicaciones futuras inmediatas
- Emprendimiento
- Tutorial de Python general
- Tutorial de Git y GitHub

## Sección 2 (4 horas)

- Tutorial de ArduPilot SITL (Software In The Loop)
- Instalación y configuración de librerías Python para control de piloto automático.
- Tutorial de DroneKit
- Tutorial de Amazon Web Services para simulación

15

## Sección 3 (4 horas)

- Tutorial de cluster de UAVs simulado
- Programación de aplicación software para un solo UAV
- Programación de aplicación software para múltiples UAV

## Sesión 4 (3 horas)

- Introducción a Deep Learning y Visión por computador con OpenCV.

# Acerca de la sección número 1

## Teoría

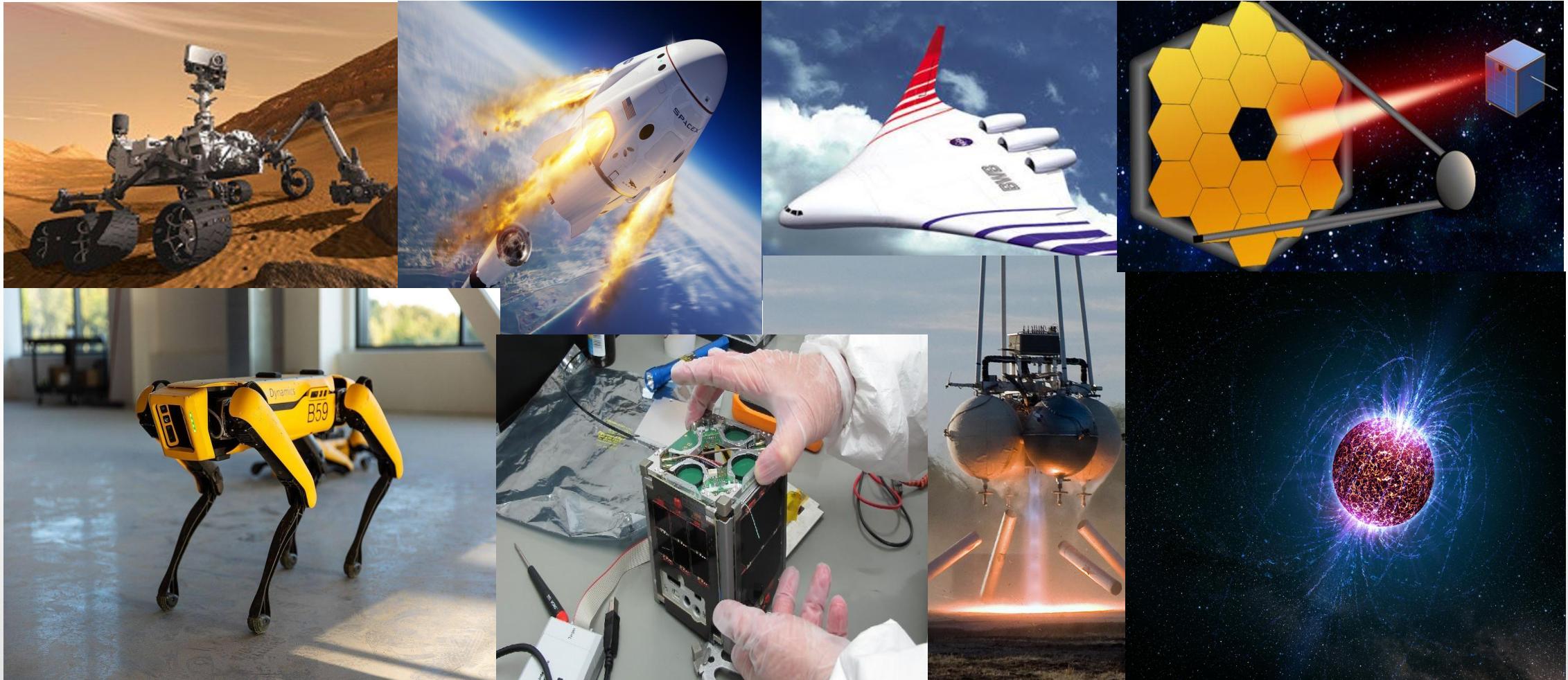
- Revisión de aplicaciones actuales con UAVs y aplicaciones futuras inmediatas
- Emprendimiento

## Práctica

- Tutorial de Git y GitHub
- Tutorial de Python general

## Tareas

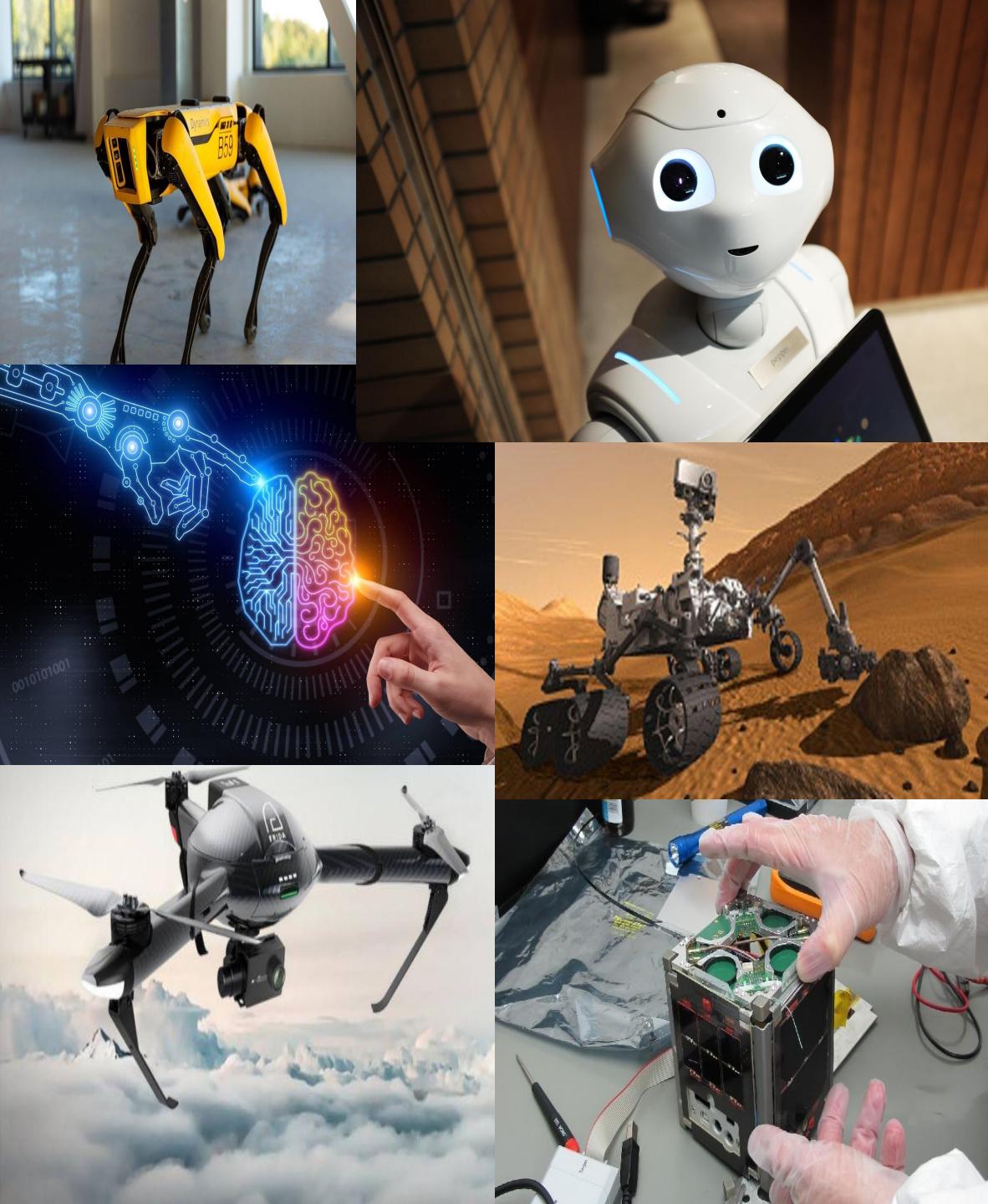
- Instalación y configuración de ArduPilot SITL (Software In The Loop)
- Creación de cuenta en Amazon Web services



# Felicitaciones !!!

# ¿Como están con la computación ?

18



# Aplicaciones con UAVs

19

# Aplicaciones con UAVs



20

**Uso tradicional**

# Aplicaciones con UAVs



Agricultura de precisión



Búsqueda y Rescate



Monitoreo/vigilancia



Construcción



Petróleo & Gas



Inmobiliaria



Policía, bomberos y  
guardia costera



Drone sports & e-sports

Qué se puede hacer ahora?

# Aplicaciones con UAVs



Manejo de inventarios



Inspección

# Aplicaciones con UAVs



Policía



Drone sports

# Aplicaciones con UAVs



Misiones espaciales



Drone sports

24

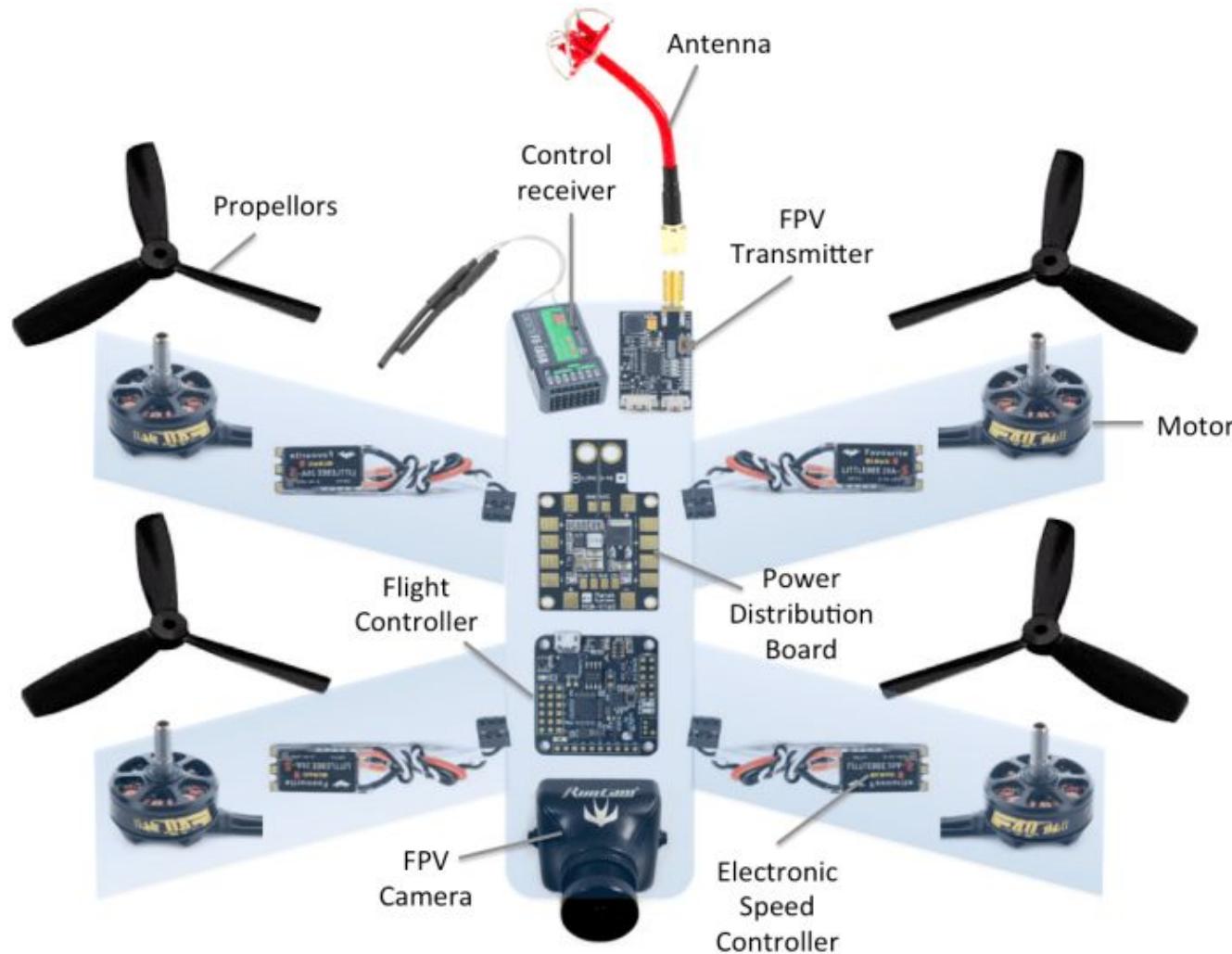
# Aplicaciones con UAVs



25

# Aplicaciones con UAVs

26



¿De que está hecho un drone?

# Aplicaciones con UAVs

Previous 

¿Qué falta ?

27

Next 

# Aplicaciones con UAVs



28

## Controladoras de vuelo

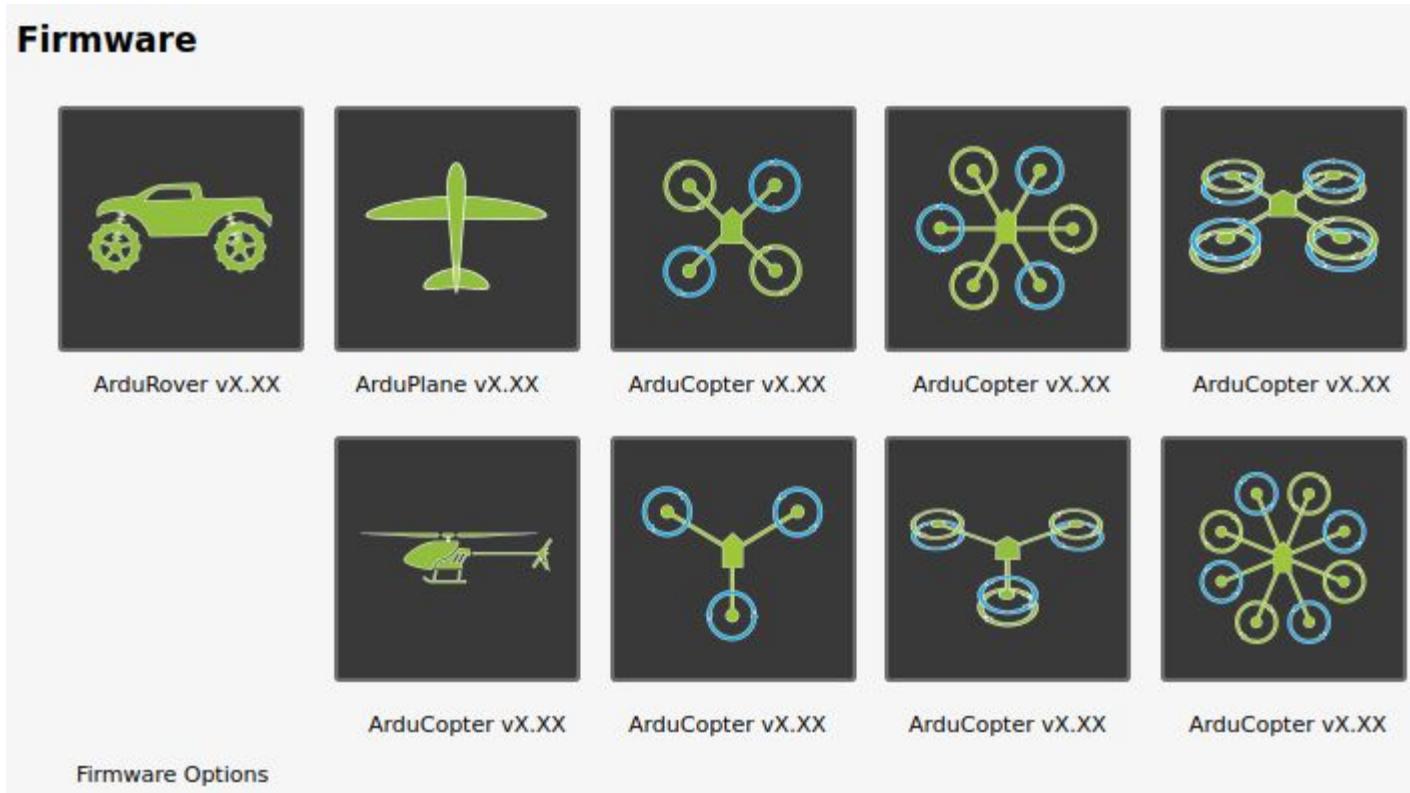
# Aplicaciones con UAVs



<https://ardupilot.org/>

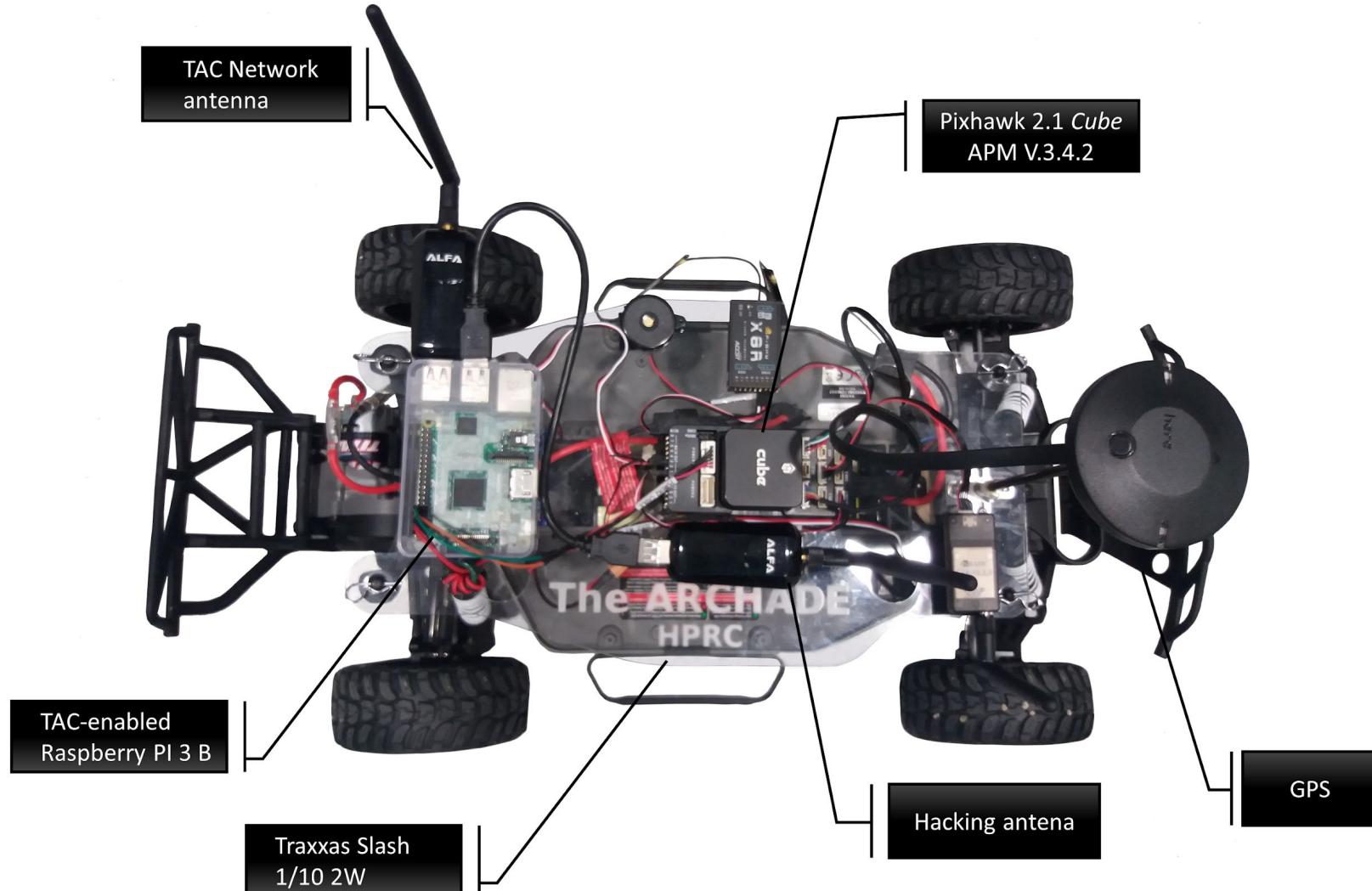
# Aplicaciones con UAVs

30



## APM Planner

# Aplicaciones con UAVs

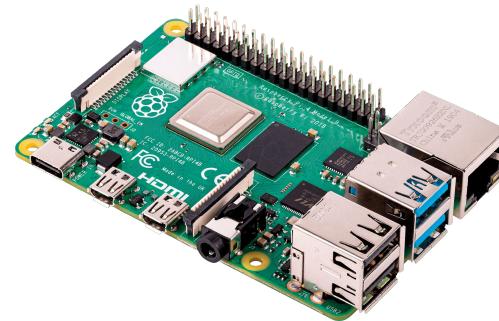


¿Qué falta ?

32

# Aplicaciones con UAVs

## Raspberry PI



USD 35

CPU Quad core Cortex-A72 (ARM v8)  
64-bit SoC @ 1.5GHz  
Memory 2GB, 4GB or 8GB  
LPDDR4-3200 SDRAM

## NVIDIA Nano



GPU 128-core Maxwell  
CPU Quad-core ARM A57 @ 1.43 GHz  
Memory 4 GB 64-bit LPDDR4 25.6 GB/s

USD 100

## Supercomputación embebida

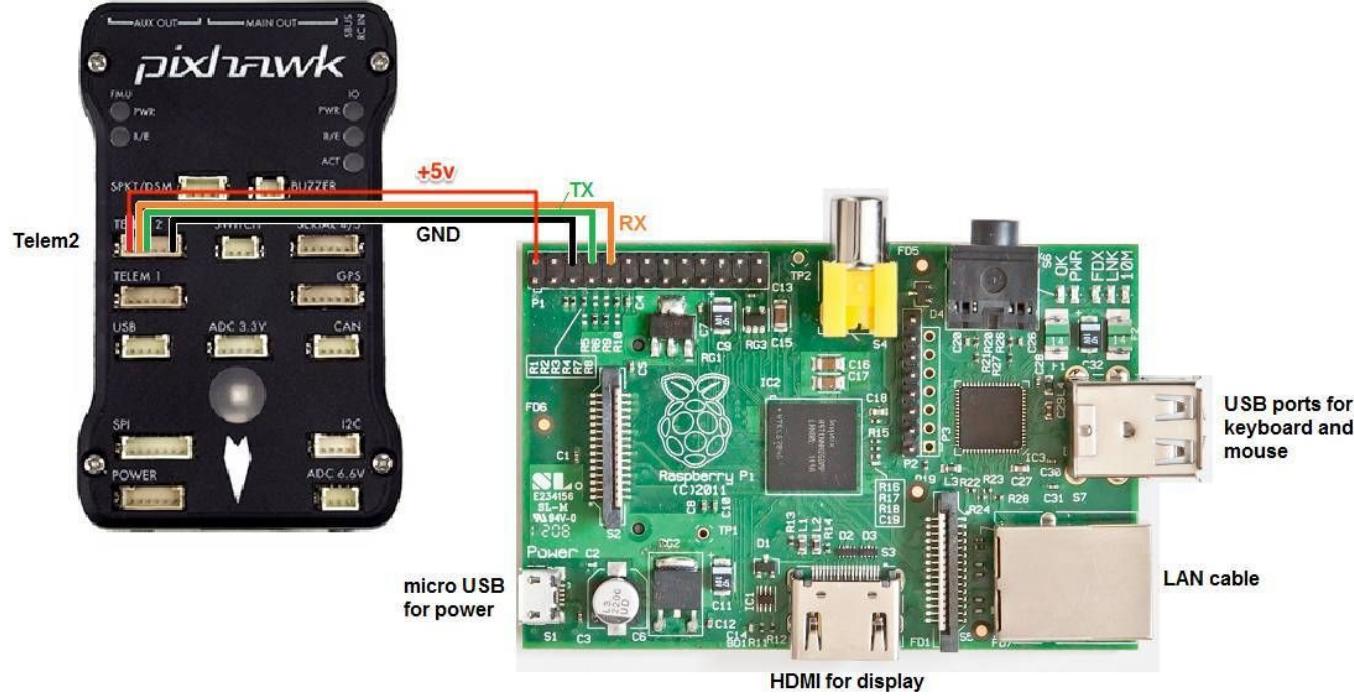


GPU 512-core Volta GPU with Tensor Cores  
CPU 8-core ARM v8.2 64-bit CPU, 8MB L2 + 4MB L3  
Memory 32GB 256-Bit LPDDR4x | 137GB/s

USD 700



# Aplicaciones con UAVs



<https://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html>



# Aplicaciones con UAVs

Previous 

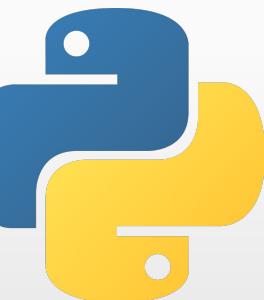
¿Qué falta ?

35

Next 

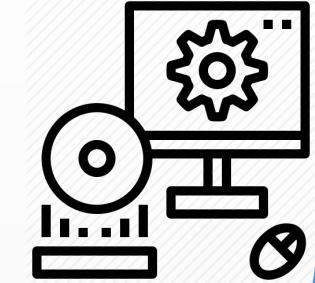
# Aplicaciones con UAVs

## Software!!



The slide features a blue background with a faint image of Earth. In the top right corner is the Club 4.0 logo, which includes a small circular emblem with a star-like pattern and the text "UbiHPC". Below the logo, the word "Club 4.0" is written in large yellow letters, followed by the tagline "Por una Colombia innovadora y competitiva" in smaller white text. In the center, there is a photograph of a quadcopter drone flying over clouds. To the right of the drone is a white rectangular box containing the Python logo and the word "python". At the bottom of the slide, the text "Introducción al desarrollo de software para vehículos aéreos no tripulados (UAVs) inteligentes" is displayed in white, along with the email address "club40ubihpc@gmail.com".

# Aplicaciones con UAVs



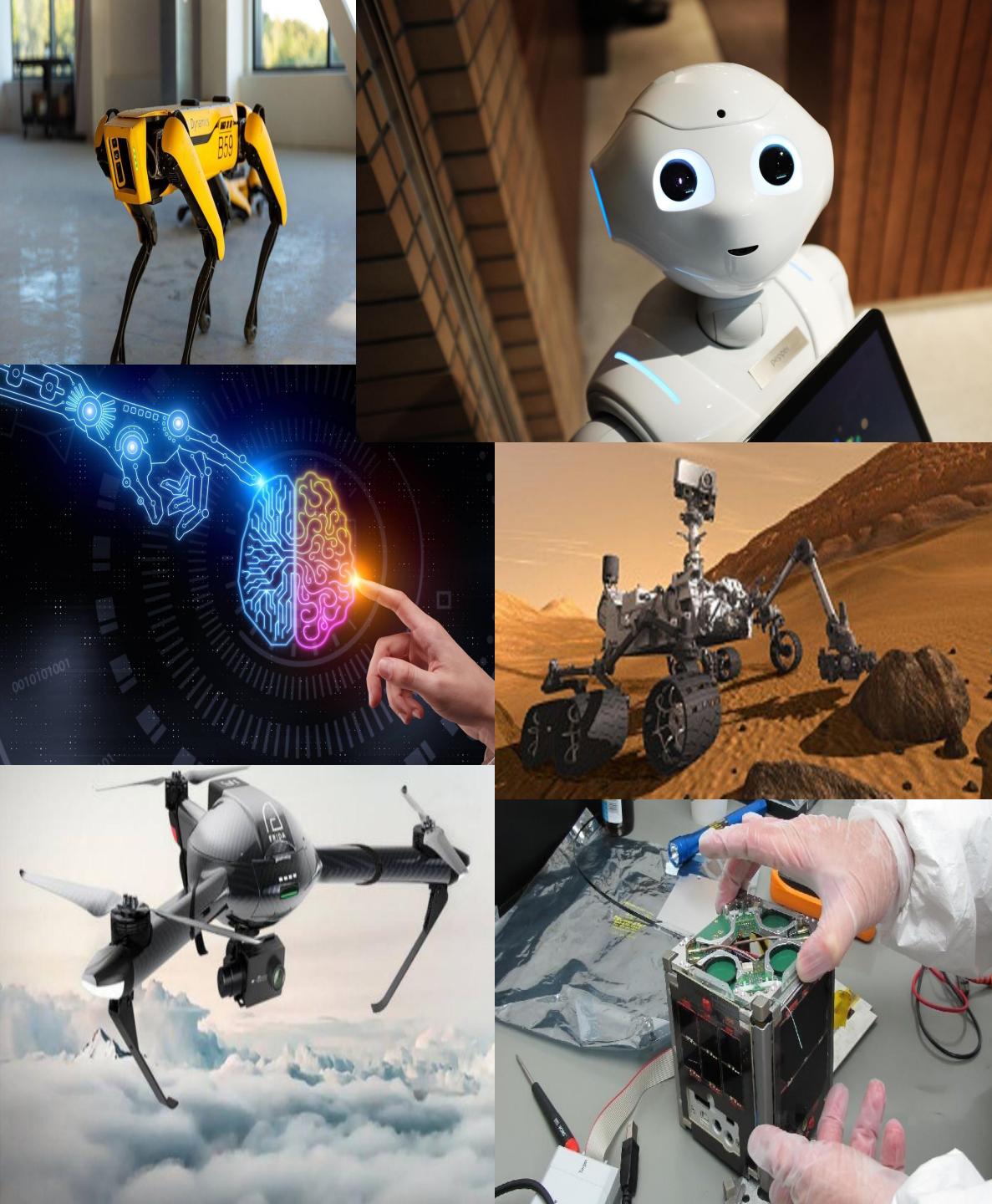
Academia



Empleo



Emprendimiento



# Emprendimiento

38

# Emprender en Colombia?



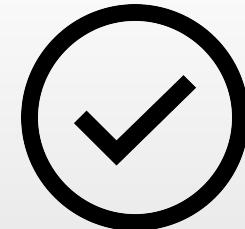
Convertir a Colombia en el Silicon  
Valley de LatinoAmérica en un  
periodo de 10 años



Comunidad



Educación



Casos de éxito



Inversión

# Comunidad



Gobierno



Universidad



Industria

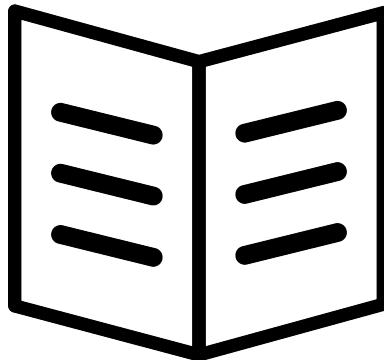


# Educación

Previous 



## Recursos online



- Nuevos cursos
- Semilleros
- Grupos de investigación

## Universidades



**UbiHPC**

- Programación
- Ciencia de datos
- Inteligencia Artificial
- Visión por computador
- Aplicaciones móviles
- Robótica

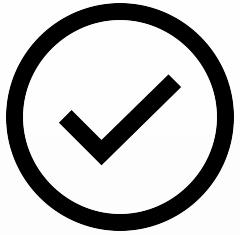
## Club 4.0

41

Next 

# Casos de éxito

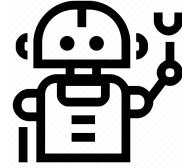
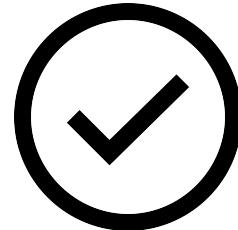
Previous ↑



42

Next ↓

# Casos de éxito



Robótica



Inteligencia  
artificial



IoT



Realidad virtual y  
Realidad aumentada



# Inversión en Colombia

↑ Previous



# Apps.co

44



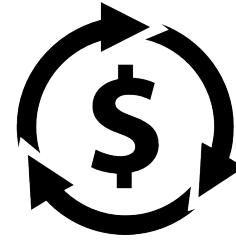
**rockstart.**  
we love startups



Next ↓

# Inversión internacional

↑ Previous



45

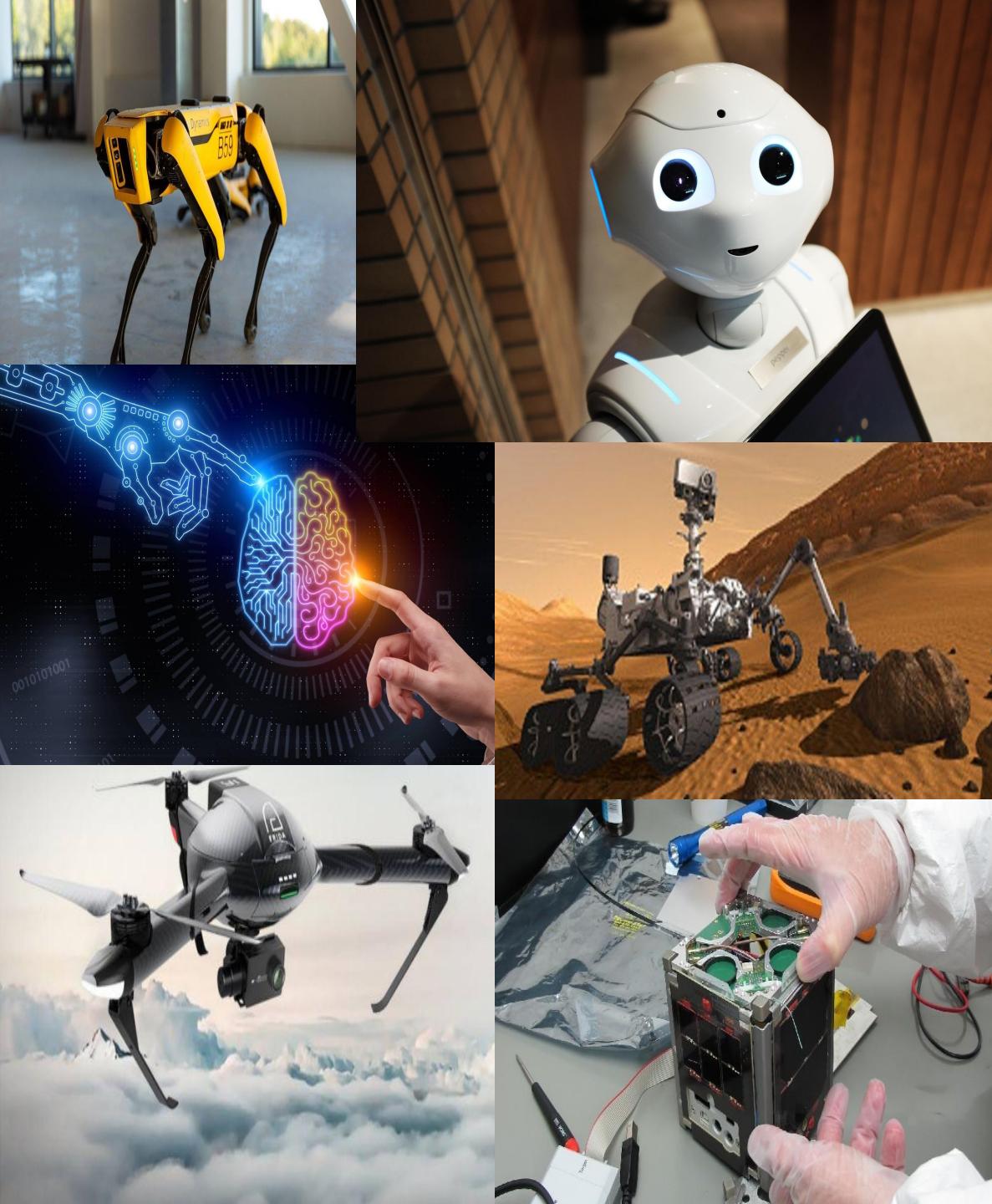
Next ↓

# Emprendimiento nacional e internacional



## Tracción

- Ventas
- Usuarios
- Pilotos
- Artículos científicos?



# Tutorial de Python



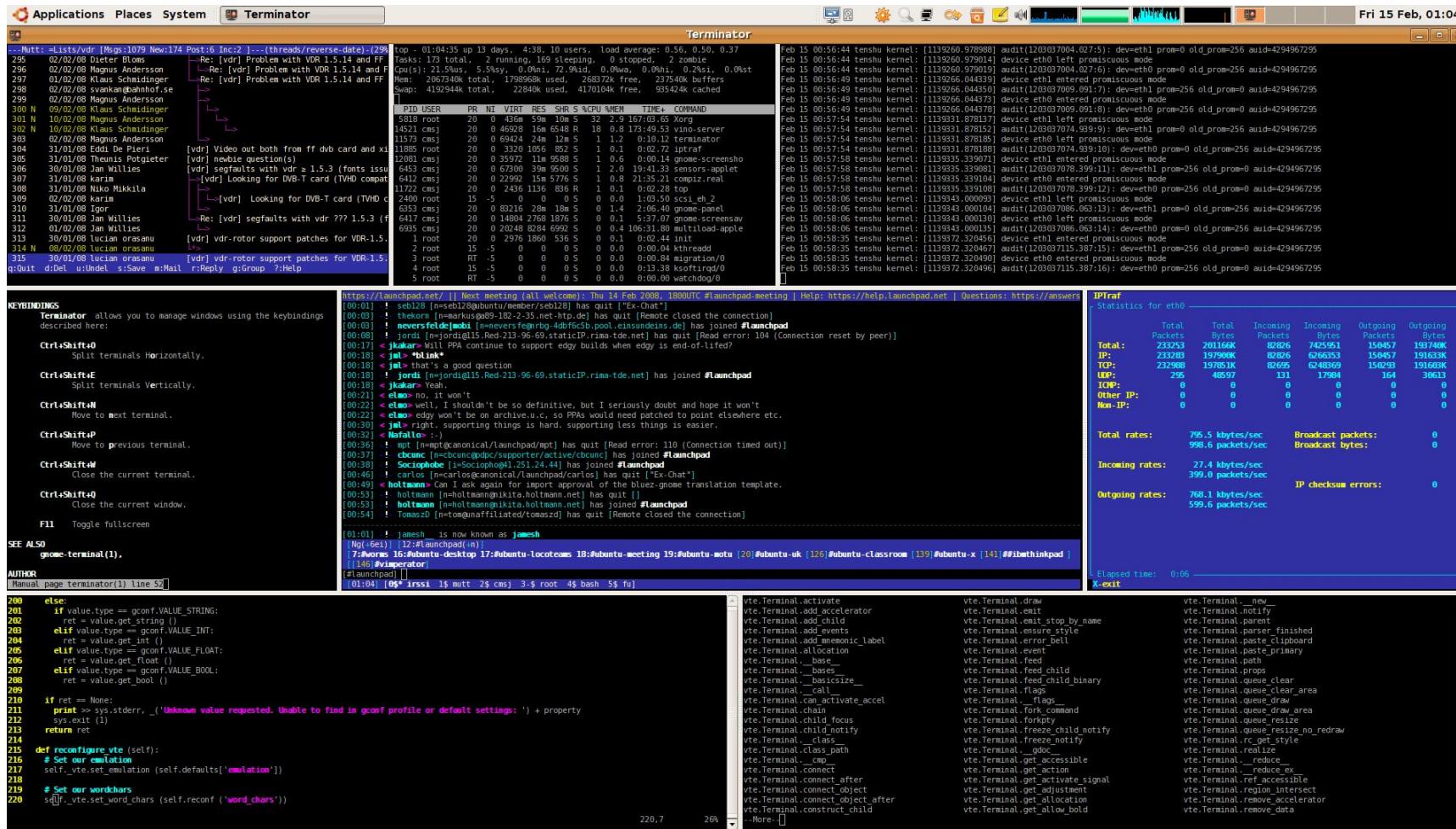
- Lenguaje de scripting (interpretado)
- Multiplataforma: Windows, MacOS, Linux
- Multiparadigma: Imperativo, POO
- Dinámicamente tipado
- Open source
- Modo interactivo (intérprete, Jupyter notebook)
- Requiere indentación

# Tutorial de Python

 Previous

49

Next  
→



```
<sudo> apt-get install terminator
```

# Tutorial de Python

## Configuración inicial

```
$ cd                                     # Moverse al $HOME del usuario  
$ mkdir intro_to_smart_uavs              # Crear directorio intro_to_smart_uavs  
$ cd intro_to_smart_uavs                 # Moverse al directorio intro_to_smart_uavs  
$ mkdir session1 session2 session3 session4 # Crear directorios sesion<1,2,3,4>
```

## Python interpreter

```
$ python  
Python 3.8.1 (default, Sep 1 2020, 22:29:32)  
[GCC 7.3.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```

## Printing

```
>>> print('This is the intro to smart UAVs course')  
>>> session = 1  
>>> print(x)  
>>> print('This is session number '+session) # error  
>>> print('This is session number '+str(session))  
>>> print(f'This is session number {session}') # only in Python3
```

# Tutorial de Python

## Variables

Enteros	Flotantes	Strings	Booleans	Complex	Lists	Dictionaries
\$ x = 2 \$ y = 3	\$ z = 2. \$ w = 7.1	\$ session = 'intro' \$ session = "1"	\$ cool = True \$ finished = False	\$ a = 3.55 + 7.1j	\$ list1 = [1,2,3,4] \$ list2 = ['a','b','c'] \$ list3 = [1,'a',3]	\$ dict1 = {} \$ dict2 = {'x':1,'y':2} \$ dict3 = {1:'x',2:'z'} \$ dict3.keys() \$ dict3.values()

## Ejercicio

\$ x = 2 \$ y = 3 \$ x + y \$ x - y \$ y/x \$ x**y \$ x**1/2 \$ x**(1/2) \$ x%y	\$ list1 = [1,2,3,4] \$ list2 = ['a','b','c'] \$ list4 = list1 + list2 \$ list5 = [i for i in range(10)] \$ print(list5) \$ print(list5[0]) \$ print(list5[-1]) \$ print(list5[:3]) \$ print(list5[2:]) \$ print(list5[4:7])	\$ data = {'name': <your_name>, 'age': <your_age>} \$ print(f"My name is {data['name']}") \$ keys = list(data.keys()) \$ data['location'] = <your_city> \$ print(data) \$ y = { x: x[::-1] for x in [str(i) for i in list(data.values())]}
---	---	---

# Tutorial de Python

## Condicionales

```
$ x = 2
$ if (x ==2):
    print('true')
```

```
$ x = 3
$ if (x ==2):
    print('false')
else:
    print('true')
```

```
$ x = 2
$ if (x <=1): print('X is less or equal than one')
elif x == 2: print('x is equal to two')
else: print('x is greater than two')
```

## For loop

```
$ for i in range(10):
    print(i)
$ list1 = [1,1,2,3,5,8]
$ for i in list1: print(i)
```

## While loop

```
$ x = 0
$ while(x < 9):
    x += 1
```

## Ejercicio

Crear una función que calcule los primeros X elementos de la serie Fibonacci

## Solución

```
def compute_fibonacci(x):
    series = [1]
    while(len(series)<x):
        if len(series) == 1: series.append(series[-1])
        else: series.append(series[-1]+series[-2])
    return series
```

## Functions

```
$ def test(x):
    print(x)
$ def sum(x,y):
    return x+y
$ sum(13,58)
$ def create_list(list_size):
    l = [i+1 for i in range(int(list_size))]
    return l
$ def factorial(x):
    if x == 2: return x
    else: return x*factorial(x-1)
$ factorial(4)
```

# Tutorial de Python

## Scripts

- Un archivo con extensión .py
- Un script se ejecuta línea a línea
- Script shebang: `#!/usr/bin/env python3`
- Ejecutar un script: `python script.py`  
`./script.py` (requiere permisos de ejecución → `chmod +x script.py`)

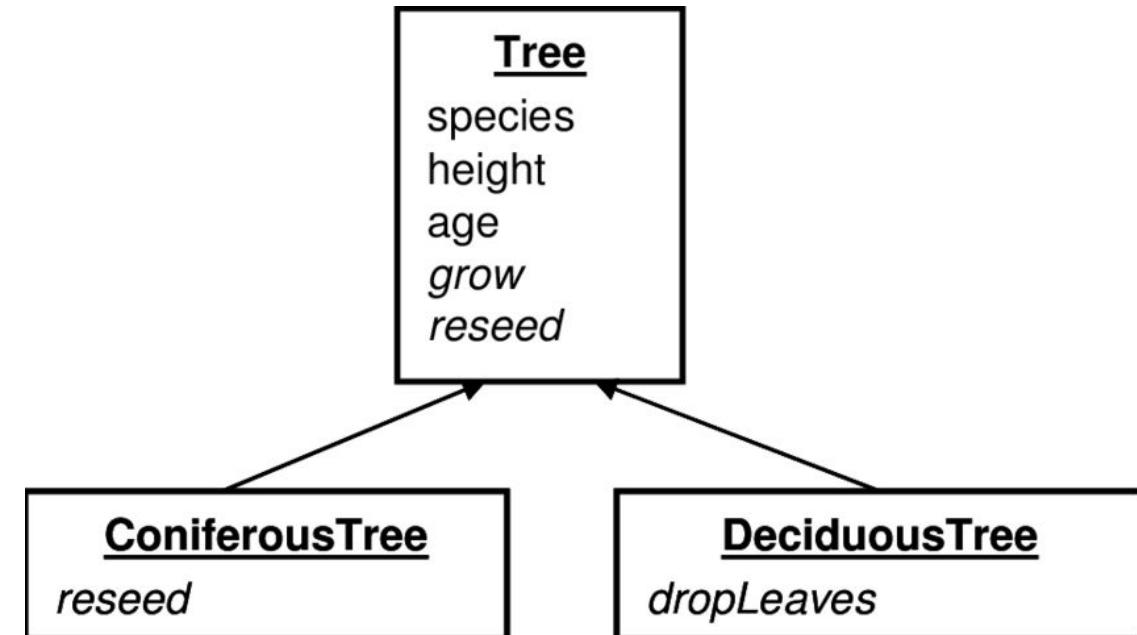
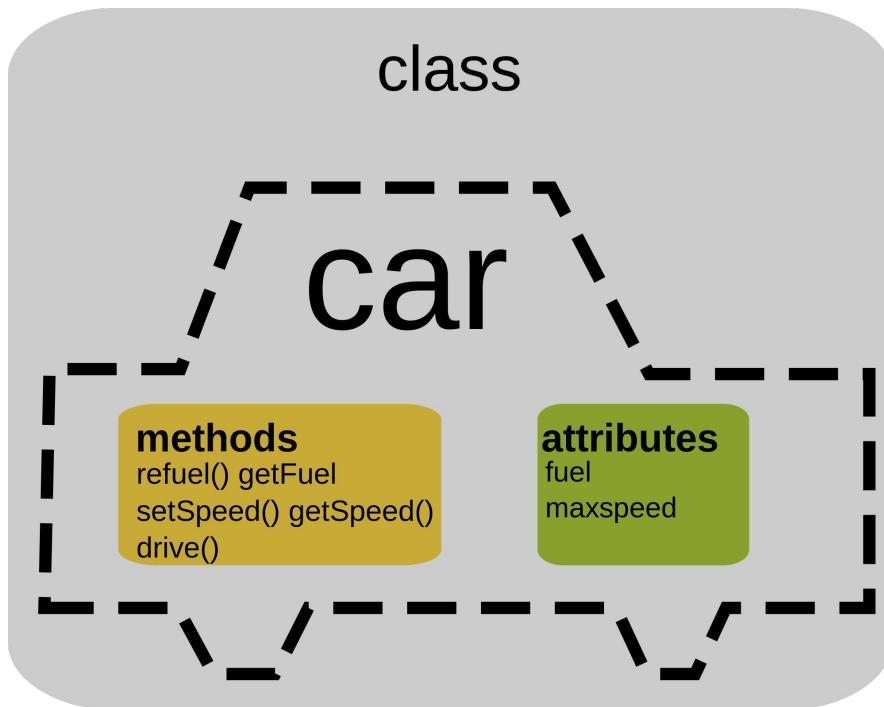
## Practica

53

```
$ sudo apt-get install nano / # apt-get install nano
$ cd
$ nano .bashrc
    Agregar la siguiente linea al final del archivo (ctrl v → Para bajar rápido):
        export PYTHONPATH=$PYTHONPATH:$HOME/intro_to_smart_uavs
$ source .bashrc
$ cd intro_to_smart_uavs
$ touch __init__.py
$ cd session1
$ touch __init__.py drones.py mydrone.py
$ chmod +x drones.py mydrone.py
```

# Tutorial de Python

## Programación orientada a objetos



# Tutorial de Python

## Programación orientada a objetos



```
# -*- coding: utf-8 -*-
from calendar import timegm
from time import gmtime

class Drone:

    """ Drone class """

    def __init__(self, fuel, max_speed):

        """ Initializing Drone object """
        self._fuel = fuel
        self._max_speed = max_speed
        self._speed = 0.

    def add_fuel(self,fuel):

        """ Adding fuel to tank """
        self._fuel += fuel

    def get_fuel(self):

        """ Getting Drone fuel """
        return self._fuel

    def set_speed(self,speed):

        """ Setting Drone speed """
        self._speed = speed if speed < self._max_speed else self._max_speed

    def get_speed(self):

        """ Getting Drone speed """
        return self._speed

    def fly(self):

        """ Flying the drone """
        current_time = timegm(gmtime())
        # Implement the flying method
```

# Tutorial de Python

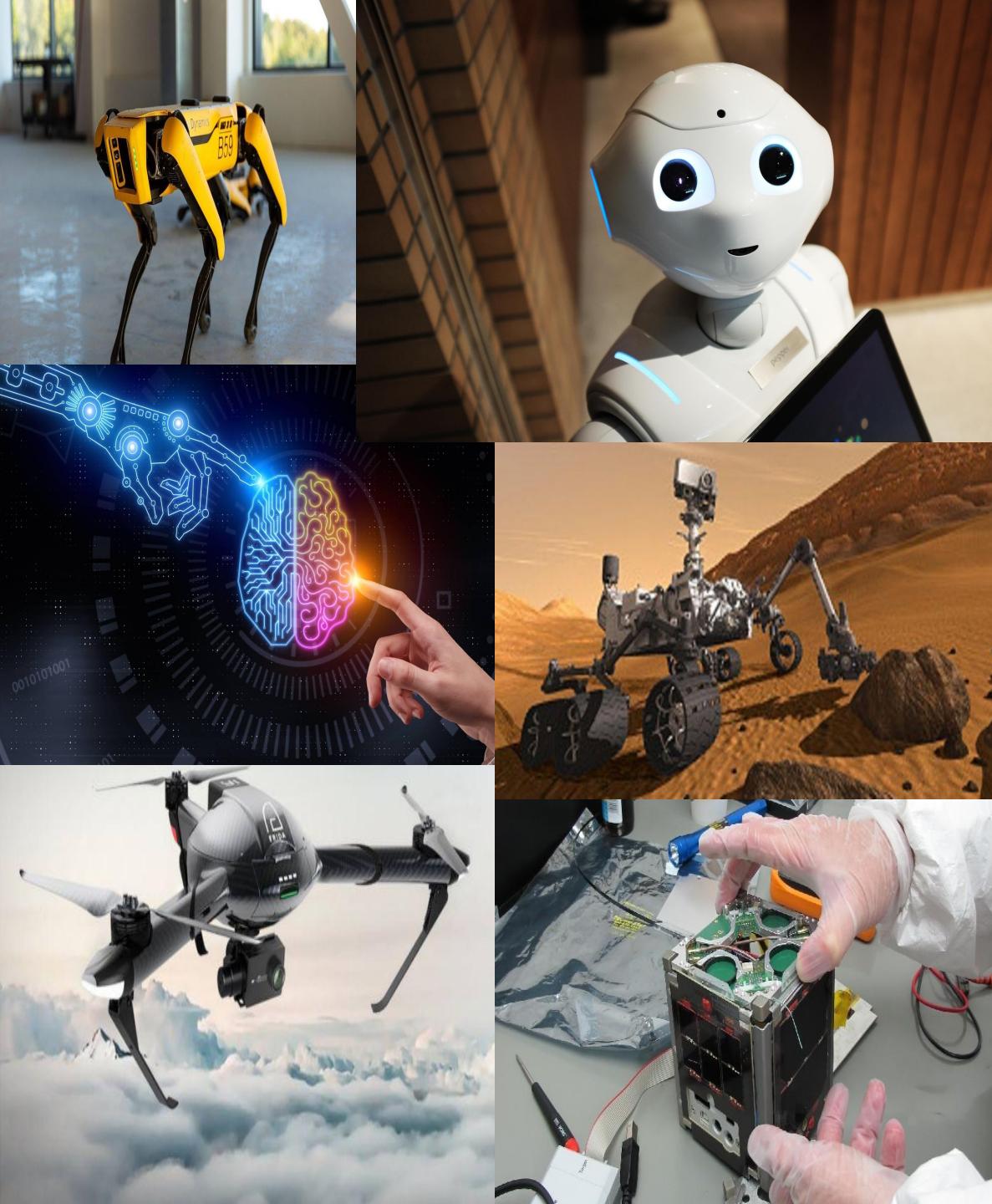
## Programación orientada a objetos



```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
from drones import Drone

fuel = <fuel>
max_speed = <max_speed>
refuel_quantity = <refuel_quantity>
drone = Drone(fuel, max_speed)
###

print('Current fuel is '+str(drone.get_fuel()))
print('Current speed is '+str(drone.get_speed()))
drone.add_fuel(refuel_quantity)
print('Now the fuel is '+str(drone.get_fuel()))
drone.set_speed(<new_speed>)
print('Current speed is '+str(drone.get_speed()))
drone.fly()
```



# Git / GitHub

57



# GitHub

58

<https://github.com/leonardocfor>

Repo: teaching  
Folder: curso\_intro\_smart\_uavs

# Gracias

Leonardo Camargo Forero, Ph.D

✉ [leonardo@ubihpc.com](mailto:leonardo@ubihpc.com)  
🌐 [www.ubihpc.com](http://www.ubihpc.com)

