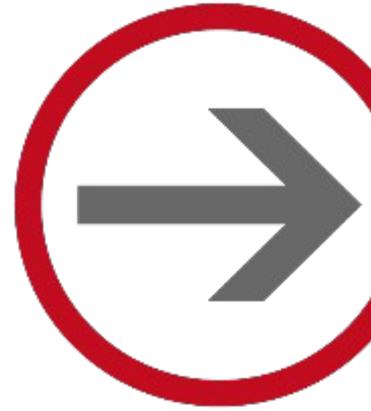


INTEGRACIÓN DE ROBOTS FÍSICOS EN UNA PLATAFORMA WEB DE ROBÓTICA EDUCATIVA



***Autor:* David Valladares Vigara**

***Tutores:* José María Cañas Plaza y Julio Manuel Vega Pérez**

La **Robótica Educativa** permite el diseño y desarrollo de robots que ayudan en la iniciación temprana en las ciencias y tecnologías. Es una enseñanza STEM.

Expansión de la robótica educativa

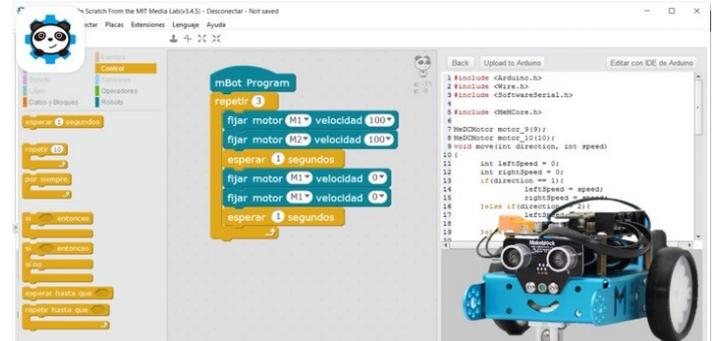
- Asignaturas específicas
- Lenguajes de programación: *Scratch*
- Kits de robótica: *mBot*, *Mindstorms EV3*, *VEX Robotics*
- Aplicaciones y entornos web: *MBlock*, *Bitbloq*, *Open Roberta Lab*, *Kibotics*
- Competiciones robóticas: *Robocampeones*



Scratch



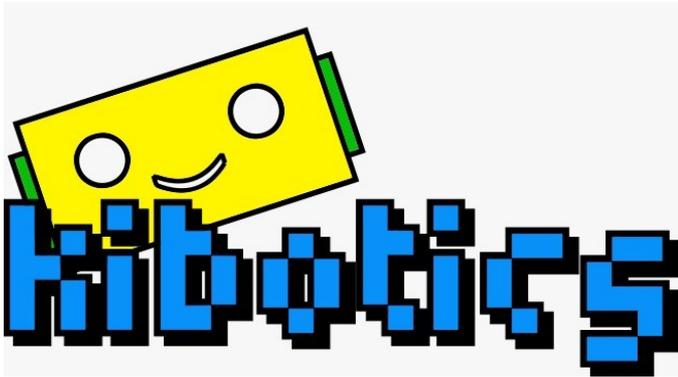
Mindstorms EV3



MBlock

INTRODUCCIÓN

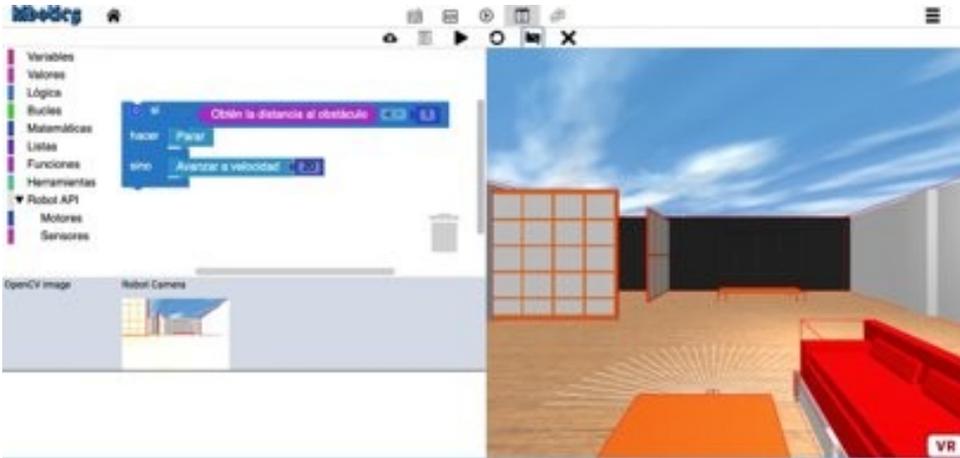
Kibotics es un entorno web enfocado a la enseñanza de la robótica y programación de forma sencilla. Fue desarrollado por la Asociación JdeRobot.



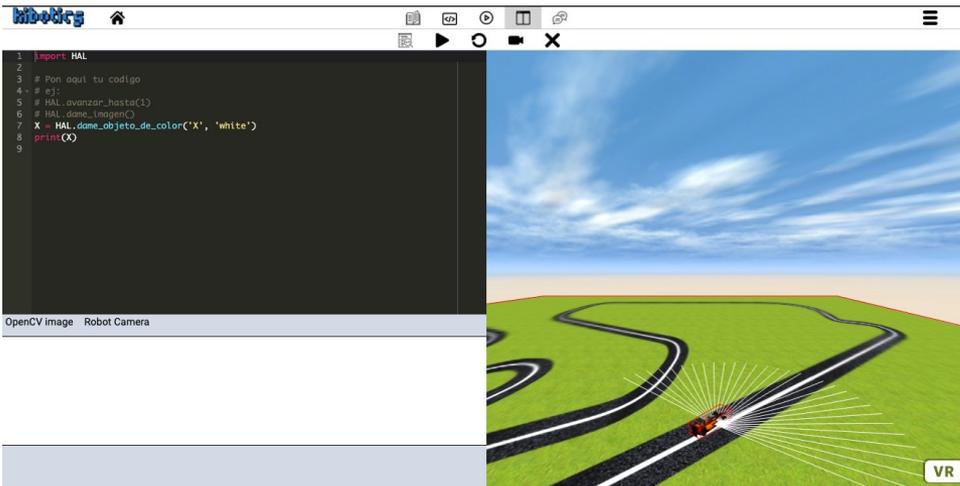
- Multirrobot
- Evaluación automática sobre los ejercicios
- Correcciones de estilo
- Ejercicios de visión artificial

INTRODUCCIÓN

Lenguajes de programación reconocidos en Kibotics



1. **Blockly**: orientado a alumnos más pequeños al ser un lenguaje visual.



2. **Python**: orientado a alumnos más mayores. Uno de los lenguajes más populares.

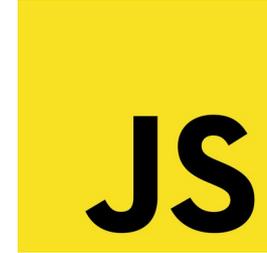
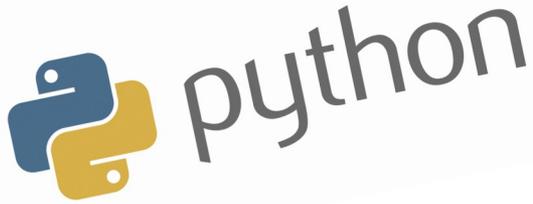
OBJETIVOS

Conseguir la integración de nuevos robots reales (mBot, dron Tello y GoPiGo3) en la plataforma Kibotics para poder programarlos desde ella.

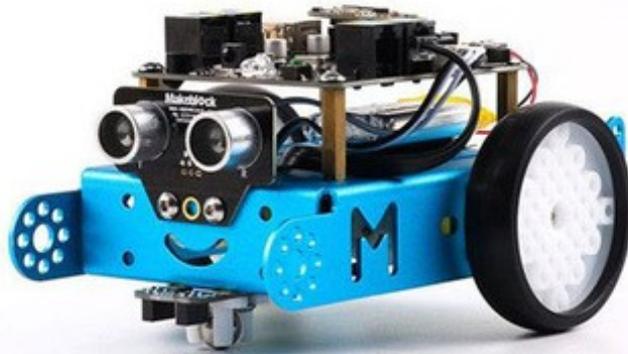
Objetivos específicos:

- Multiplataforma
- Envío sencillo y sin instalaciones previas del usuario

HERRAMIENTAS SOFTWARE

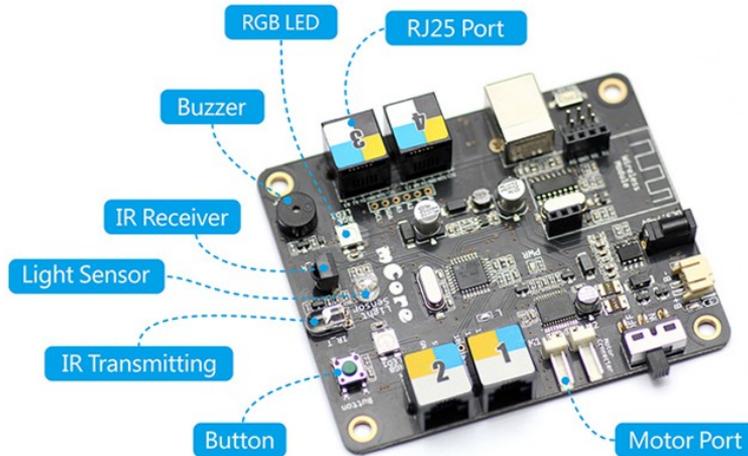


INTEGRACIÓN MBOT



1. Características del mBot

mBot (MakeBlock) es un robot orientado al aprendizaje de la robótica

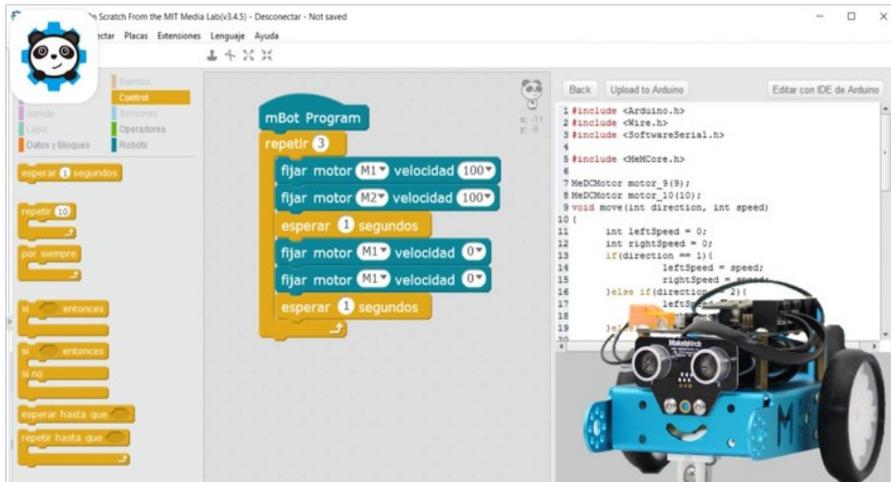


- **Hardware**

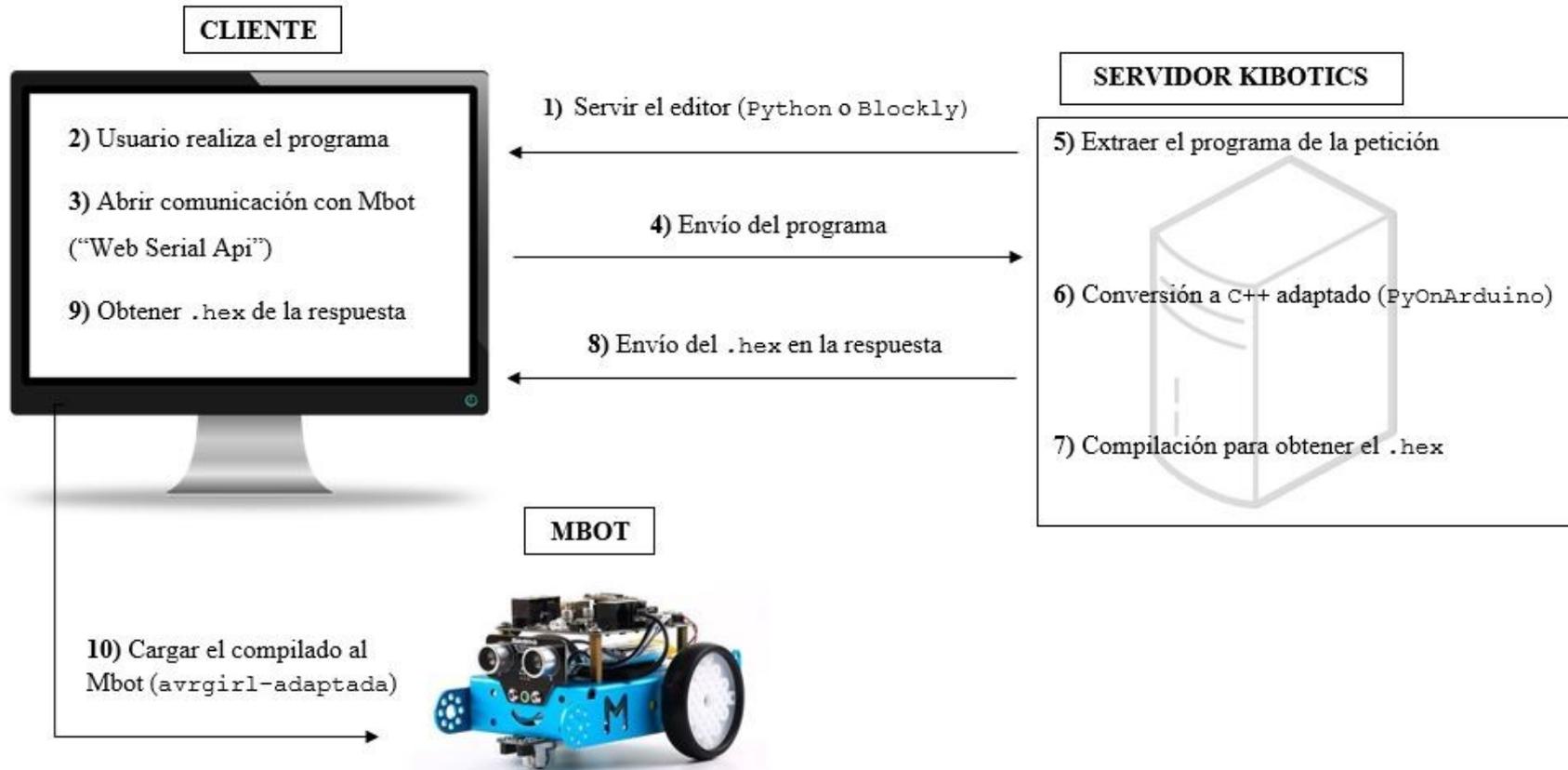
- Placa “mCore”, basada en Arduino Uno
- Sensor de línea, ultrasonido, motores..

- **Software**

- mBlock (MakeBlock)
- Scratch y Arduino
- Local o Web (requiere driver)

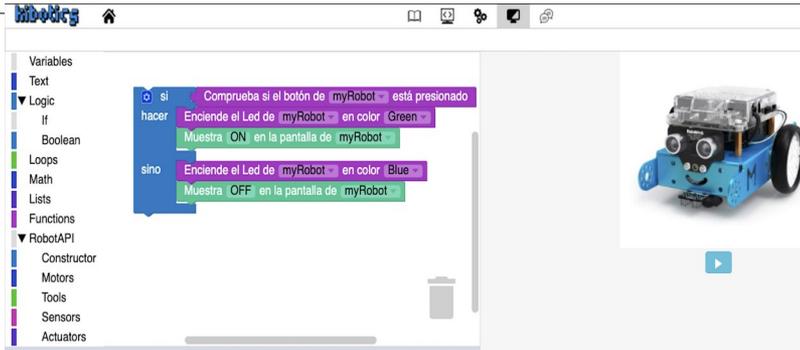


2. Diseño



3. Lado Cliente

1) El usuario realiza el programa en el editor Web

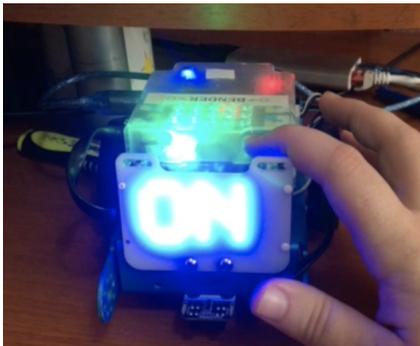


2) Al pulsar el botón, se inicia el proceso



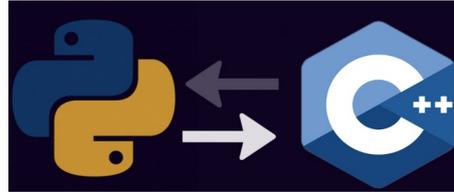
3) Se envía el código fuente al servidor para obtener el compilado

4) El compilado recibido se carga al mBot (Web Serial API y Avrgirl)



```
:100000000C942A000C9434000C9434000C943400AA
:100010000C9434000C9434000C9434000C94340090
:100020000C9434000C9434000C9434000C94340080
:100030000C9434000C9434000C9434000C94340070
:100040000C9434000C9434000C9434000C94340060
:100050000C94340011241FBECFE5D8E0DEBFCDBF25
:100060000E9436000C9445000C9400008FEF87BB73
:100070002CE231E088B3809588BB80E197E2F901FA
:0E0080003197F1F70197D9F7F5CFF894FFCF3C
:00000001FF
```

4. Lado Servidor



1) Extraemos el código fuente de la petición enviada por el navegador



2) Conversión a C++/Arduino (PyOnArduino)



3) Generación del compilado y obtención del .hex (AVRDUDE)



Código fuente

```
#include <iostream>
using namespace std;

int main() {
    cout<<"Hola mundo";
    return 0;
}
```



Compilador



Código máquina

```
100101101010
001001011110
101001000101
010100101010
101010010101
010101010101
000000110010
```

5. Validación

Experimental

- Validación => Ubuntu 18.04, Ubuntu 16.04, Windows 10, Windows 8, MacOS Mojave Versión 10.14.6

VENTAJAS

Multiplataforma (Linux, Windows, MacOs)

"Cero-instalación"

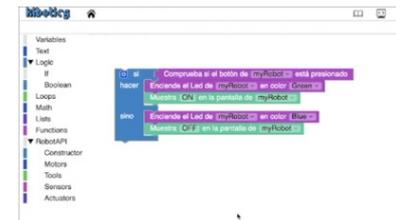
Envío directo

PROBLEMAS

Disponible solo en navegadores Chromium

Fase de experimentación, en necesario activarlo

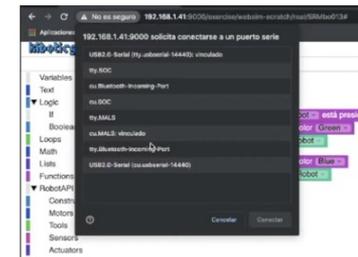
Ejemplo de envío



a)



b)



c)



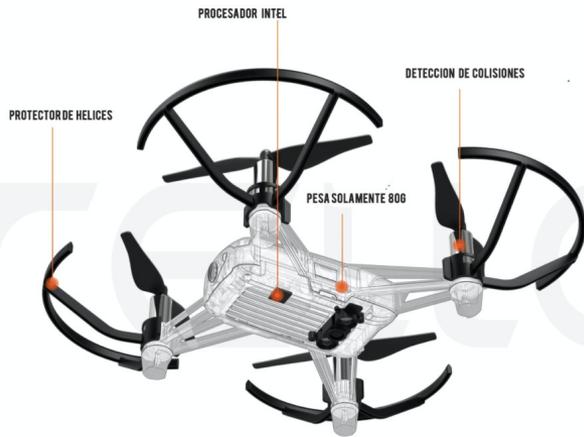
d)

INTEGRACIÓN DRON TELLO



1. Características del Tello

Tello (Ryze Technology) es un dron destinado para la iniciación



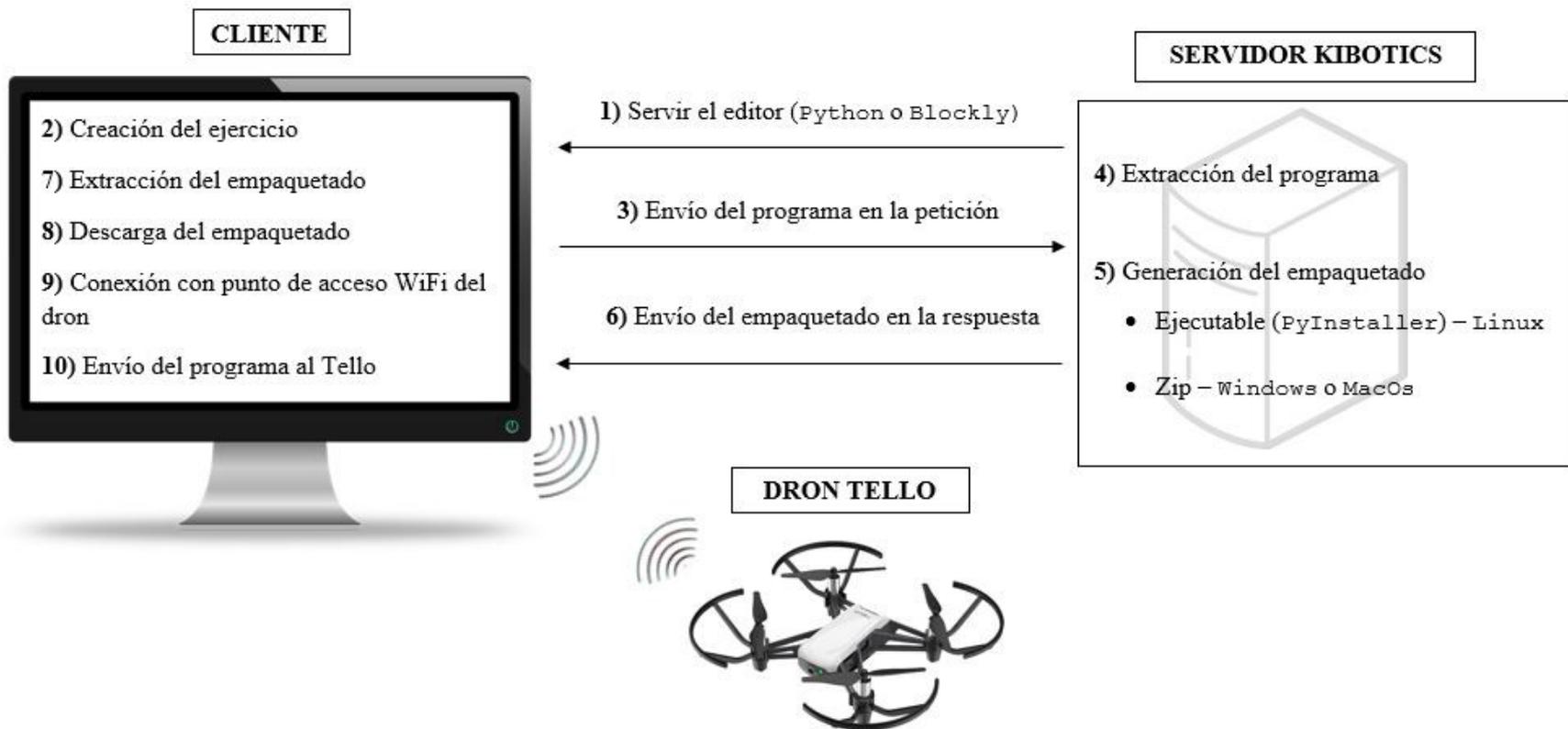
- **Hardware**

- Procesadora Intel cámara, radio de 200m, 28Km/h máximo
- Posicionamiento por VPS

- **Software**

- Tello EDU App
- IDE Scratch
- DroneBlocks

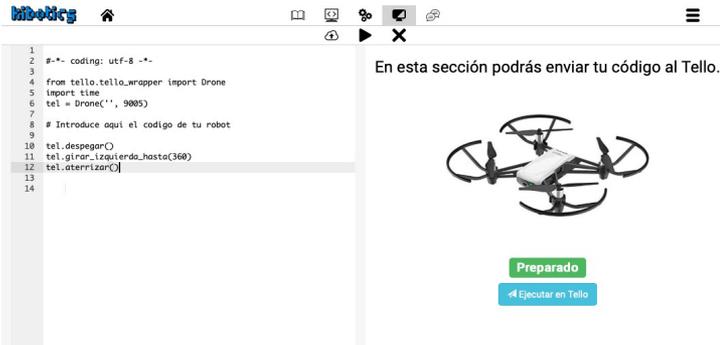
2. Diseño



- Ordenador anfitrión MacOS/Windows requiere tener instalado el intérprete de Python2.7 y drivers del Tello

3. Lado Cliente

1) El usuario realiza el programa en el editor Web



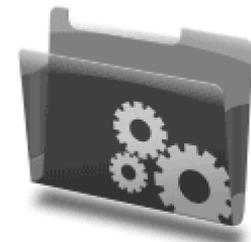
2) Al pulsar el botón, se inicia el proceso



3) Se envía el código fuente a el servidor para que prepare el ejecutable

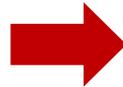
4)

- En Linux se recibe directamente el ejecutable
- En MacOS y Windows comprimido
- Conectar a la WiFi del dron y lanzar ejecutable



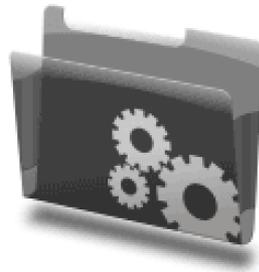
4. Lado Servidor

1) El servidor recibe el código fuente



2) En Linux: PyIntaller crea el ejecutable (contiene el intérprete de Python)

En MacOs/Windows: empaquetación y comprimido con las dep nec



3) Se envía el comprimido / ejecutable al navegador en la respuesta a la petición

5. Validación

Experimental

- **Validación S.O.:** *Ubuntu 18.04, Ubuntu 16.04, Windows 10, Windows 8, MacOS Mojave Versión 10.14.6*
- **Validación Navegadores:** *Safari Versión 14.0, Google Chrome Versión 85.0, Firefox Versión 68.9*

VENTAJAS

Multiplataforma (Linux, Windows, MacOs)

"Cero-instalación" en Linux

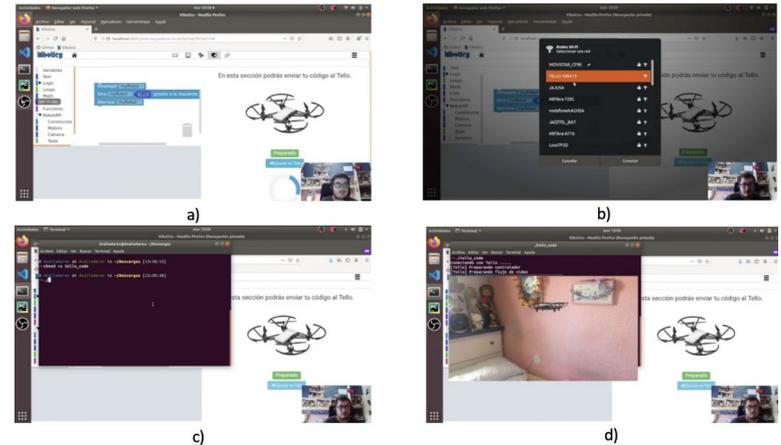
Compatible con los principales navegadores

PROBLEMAS

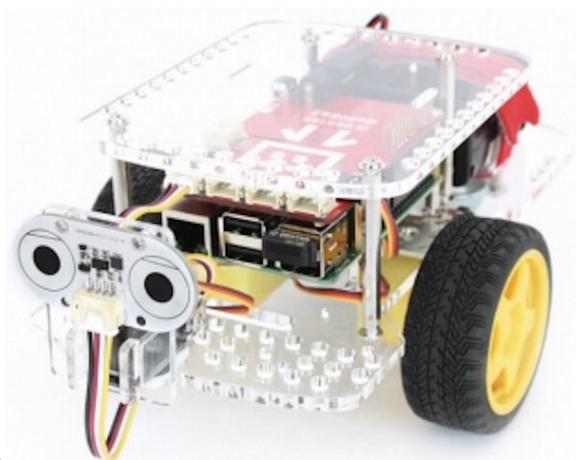
Drivers del Tello e intérprete de Python en Windows y MacOS

Envío menos directo (requiere la descarga de un ejecutable)

Ejemplo de envío

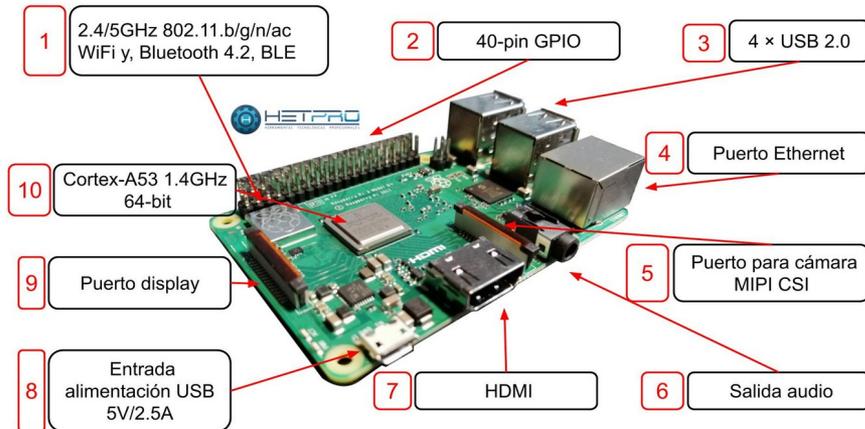


INTEGRACIÓN GOPIGO3



1. Características del GoPiGo3

GoPiGo3 (Dexter Industries) es un robot que integra una Raspberry Pi

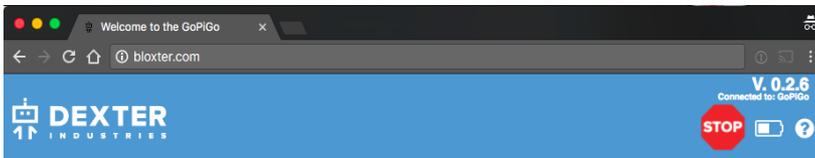


• Hardware

- Estructura acrílica gruesa
- Integra Raspberry Pi 3 modelo B y placa “GopiGo3”
- Forma de vehículo

• Software

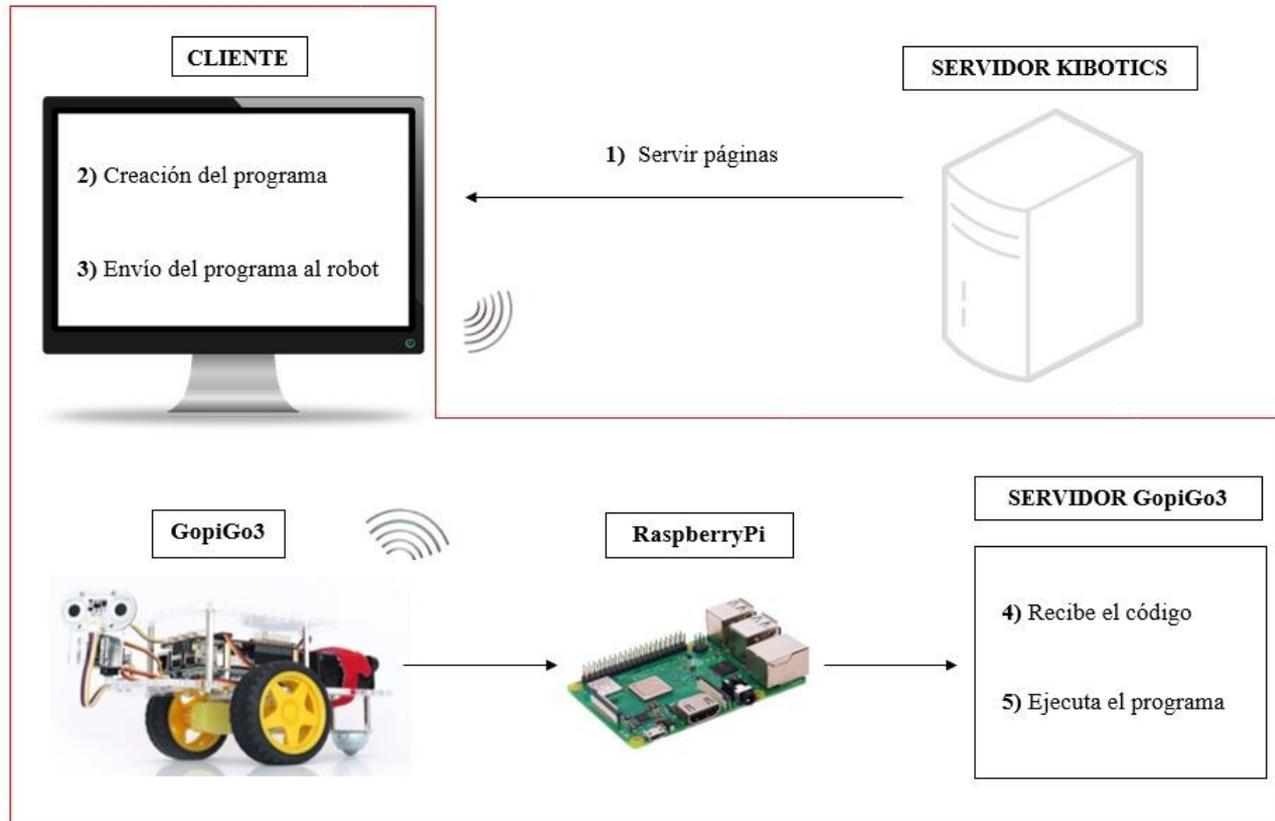
- DexterOS (en la Raspberry Pi)
 - Emite una señal WiFi a la que se conecta el usuario
 - En el navegador se abre una web para interactuar con el robot



You are connected to the GoPiGo3 and ready to ...



2. Diseño



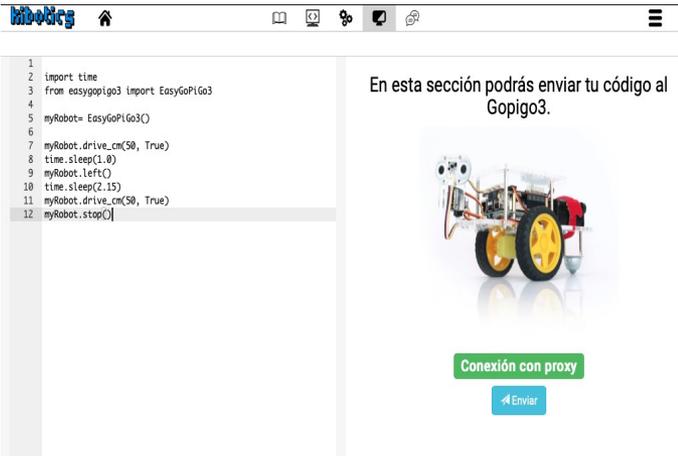
- Preparación de la Raspberry PI con Raspbian, drivers del GoPiGo3 y servidor Flask

3. Lado Cliente

1) El usuario realiza el programa en el editor Web y se pulsa el botón de envío



2) Se envía el código escrito por el usuario al servidor Kibotics para su almacenamiento



```

1
2 import time
3 from easygopigo3 import EasyGoPiGo3
4
5 myRobot= EasyGoPiGo3()
6
7 myRobot.drive_cm(50, True)
8 time.sleep(1.0)
9 myRobot.left()
10 time.sleep(2.15)
11 myRobot.drive_cm(50, True)
12 myRobot.stop()
    
```

En esta sección podrás enviar tu código al Gopigo3.



Conexión con proxy

Enviar



3) Se envía el código escrito por el usuario al servidor Flask del Robot

3. Lado Servidor Kibotics

- No toma un papel tan importante para el proceso de envío como para el mBot y Tello
- Encargado de servir las páginas al navegador
- Recibe el código escrito por el usuario desde el navegador para su almacenamiento

3. Lado Servidor GoPiGo3

1) Extrae el código del usuario



2) Crear el ejecutable en Python



3) Se lanza un proceso con el programa que ejecuta el robot



5. Validación

Experimental

- **Validación S.O.:** *Ubuntu 18.04, Ubuntu 16.04, Windows 10, Windows 7, MacOS Catalina Versión 10.15.6, MacOS Mojave Versión 10.14.6*
- **Validación Navegadores:** *Safari Versión 14.0, Google Chrome Versión 85.0, Firefox Versión 68.9*

VENTAJAS

Multiplataforma (Linux, Windows, MacOs)

"Cero-instalación" en el ordenador del usuario

Compatible con los principales navegadores

PROBLEMAS

Preparación de la Raspberry Pi, aunque típicamente se proporcionará SD de Kibotics

Ejemplo de envío

```

1
2 import time
3 from easygopigo3 import EasyGoPiGo3
4
5 myRobot = EasyGoPiGo3()
6
7 myRobot.drive_cm(30, True)
8 time.sleep(1.0)
9 myRobot.left()
10 time.sleep(2.5)
11 myRobot.drive_cm(30, True)
12 myRobot.stop()
    
```

a)

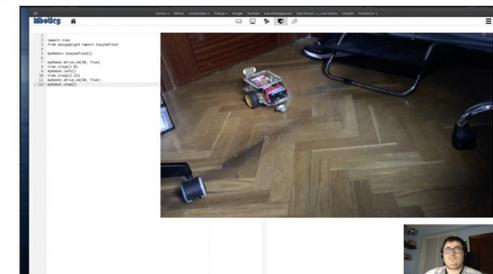
En esta sección podrás enviar tu código al Gopigo3.



Conexión con proxy

Enviar

b)



c)

CONCLUSIONES

- Soporte a los tres robots desde Kibotics
- Integración multiplataforma (Linux, Windows, MacOS)
- Simplificación del proceso de envío



mBot	Cero-instalación	Envío directo	Web Serial API
Tello	Instalación en Windows/MacOS	Ejecutable	PyInstaller / Empaquetado
GoPiGo3	Cero-instalación	Envío directo	Servidor Flask a bordo

1. Web Serial API para la comunicación con otros robots
2. Programación del GoPiGo3 con Blockly y diseño de nuevos ejercicios (visión artificial)
3. Integración de nuevos robots (LEGO EV3) y basados en Arduino aprovechando la infraestructura diseñada

**¡MUCHAS GRACIAS
POR SU ATENCIÓN!**