



# **UbiHPC Club 4.0: Introducción al desarrollo de software para vehículos aéreos no tripulados (UAVs) inteligentes -- Sección I**

Leonardo Camargo Forero, Ph.D

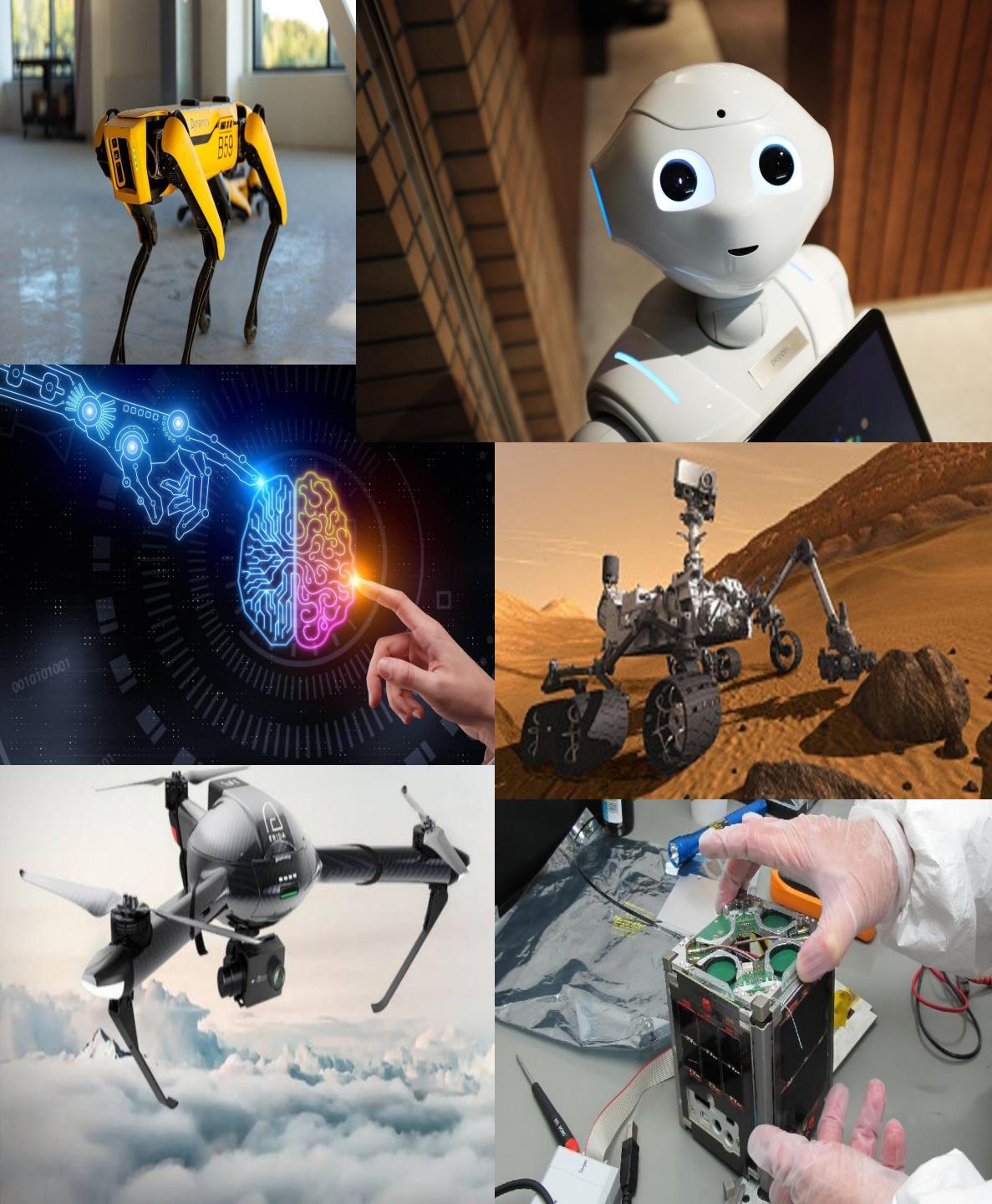
CEO **UbiHPC**  
[www.ubihpc.com](http://www.ubihpc.com)



# Agenda

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

# Introducción



# Acerca de mi

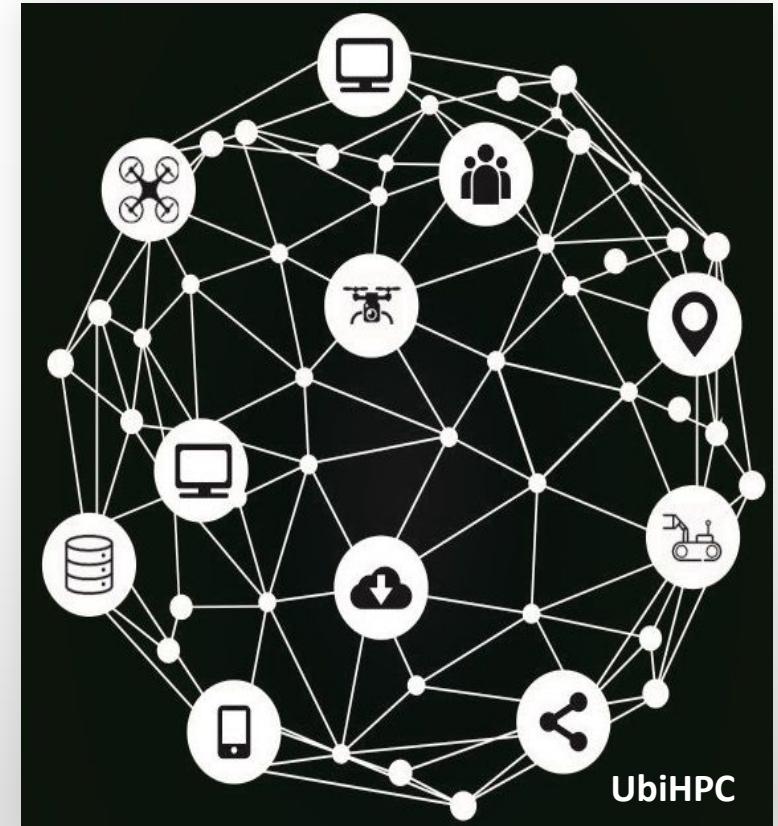
## Educación

- 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

Noviembre 2019



Leaders in Innovation Fellowship  
60 companies in Latin America

# UbiHPC



Analítica de datos e IA

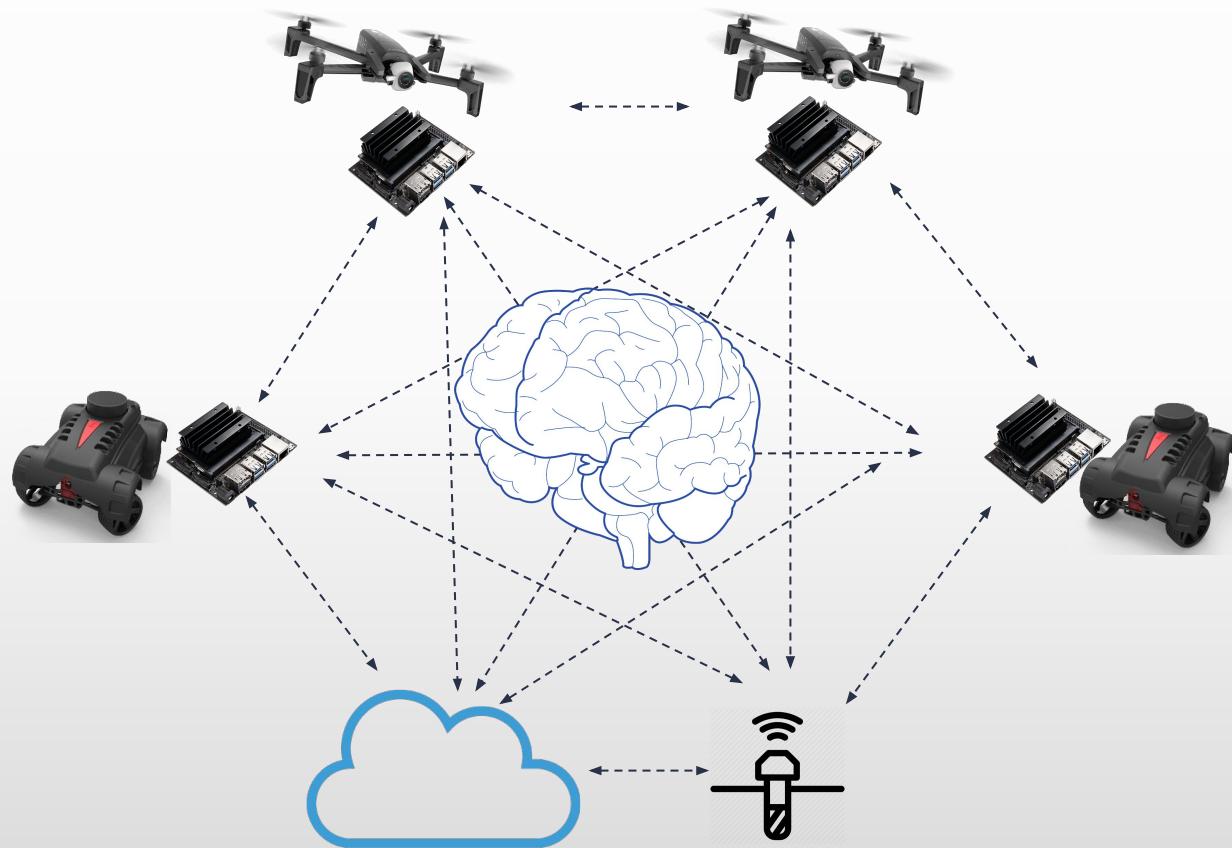


Internet de las Cosas Inteligentes  
(AIoT)



Sello de música genética

# The ARCHADE



Inteligencia colectiva

# 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



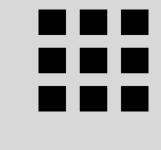
SimPlat



SaaS



Desktop app



Apps market

The ARCHADE software suite

# Detector y predictor de tormentas eléctricas (The ARCHADE)

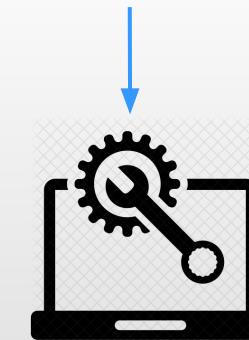


The ARCHADE SDK - AI (CNN)

Detección de rayos      Predicción de clima



50K USD/hora



The ARCHADE SDK - AI (RNN)

Predicción de dónde y cuándo  
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

10

# 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)

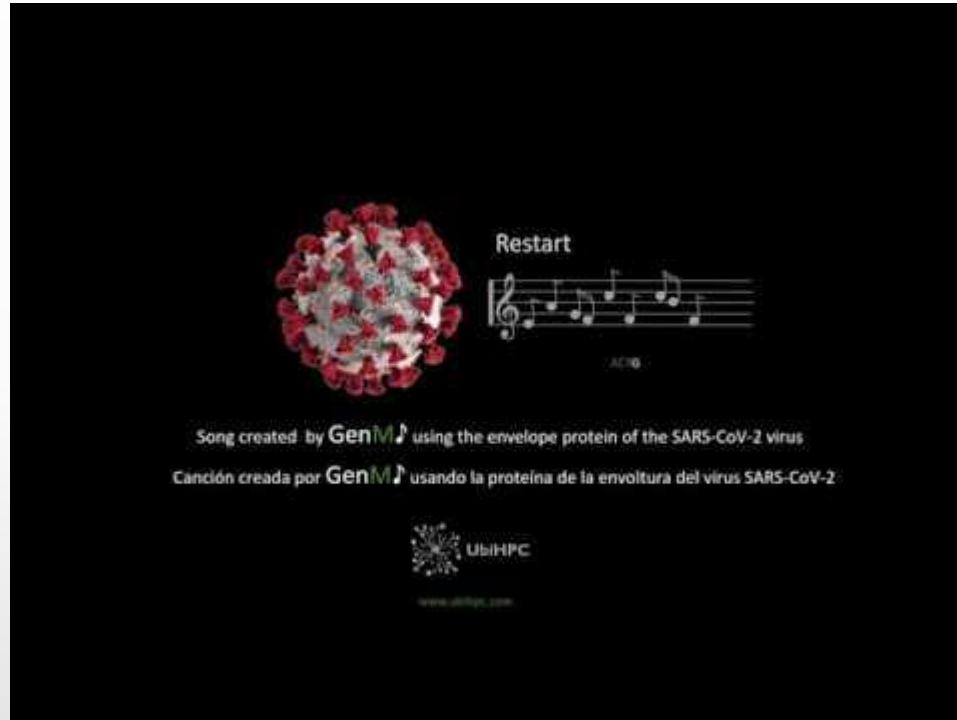


Jingle para empresa  
(Aguacate)

## Canciones especiales

Mascotas, personas, etc

# GenM



# Club 4.0

club40ubihpc@gmail.com

**Club 4.0**  
Por una Colombia innovadora y competitiva

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

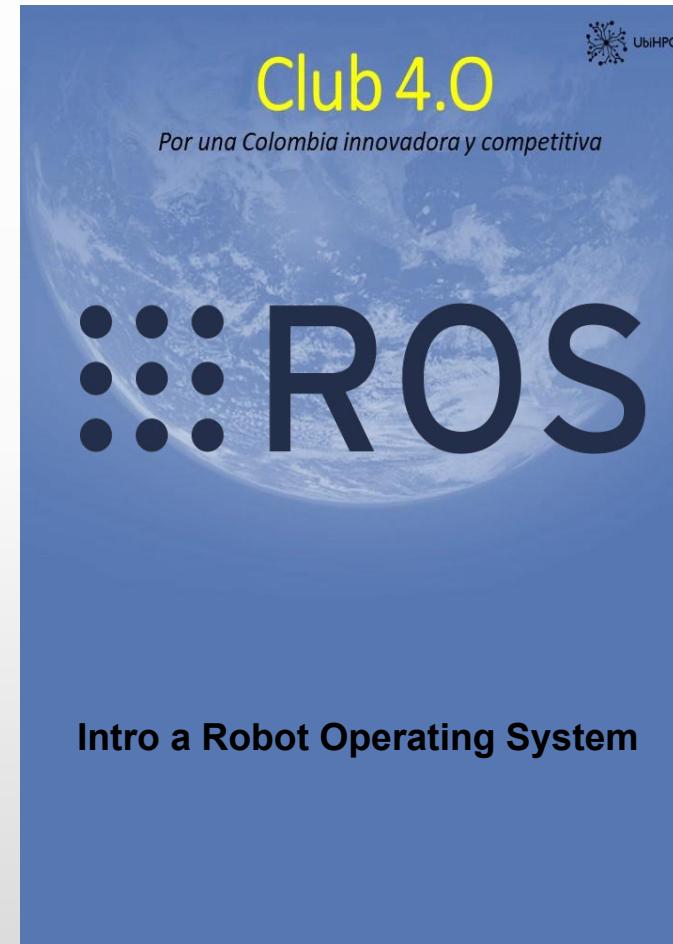
**Club 4.0**  
Por una Colombia innovadora y competitiva

Inteligencia Artificial y Ciencia de datos

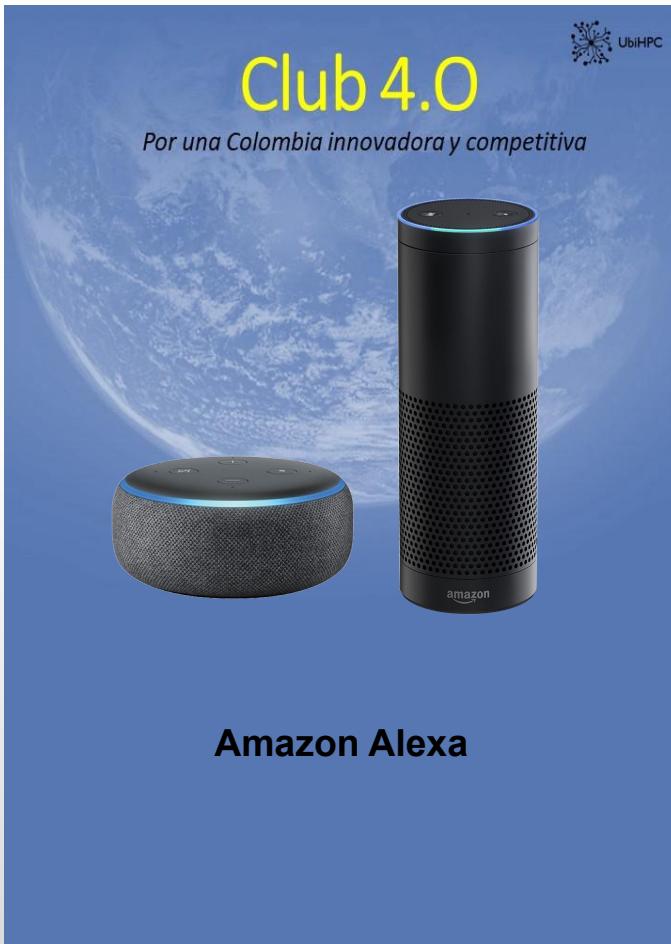
# Club 4.0



club40ubihpc@gmail.com



# Club 4.0



club40ubihpc@gmail.com



15

# 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 de librerías Python para control de piloto automático.
- Tutorial de DroneKit
- Tutorial de Amazon Web Services

## Sección 3 (4 horas)

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

## Sesión 4 (3 horas)

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



Charla aplicaciones con UAVs y oportunidades en Universitat Politècnica de Catalunya

17



Charla integración de UAVs en espacio aéreo y oportunidades en Cranfield university



Charla SubT, Landing on an asteroid using computer vision y oportunidades en NASA Jet Propulsion Laboratory

# Acerca de la sección número 1

## Teoría

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

## Práctica

- Tutorial de Python
- Tutorial de Git y GitHub

## Tareas

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

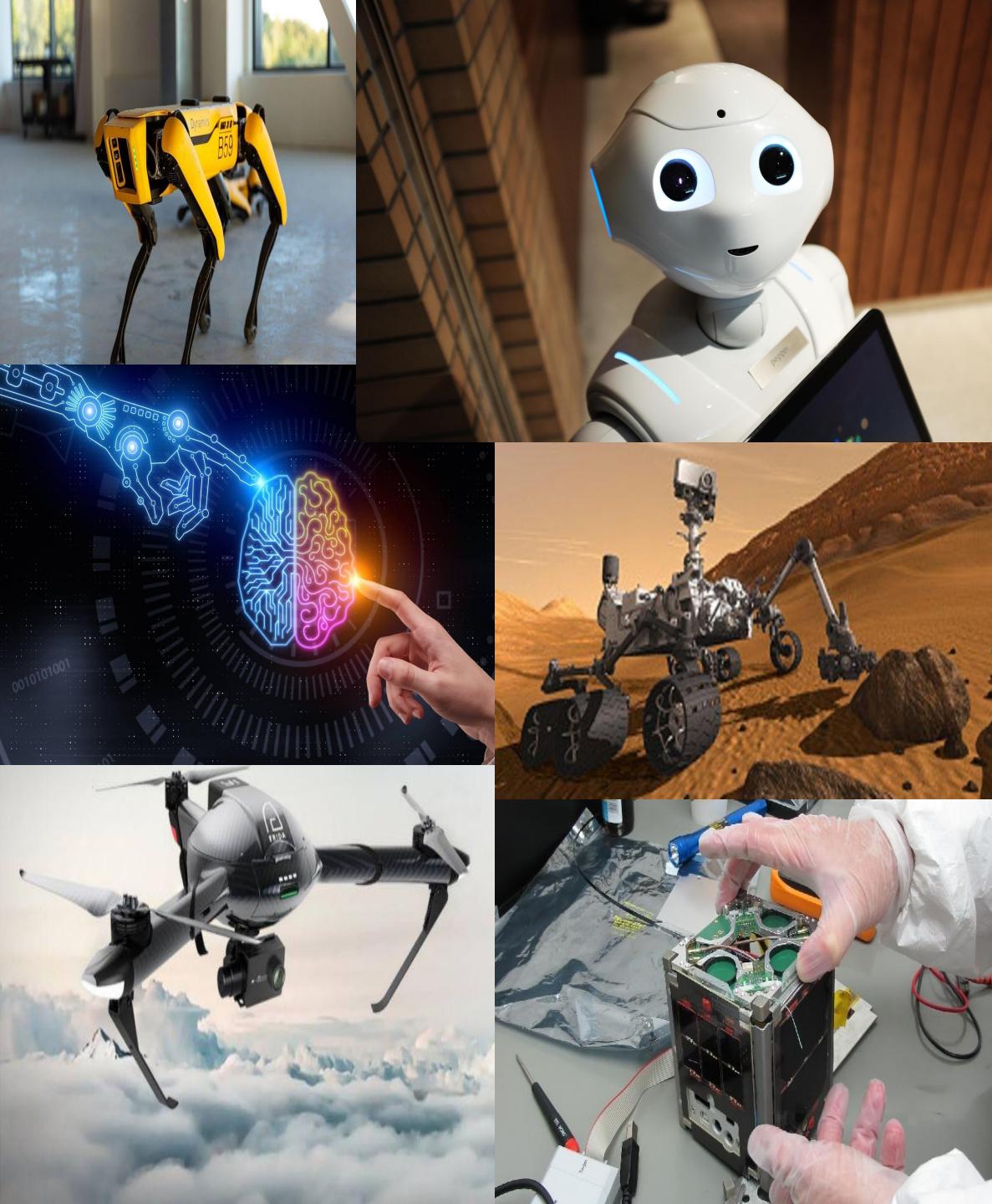


# Felicitaciones !!!

# ¿Como están con la computación ?

20

# Aplicaciones con UAVs



# Aplicaciones con UAVs



22

**Uso tradicional**

# Aplicaciones con UAVs



Agricultura de precisión



Búsqueda y Rescate



Monitoreo/vigilancia



Construcción



Petróleo & Gas



Inmobiliaria



Policía, bomberos y  
guardia costera

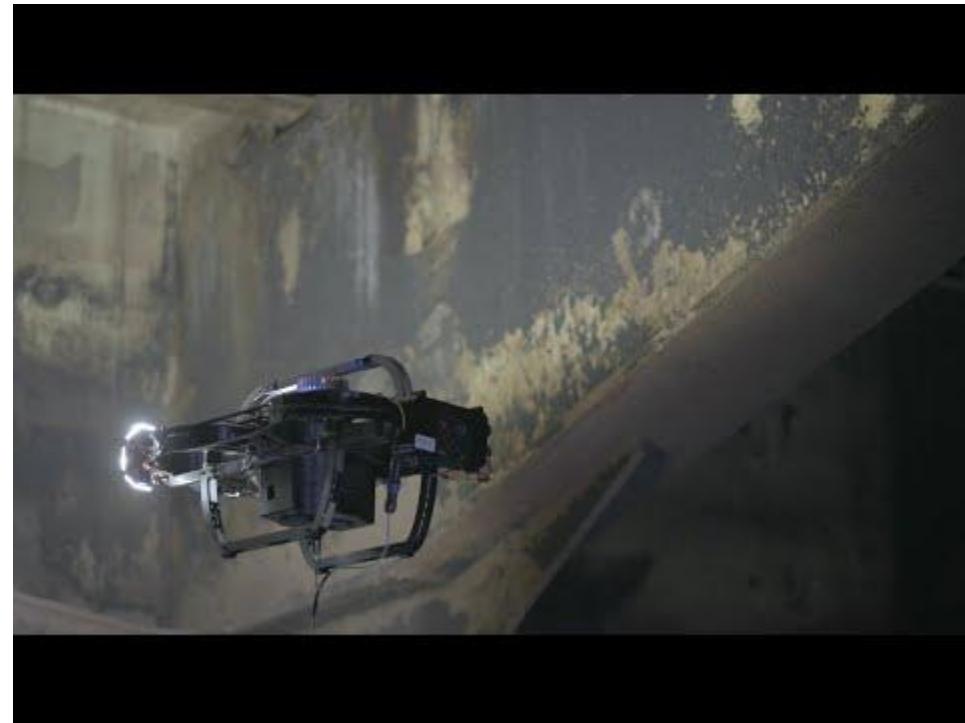


Drone sports & e-sports

# Aplicaciones con UAVs



Manejo de inventarios



Inspección

24

# Aplicaciones con UAVs



Policía



Drone sports

# Aplicaciones con UAVs



Misiones espaciales



DARPA SubT

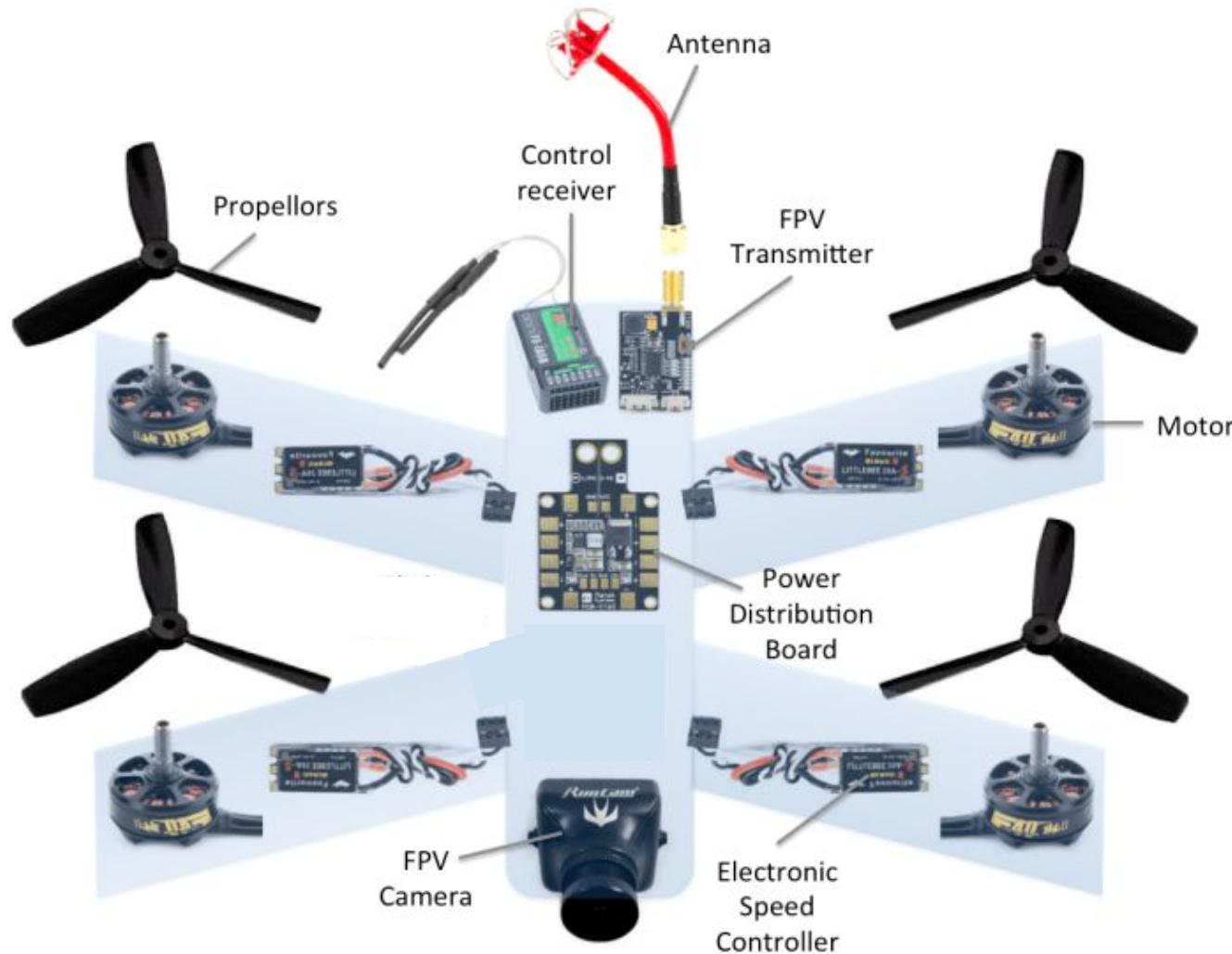
26

# Aplicaciones con UAVs



27

# Aplicaciones con UAVs



¿De que está hecho un drone?

# Aplicaciones con UAVs

Previous 

¿Qué falta ?

29

Next 

# Aplicaciones con UAVs



30

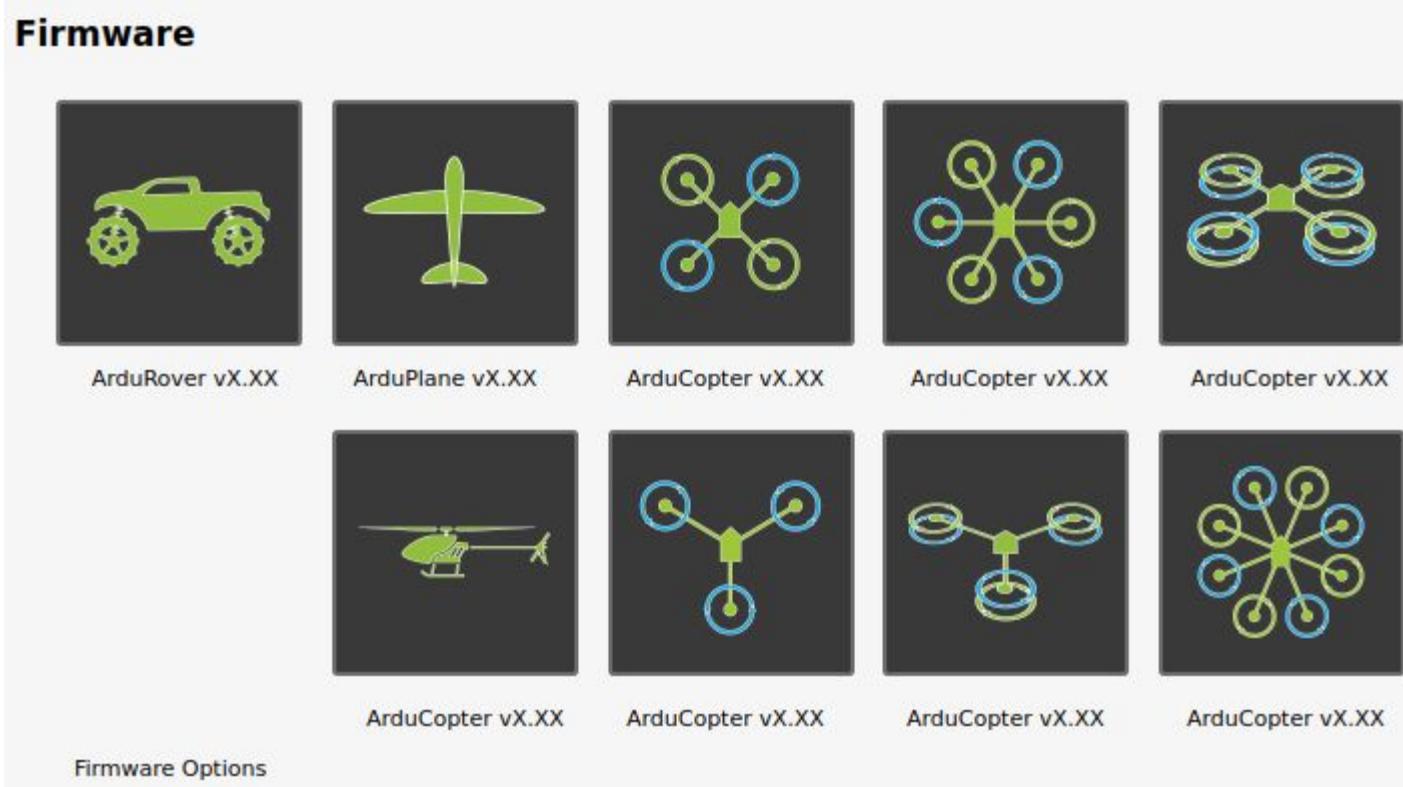
## Controladoras de vuelo

# Aplicaciones con UAVs



<https://ardupilot.org/>

# Aplicaciones con UAVs

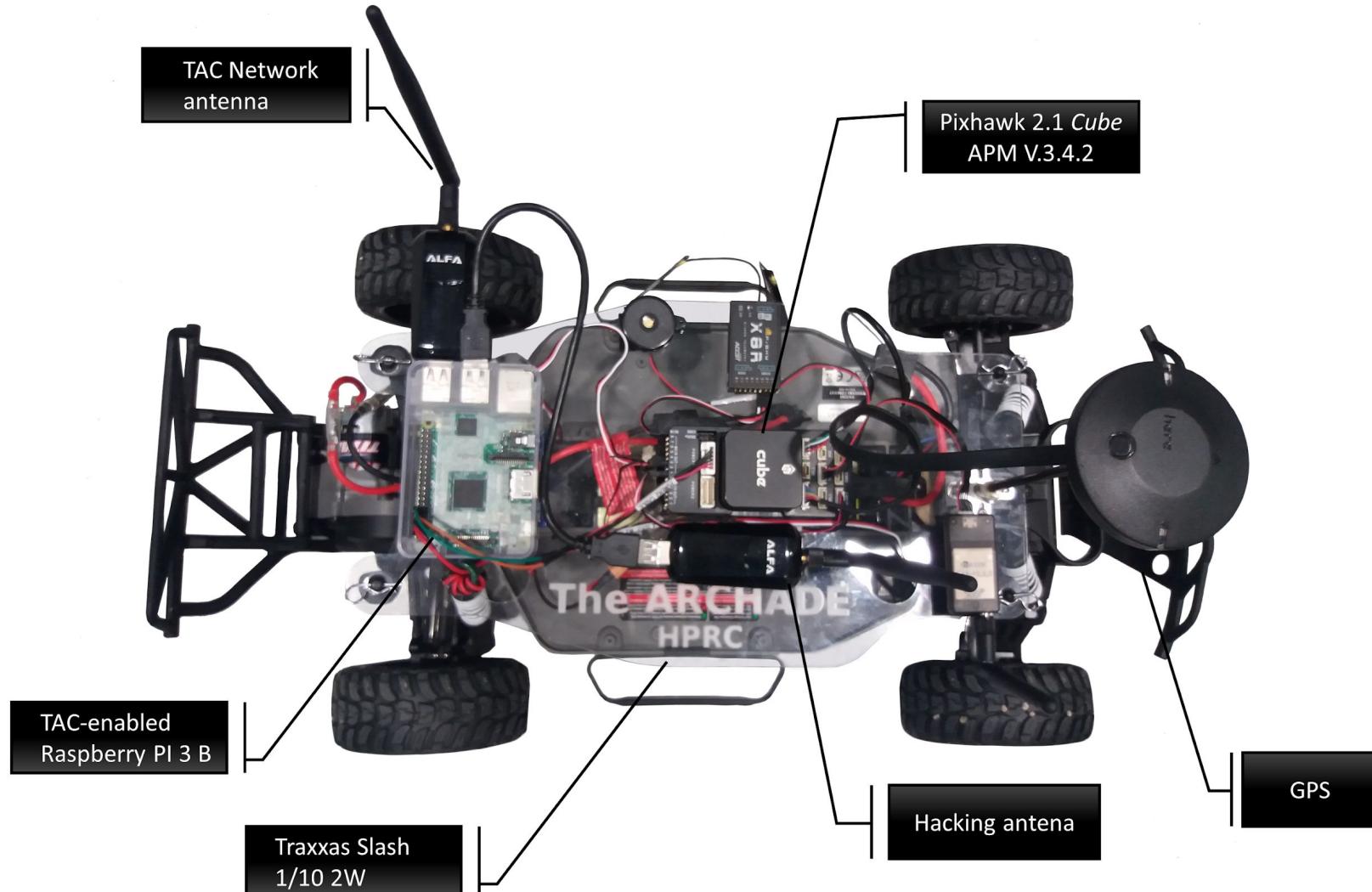


32

## APM Planner

# Aplicaciones con UAVs

33



# Aplicaciones con UAVs

↑ Previous

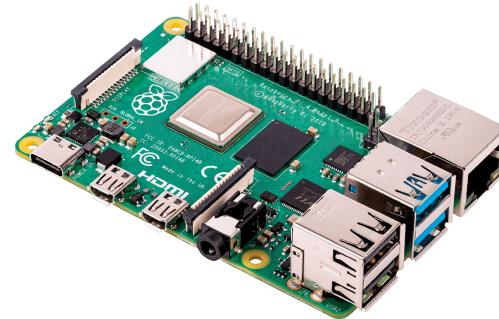
¿Qué falta ?

34

Next →

# 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



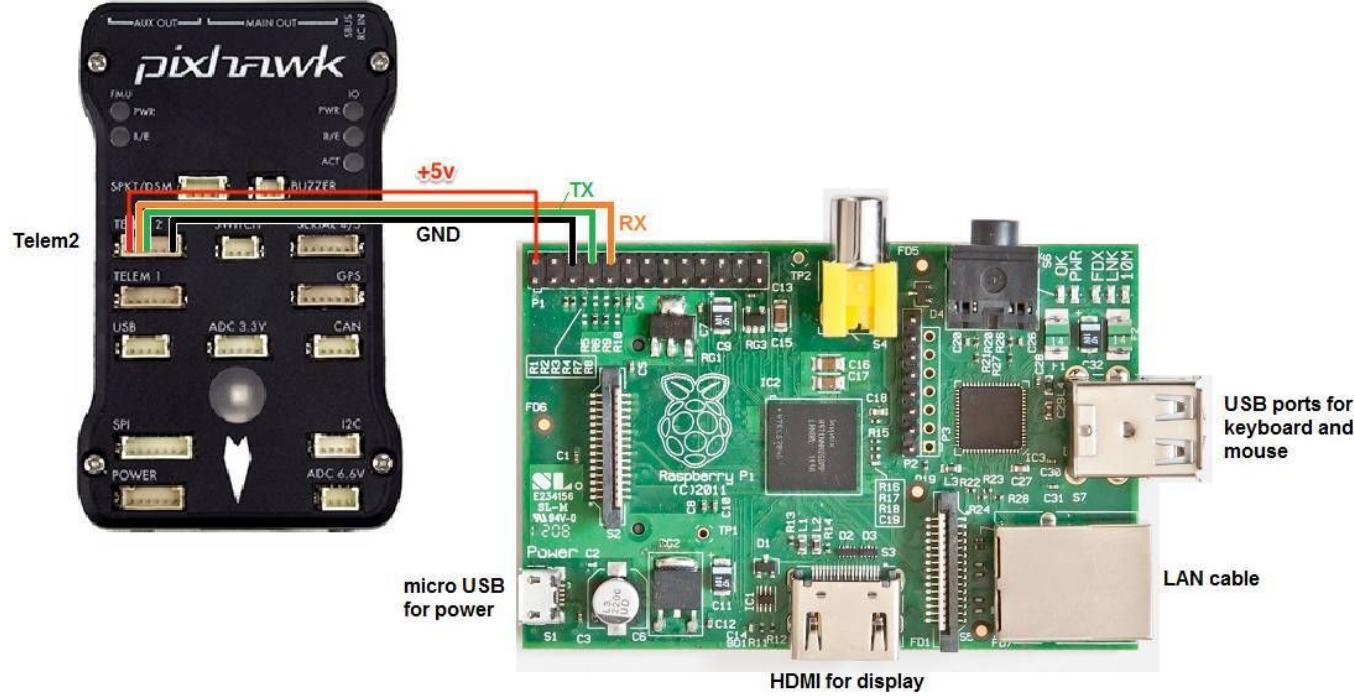
GPU 512-core Volta GPU with Tensor Cores  
CPU 8-core ARM v8.2 64-bit CPU, 8MB L2 + 4MB L3  
Memory 32GB 256-Bit LPDDR4x | 137GB/s

USD 700



35

# Aplicaciones con UAVs



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



# Aplicaciones con UAVs

Previous 

¿Qué falta ?

37

Next 

# Aplicaciones con UAVs

## Software!!

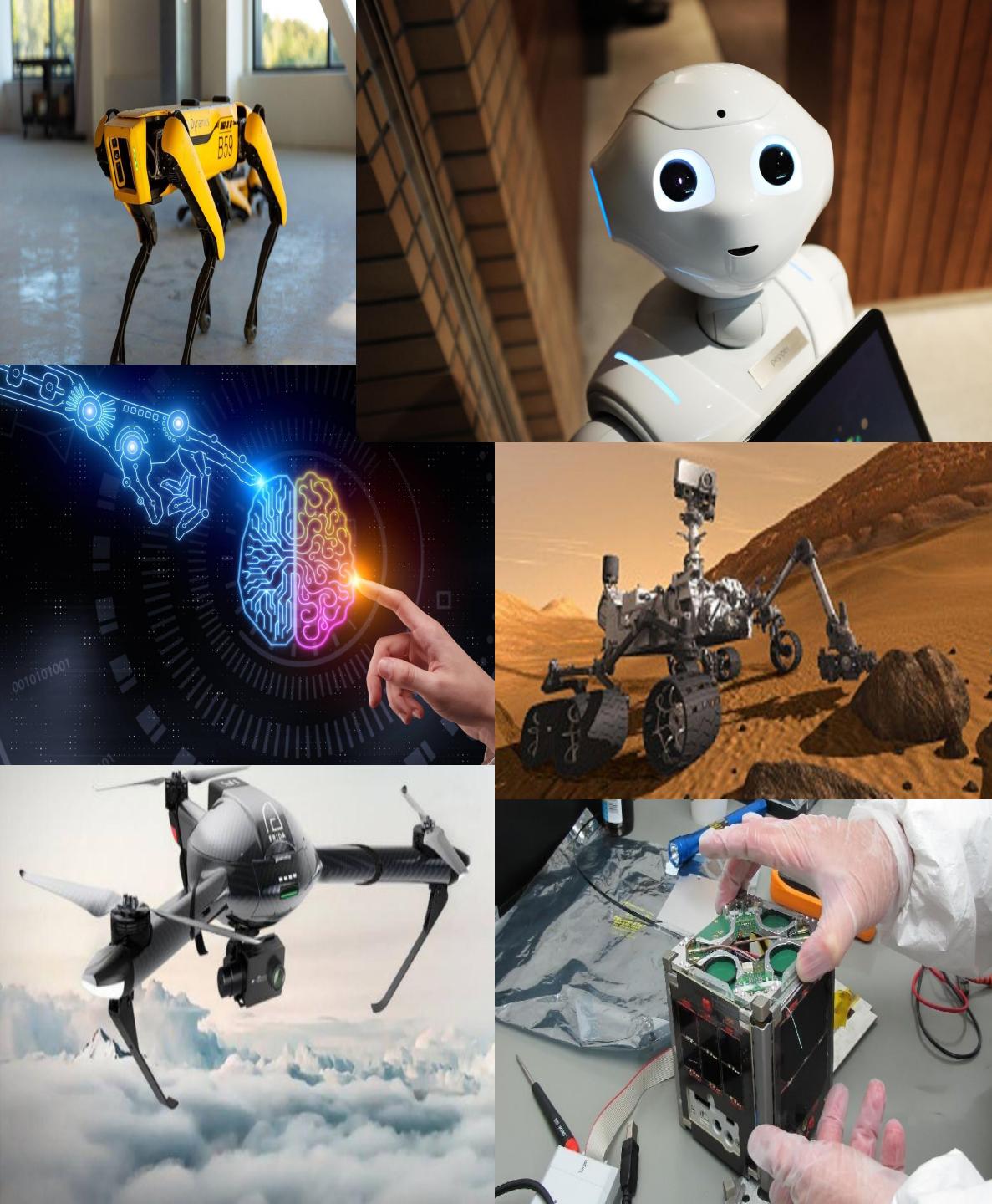


Club 4.0  
Por una Colombia innovadora y competitiva



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

club40ubihpc@gmail.com



# Emprendimiento

39

# Emprendimiento



Academia



Empleo



Emprendimiento

# Emprender en Colombia?



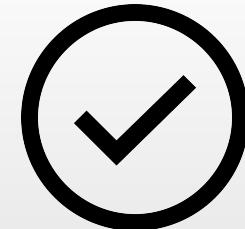
Convertir a Colombia en el Silicon Valley de LatinoAmérica en un periodo de 10 años



Comunidad



Educación



Casos de éxito



Inversión

# Comunidad



Gobierno



Universidad



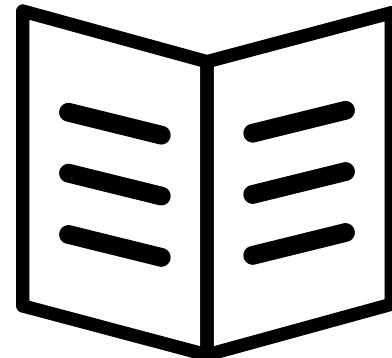
Industria

# Educación

↑ Previous



## Recursos online



- Nuevos cursos
- Semilleros
- Grupos de investigación

## Universidades



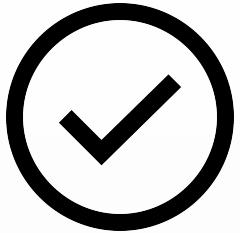
- Programación
- Ciencia de datos
- Inteligencia Artificial
- Visión por computador
- Aplicaciones móviles
- Robótica

## Club 4.0



# Casos de éxito

Previous ↑



44



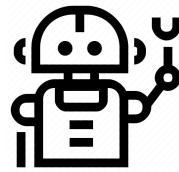
Next ↓

# Casos de éxito

↑ Previous

45

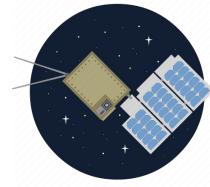
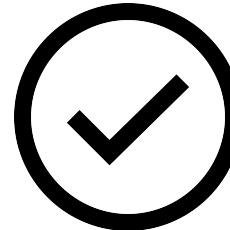
Next ↓



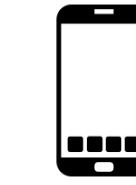
Robótica



Inteligencia  
artificial



Espacio



Aplicaciones  
móviles



IoT



Realidad virtual y  
Realidad aumentada



# Inversión en Colombia

↑ Previous



# Apps.co

46



**rockstart.**  
we love startups



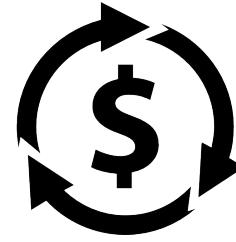
Next ↓

# Inversión internacional

↑ Previous



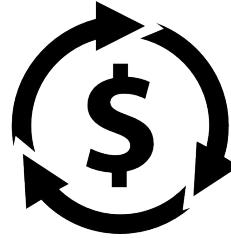
**500**  
startups



47

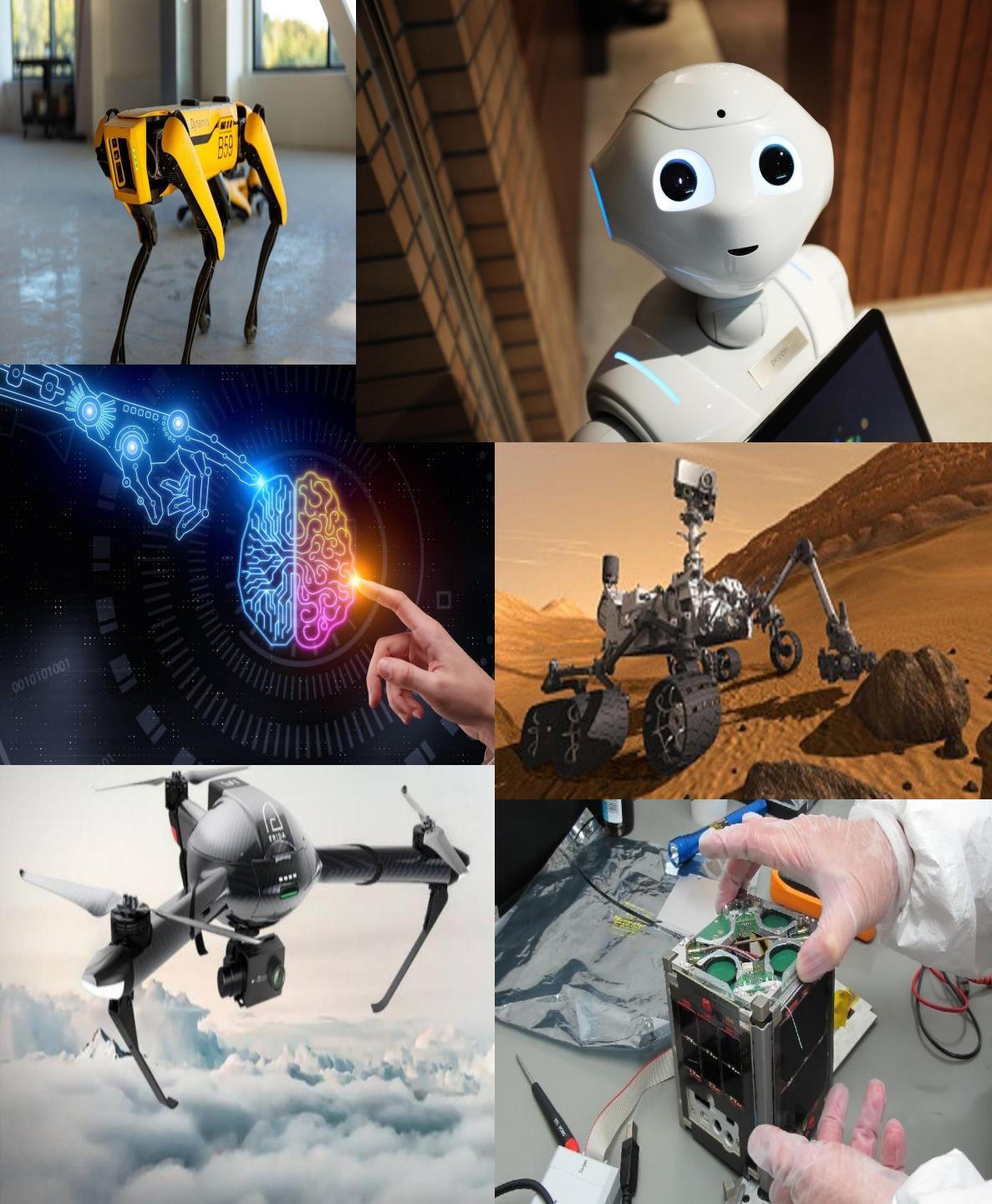
Next ↓

# Emprendimiento nacional e internacional

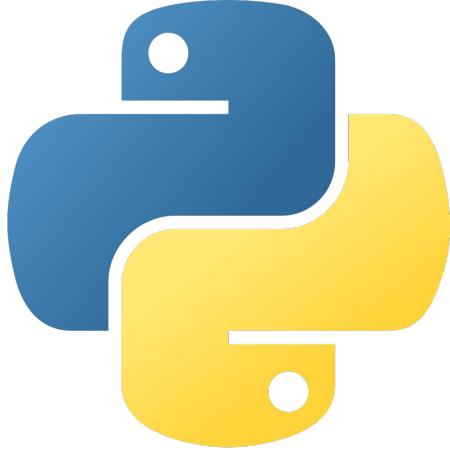


## Tracció n

- 
- Ventas
  - Usuarios
  - Pilotos
  - Artículos científicos?

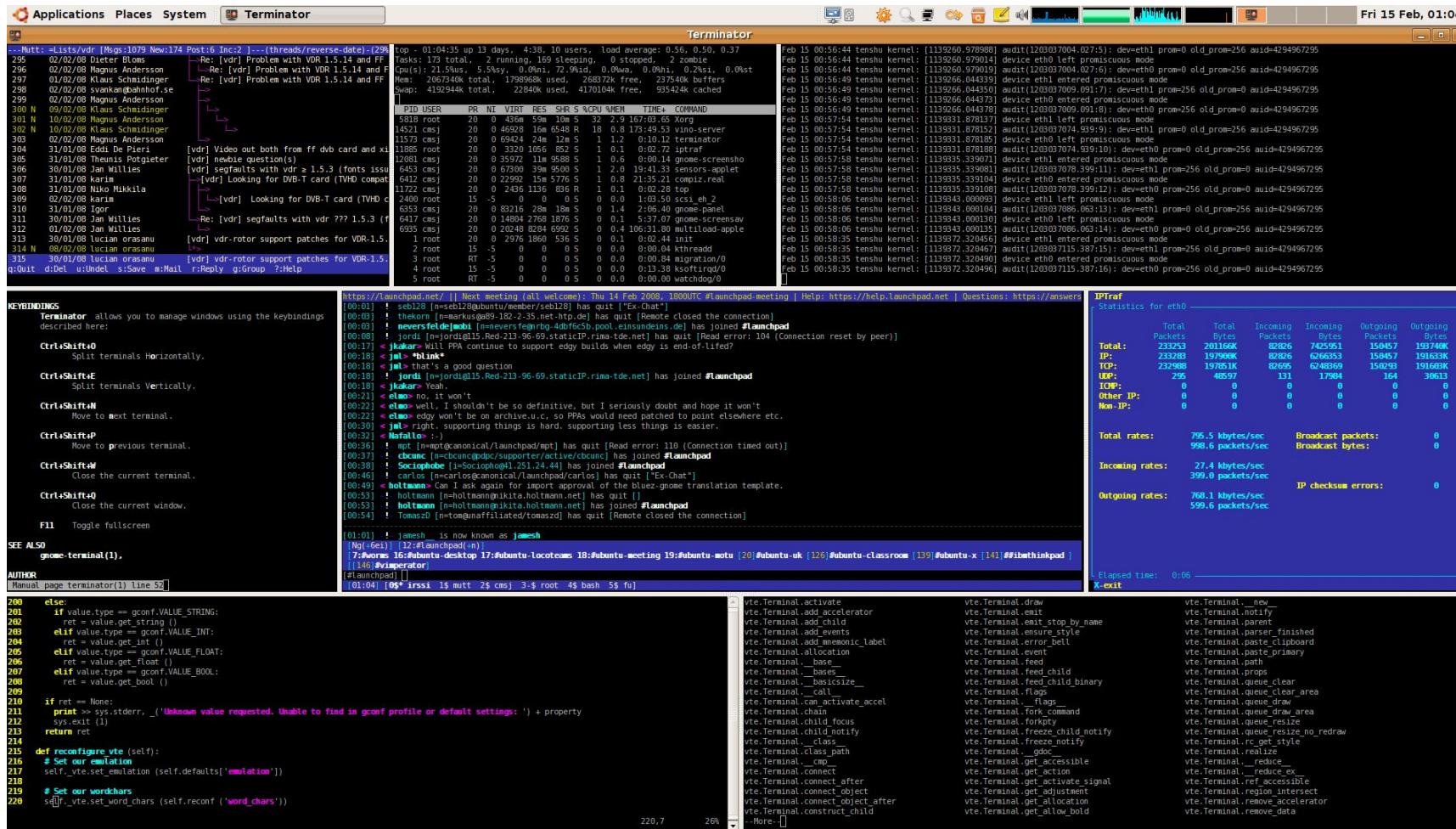


# Tutorial de Python



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

# Tutorial de Python



<sudo> apt-get install terminator

sudo apt-get update

# 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

```
$ python3  
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	\$ z = 2.	\$ session = 'intro'	\$ cool = True	\$ 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', 'z':3} \$ dict3.keys() \$ dict3.values()
\$ y = 3	\$ w = 7.1	\$ session = "1"	\$ finished = False			

## Ejercicio

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

# 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`)

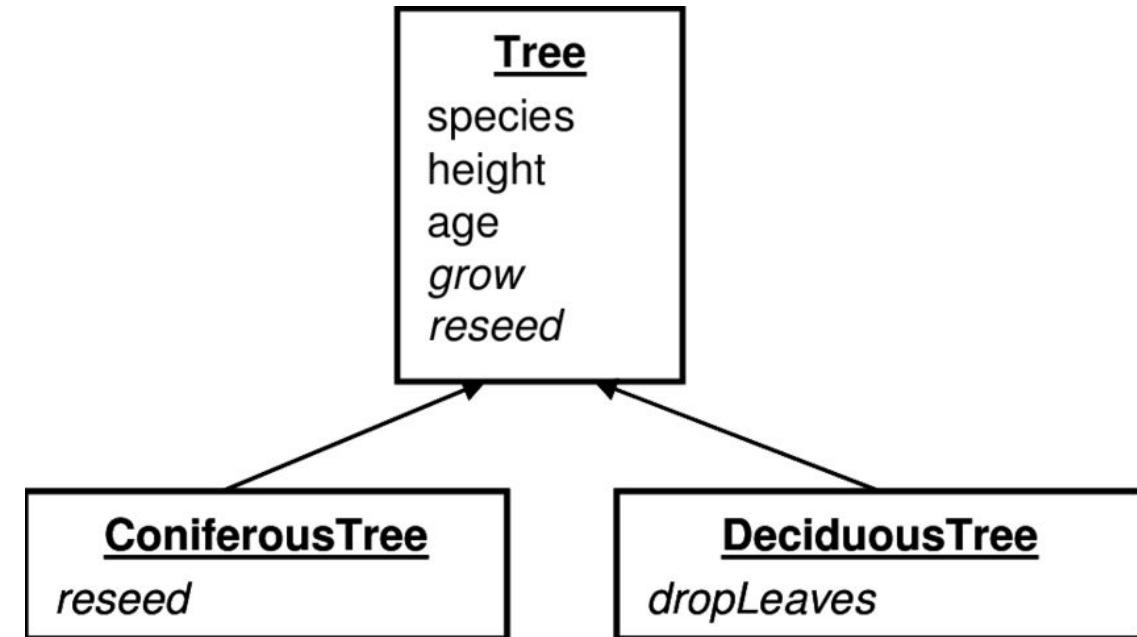
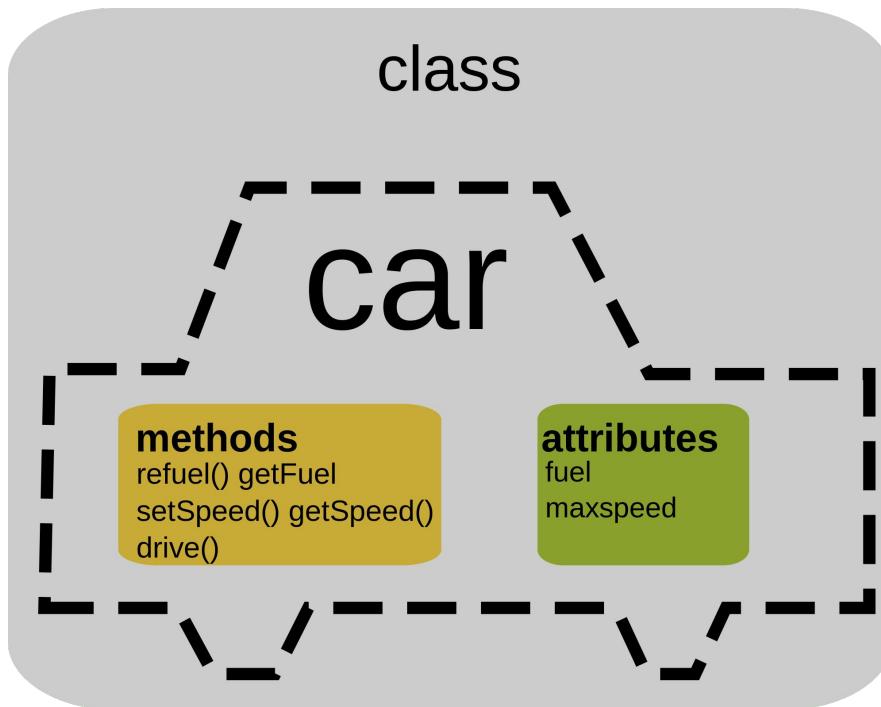
## Practica

55

```
$ 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
$ cd .. && touch session2/__init__.py session3/__init__.py session4/__init__.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

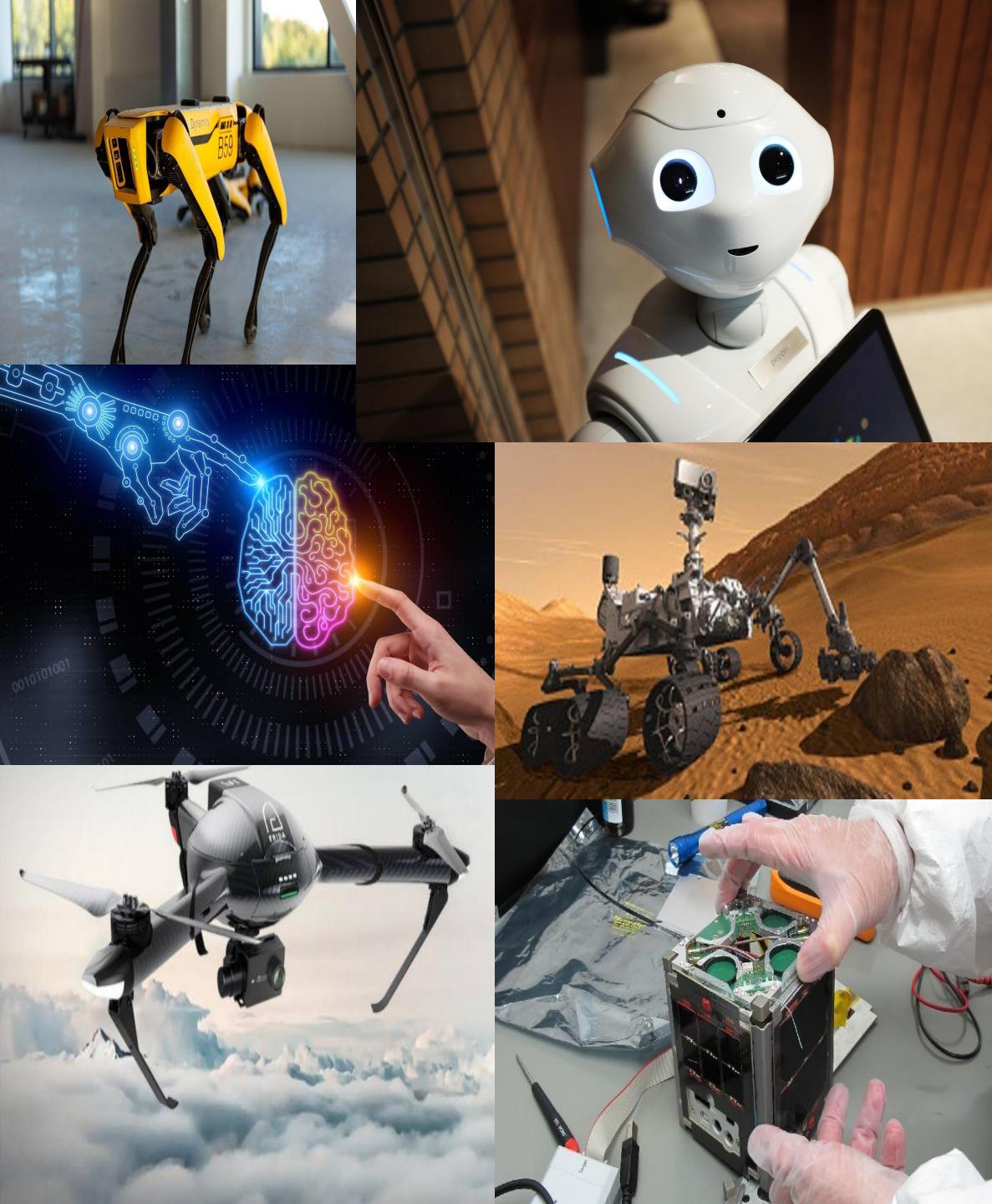


```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
from drones import Drone

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()
```

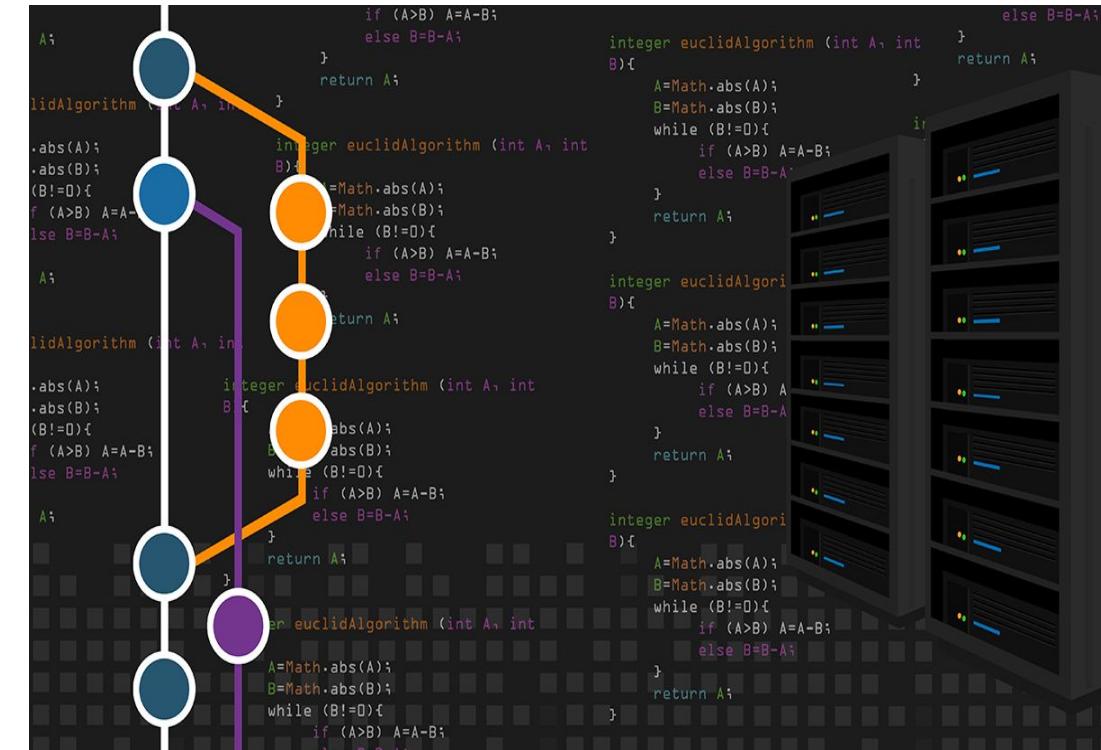
# Git / GitHub



# Git

## Sistema de control de versiones

- Branches
- Distribuido (historia completa, commits locales, escalabilidad)
- Pull requests (Pedir a otro desarrollador que haga un “merge” de un branch) -- trazabilidad
- Comunidad
- Faster release cycle



60

Next  
↓



# GitHub

<https://github.com/leonardocfor>

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

# Git

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner \*      Repository name \*



/

intro\_to\_smart\_uavs



Great repository names are short and memorable. Need inspiration? How about [fluffy-waffle](#)?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

[Create repository](#)

## Quick setup — if you've done this kind of thing before

or    [HTTPS](#)    [SSH](#)    [https://github.com/leonardocfor/intro\\_to\\_smart\\_UAVs.git](https://github.com/leonardocfor/intro_to_smart_UAVs.git)

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

## ...or create a new repository on the command line

```
echo "# intro_to_smart_UAVs" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin https://github.com/leonardocfor/intro_to_smart_UAVs.git
git push -u origin master
```

## ...or push an existing repository from the command line

```
git remote add origin https://github.com/leonardocfor/intro_to_smart_UAVs.git
git branch -M master
git push -u origin master
```

## ...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

# Git

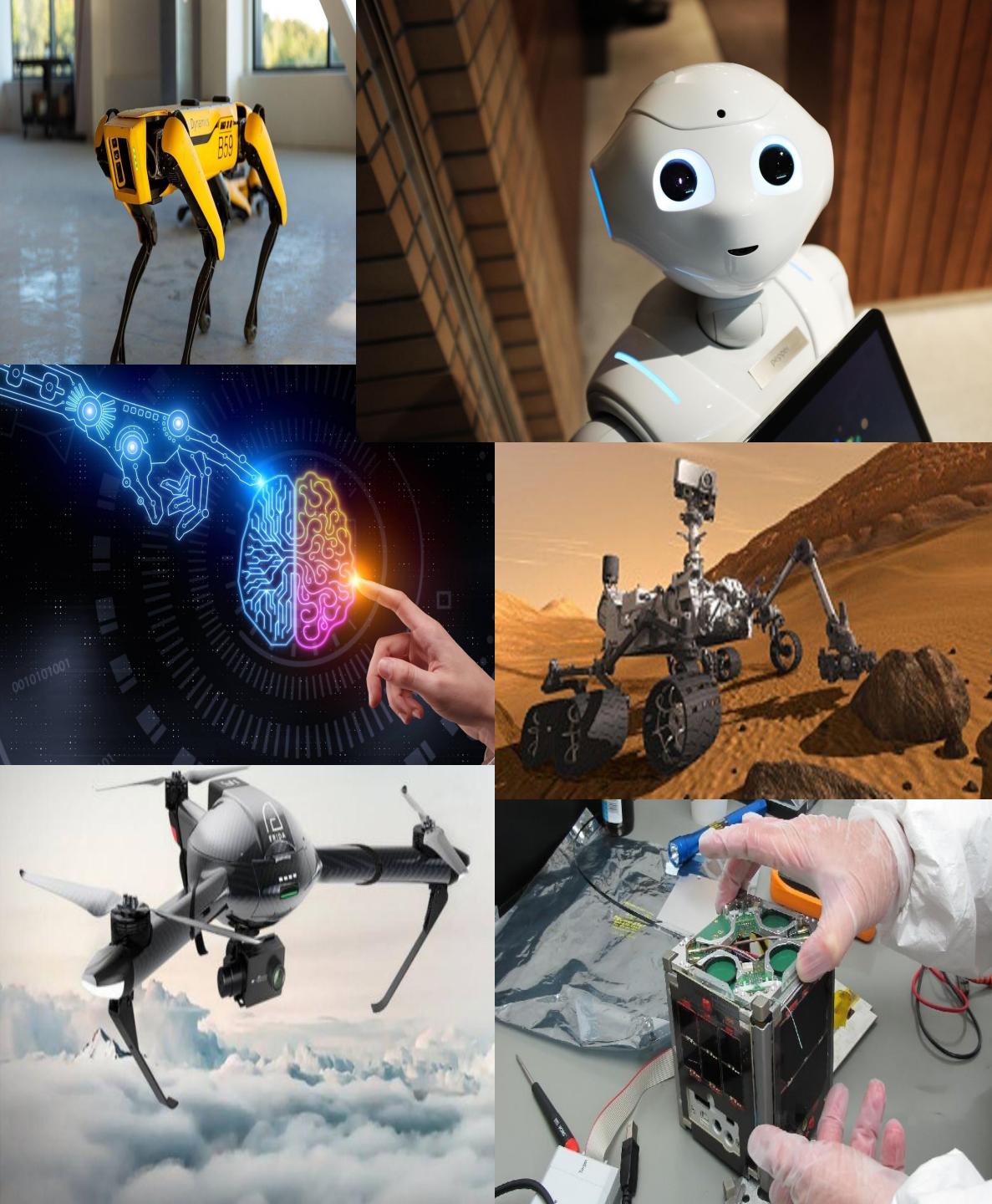
## Practica

```
$ sudo apt-get install git // # apt-get install git  
$ cd  
$ cd intro_to_smart_uavs  
$ git init  
$ git add __init__.py session1/__init__.py session2/__init__.py session3/__init__.py session4/__init__.py  
$ git commit -m "Adds course folder structure"  
$ git branch -M master  
$ git remote add origin https://github.com/<your_user>intro_to_smart_UAVs.git  
$ git push -u origin master
```

62

## Ejercicio

Hacer un commit con los archivos drones.py mydrone.py

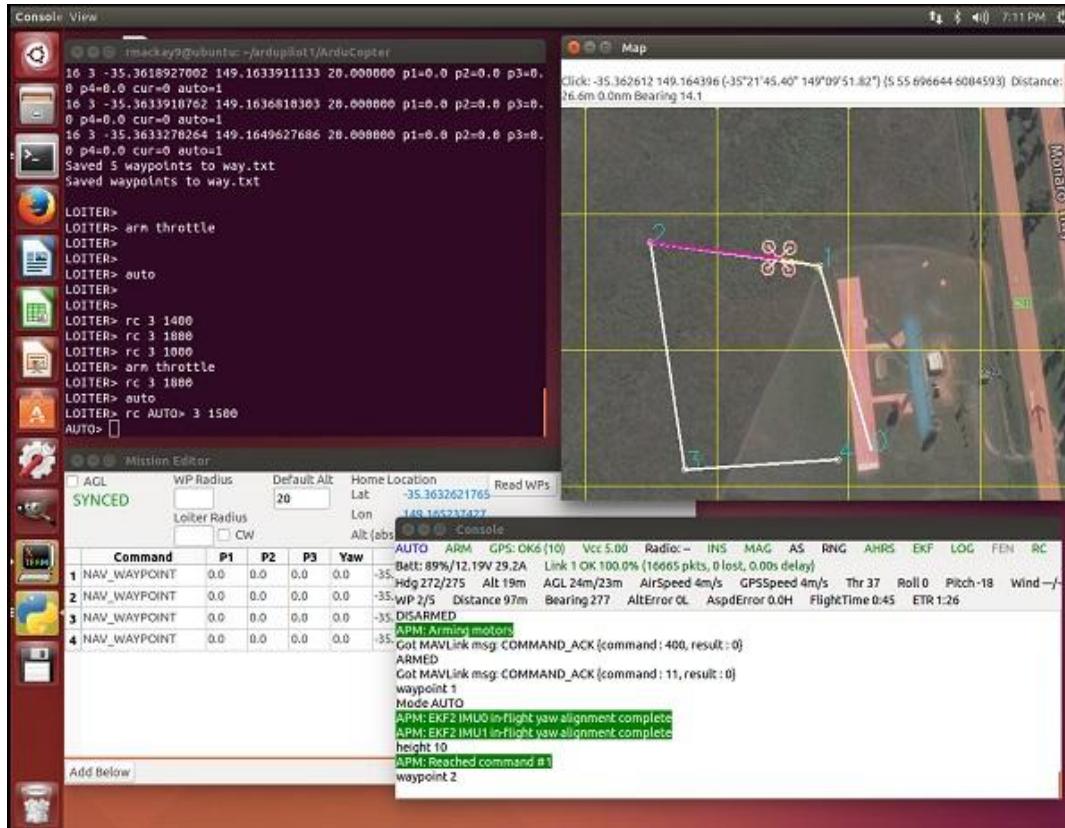


# Para la siguiente sección

# Para la siguiente sección

## Tareas

### 1. Instalación y configuración de ArduPilot SITL (Software In The Loop)



```
$ cd
$ git clone https://github.com/ArduPilot/ardupilot.git
$ cd ardupilot
$ git submodule update --init --recursive
$ cd
$ nano .bashrc (al final del archivo)
$ export PATH=$PATH:$HOME/ardupilot/Tools/autotest
$ source .bashrc
$ cd ardupilot/ArduCopter # AntennaTracker, ArduPlane, ArduSub, Rover
$ sim_vehicle.py -w
```

# Para la siguiente sección

## Tareas

### 1. Instalación y configuración de ArduPilot SITL (Software In The Loop)

```
[753/760] Compiling ArduCopter/mode_smart_rtl.cpp
[754/760] Compiling ArduCopter/Log.cpp
[755/760] Compiling ArduCopter/compassmot.cpp
[756/760] Compiling ArduCopter/tuning.cpp
[757/760] Compiling ArduCopter/RC_Channel.cpp
[758/760] Compiling ArduCopter/system.cpp
[759/760] Compiling ArduCopter/takeoff.cpp
[760/760] Linking build/sitl/bin/arducopter
Waf: Leaving directory '/home/leonardo/ardupilot-updated/build/sitl'

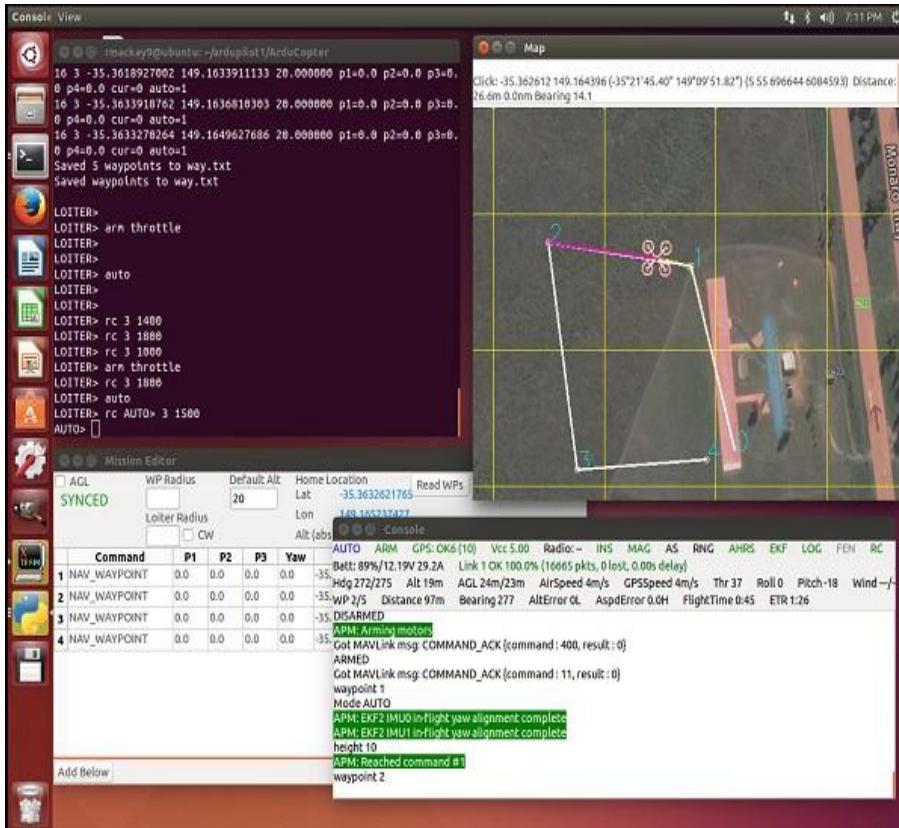
BUILD SUMMARY
Build directory: /home/leonardo/ardupilot-updated/build/sitl
Target      Text     Data    BSS   Total
-----
bin/arducopter  3106969  128283  92512  3327764

Build commands will be stored in build/sitl/compile_commands.json
'build' finished successfully (3m0.677s)
SIM_VEHICLE: Using defaults from (/home/leonardo/ardupilot-updated/Tools/autotest/default_params/copter.parm)
SIM_VEHICLE: Run ArduCopter
SIM_VEHICLE: "/home/leonardo/ardupilot-updated/Tools/autotest/run_in_terminal_window.sh" "ArduCopter" "/home/leonardo/ardupilot-updated/build/sitl/bin/arducopter" "-S" "-w" "--model" "+" "--speedup" "1" "--defaul
ts" "/home/leonardo/ardupilot-updated/Tools/autotest/default_params/copter.parm" "-I0"
SIM_VEHICLE: Run MavProxy
SIM_VEHICLE: "mavproxy.py" "--out" "127.0.0.1:14550" "--out" "127.0.0.1:14551" "--master" "tcp:127.0.0.1:5760" "--sitl" "127.0.0.1:5501"
RiTW: Starting ArduCopter : /home/leonardo/ardupilot-updated/build/sitl/bin/arducopter -S -w --model + --speedup 1 --defaults /home/leonardo/ardupilot-updated/Tools/autotest/default_params/copter.parm -I0
Connect tcp:127.0.0.1:5760 source_system=255
Log Directory:
Telemetry log: mav.tlog
Waiting for heartbeat from tcp:127.0.0.1:5760
MAV> APM: ArduCopter V4.1.0-dev (f8238486)
APM: e80850c49e3d43dc86dfd4bca120959
APM: Frame: QUAD
online system 1
STABILIZE> Mode STABILIZE
Received 1148 parameters
```

# Para la siguiente sección

## Tareas

## 1. Instalación y configuración de ArduPilot SITL (Software In The Loop)



```
$ ctrl+c  
$ sim_vehicle.py -v ArduCopter --map --console
```

# Google

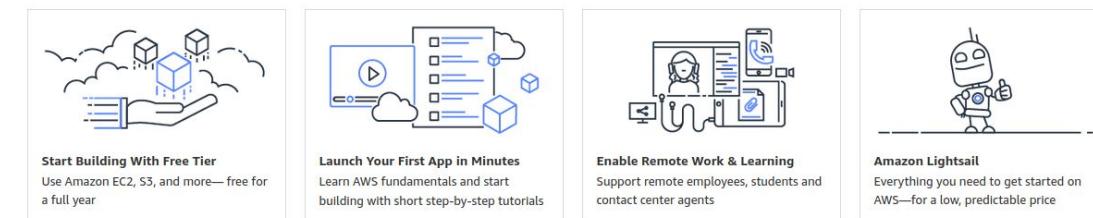
# Para la siguiente sección

## Tareas

### 2. Creación de cuenta en Amazon Web services



67



### Explore Our Products



<https://aws.amazon.com/>

# Gracias

Leonardo Camargo Forero, Ph.D

✉ [leonardo@ubihpc.com](mailto:leonardo@ubihpc.com)

🌐 [www.ubihpc.com](http://www.ubihpc.com)

