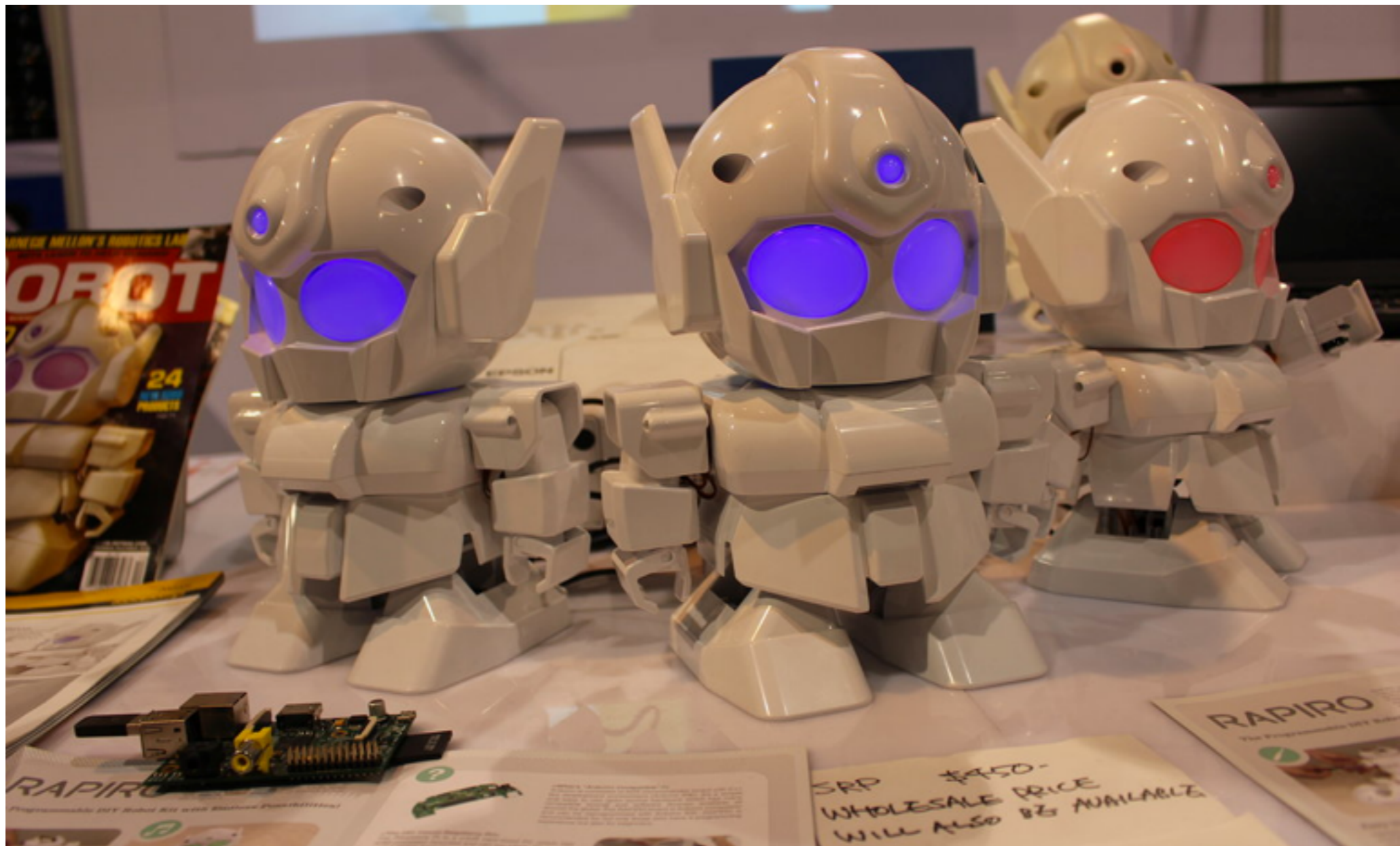


SG 
VIRTUAL
CONFERENCE
6ta edición

Re-evolución Robótica

Presentado por:
Javier Cervantes Ponce
[@solojavier](#)



"The future is already here, it's just not evenly distributed"

William Gibson

La re-evolución robótica

llegó para quedarse



La re-evolución robótica

Ilegó para quedarse

¿Estamos preparados?



La re-evolución robótica

Ilegó para quedarse

¿Estamos preparados?

RobotOps al rescate



RobotOps

- Mientras construimos el “**Internet de las cosas**” con la integración de millones de dispositivos conectados, necesitamos encontrar la manera de hacer que estos nuevos sistemas “críticos” **funcionen correctamente.**

RobotOps

- Mientras construimos el “**Internet de las cosas**” con la integración de millones de dispositivos conectados, necesitamos encontrar la manera de hacer que estos nuevos sistemas “críticos” **funcionen correctamente**.
- “RobotOps” busca extrapolar las **lecciones aprendidas en el pasado** con algunos nuevos ajustes.

RobotOps

- "RobotOps" busca la **aplicación de diferentes prácticas** para mantener la **operación de la internet de las cosas**, en el mismo sentido que prácticas similares mantienen operando a internet.

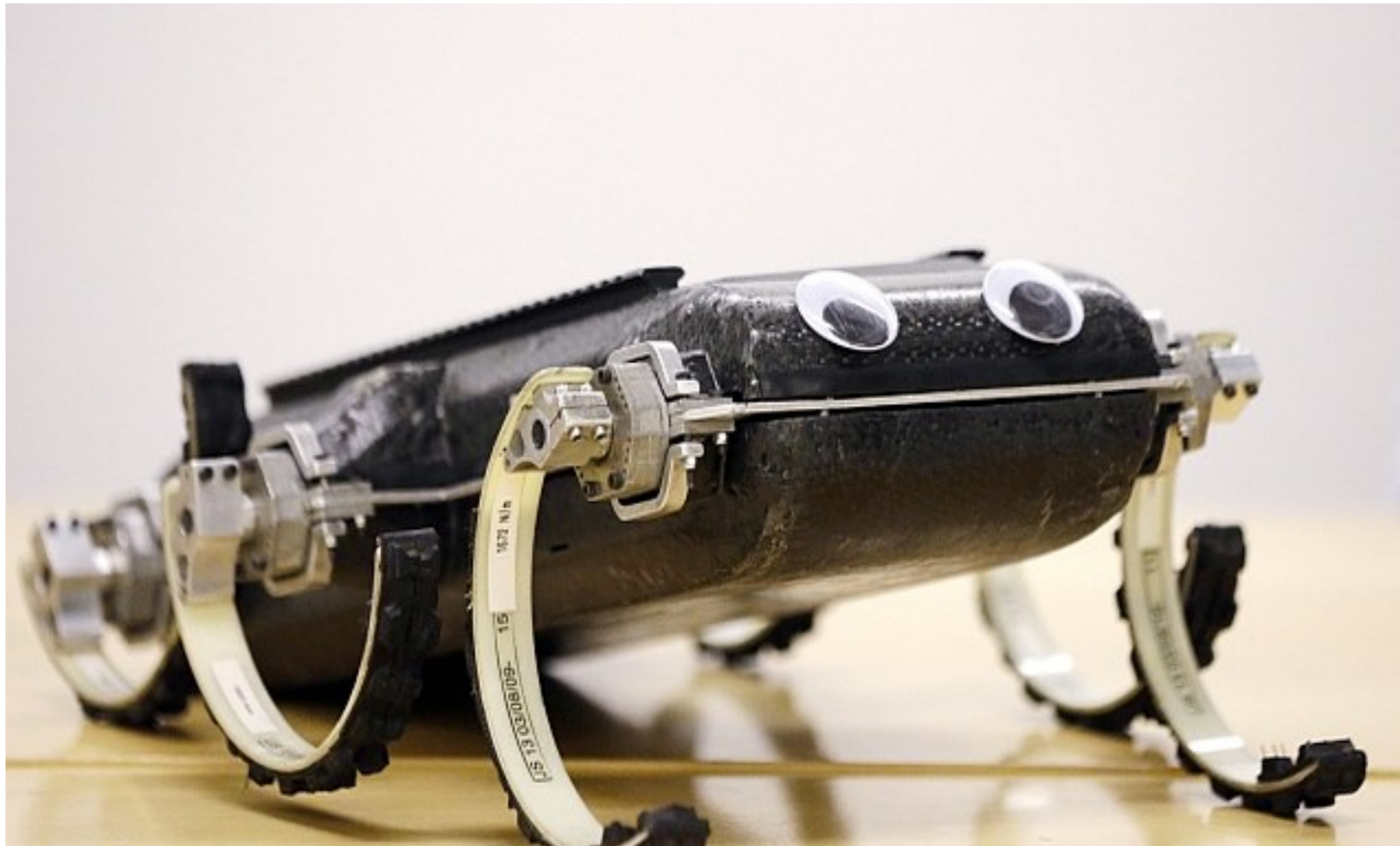
RobotOps

- "RobotOps" busca la **aplicación de diferentes prácticas** para mantener la **operación de la internet de las cosas**, en el mismo sentido que prácticas similares mantienen operando a internet.
- La definición de estas prácticas será **absolutamente esencial para empresas, gobiernos y organizaciones** que cuenten con redes de sensores distribuidos, robots/drones como parte de su infraestructura.



10 reglas de RobotOps

Para que cualquier esfuerzo de desarrollo pueda enfrentarse a los retos que se están presentando



Test Driven Robotics

Test Driven Robotics

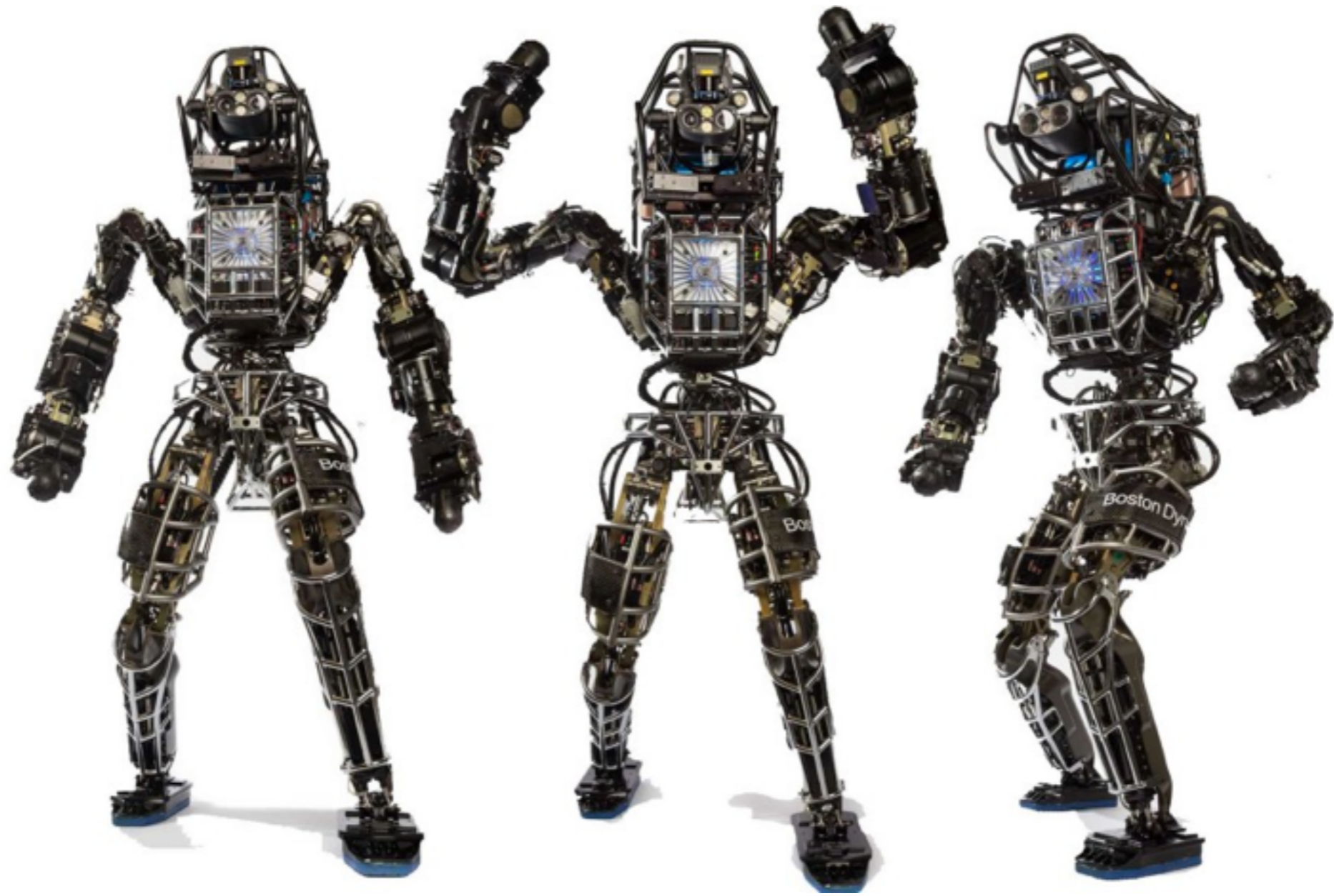
- La aplicación de TDD ha sido uno de los ingredientes esenciales para que los mejores equipos puedan construir software confiable, sin sacrificar la velocidad.



Test Driven Robotics

- La aplicación de TDD ha sido uno de los ingredientes esenciales para que los mejores equipos puedan construir software confiable, sin sacrificar la velocidad.
- Test Driven Robotics utiliza el mismo enfoque pero para el desarrollo de software para sistemas embebidos y hardware.





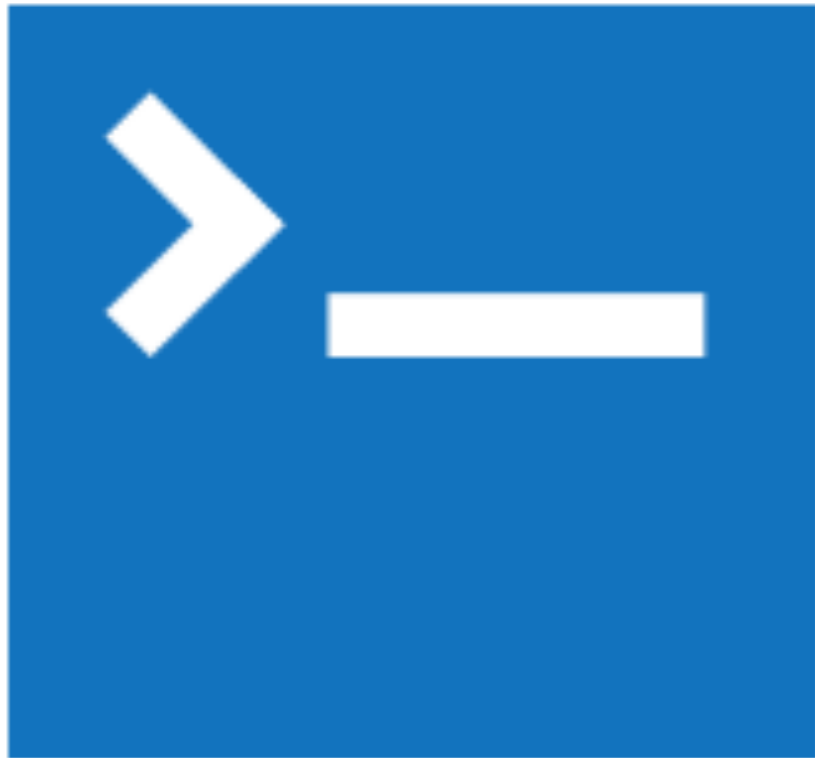
Linea de Comandos

Linea de comandos



- Las actualizaciones de software/firmware, el reinicio de dispositivos y otras actividades deben poder realizarse utilizando la línea de comandos.

Linea de comandos



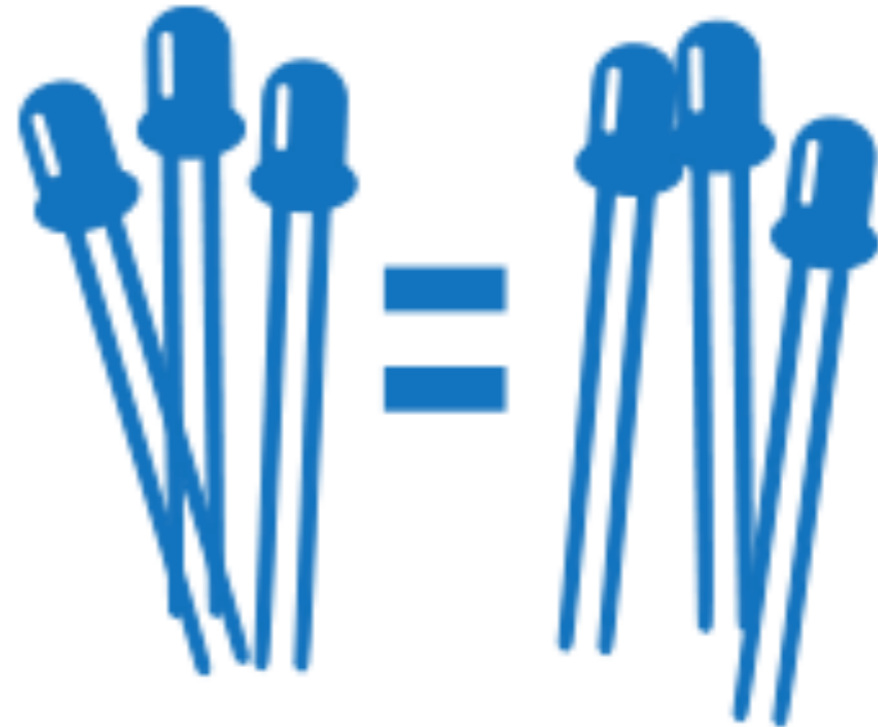
- Las actualizaciones de software/firmware, el reinicio de dispositivos y otras actividades deben poder realizarse utilizando la línea de comandos.
- No hay manera operar a gran escala sin la posibilidad de automatizar estos y otros procesos.



Múltiples plataformas/dispositivos

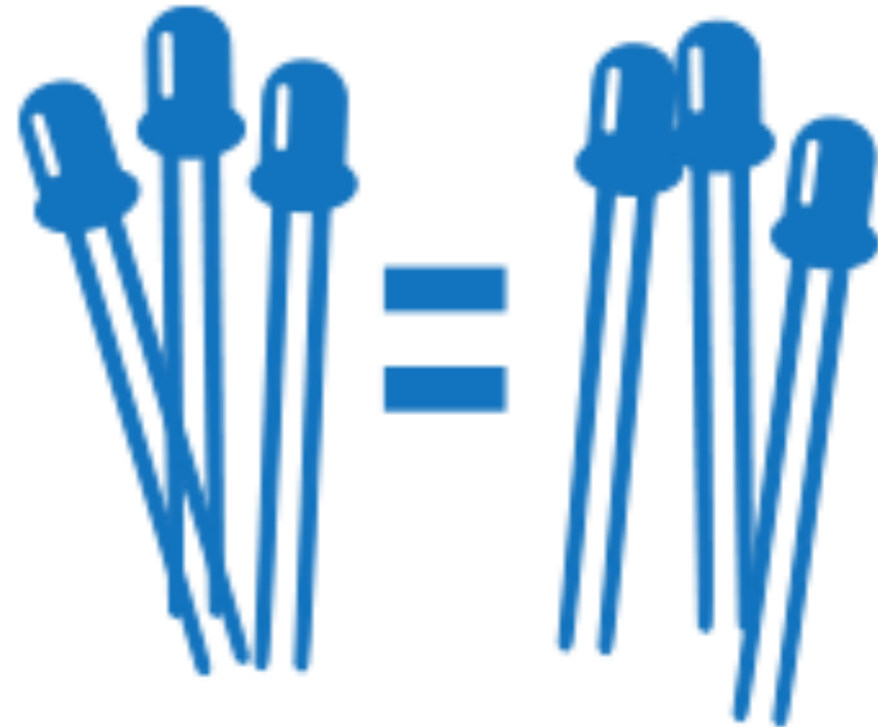
Múltiples plataformas/dispositivos

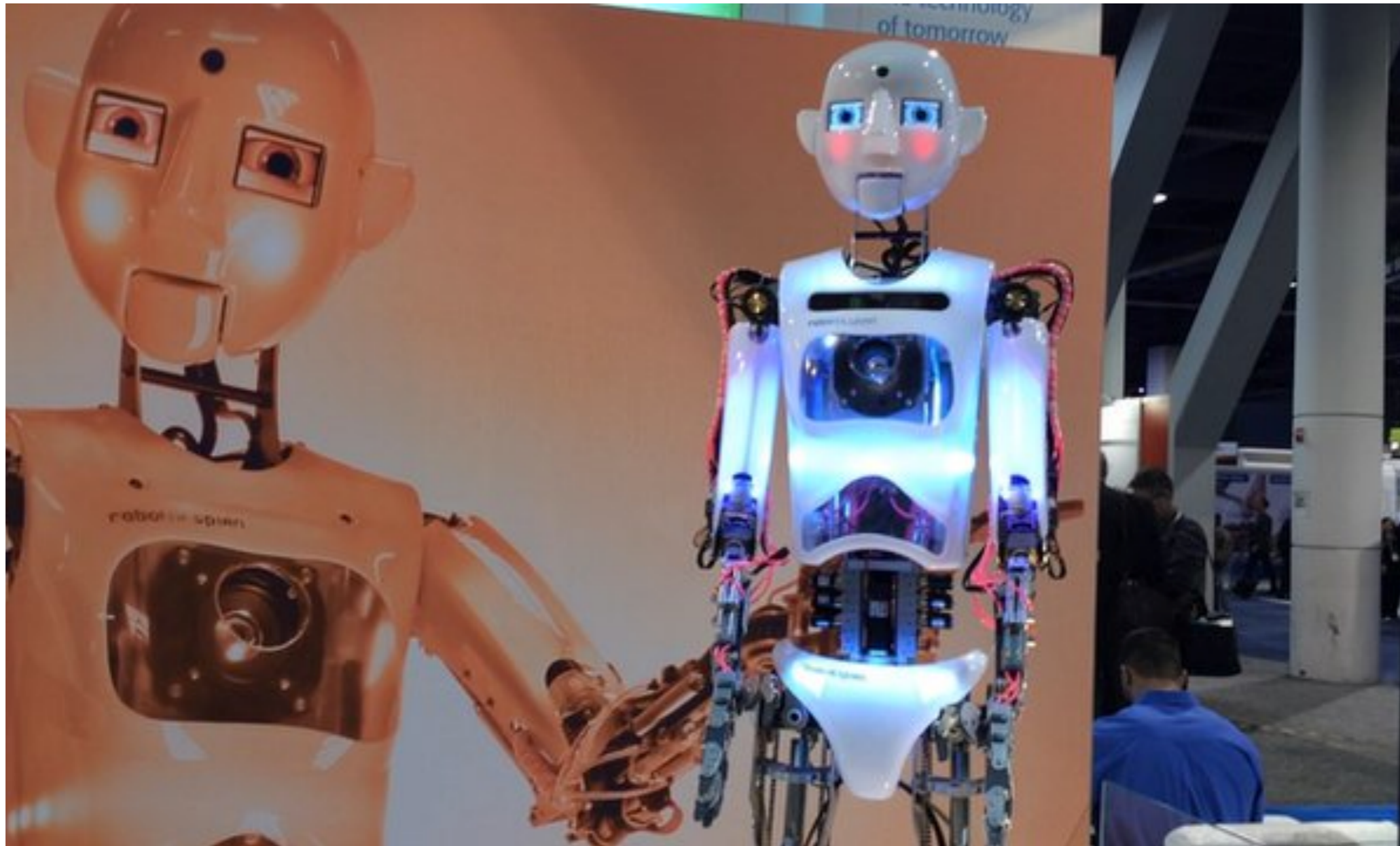
- Es importante contar con estándares de codificación en todo los productos, ya que es un elemento clave para la calidad del desarrollo y el mantenimiento del software.



Múltiples plataformas/dispositivos

- Es importante contar con estándares de codificación en todo los productos, ya que es un elemento clave para la calidad del desarrollo y el mantenimiento del software.
- Escribir código diferente para incorporar plataformas similares es ineficiente y dificulta la entrada de nuevos desarrolladores a un proyecto





Generadores de código

Generadores de código

- No todo el código puede ser auto-generado.
- Sin embargo se puede contar con scripts que generen un esqueleto para comenzar nuevos proyectos, agregar soporte adicional, o hacer tareas comunes.



Generadores de código

- No todo el código puede ser auto-generado.
- Sin embargo se puede contar con scripts que generen un esqueleto para comenzar nuevos proyectos, agregar soporte adicional, o hacer tareas comunes.
- Esta herramienta es clave para mantener la consistencia del código e incrementar la velocidad de desarrollo.

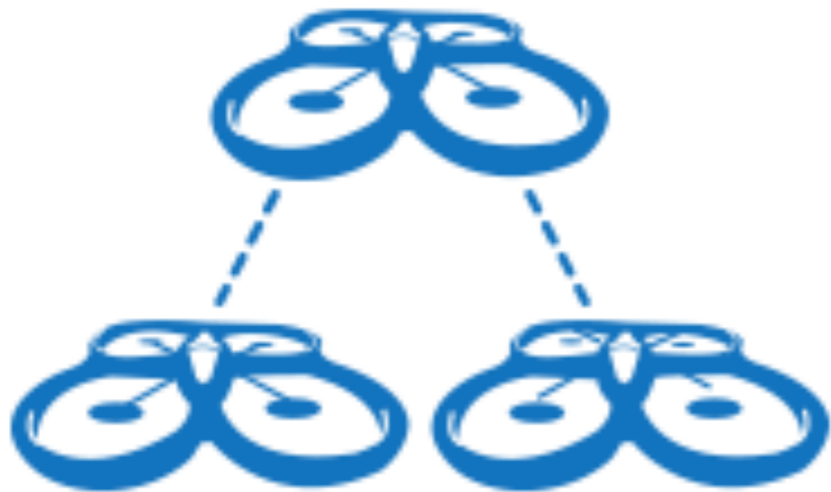




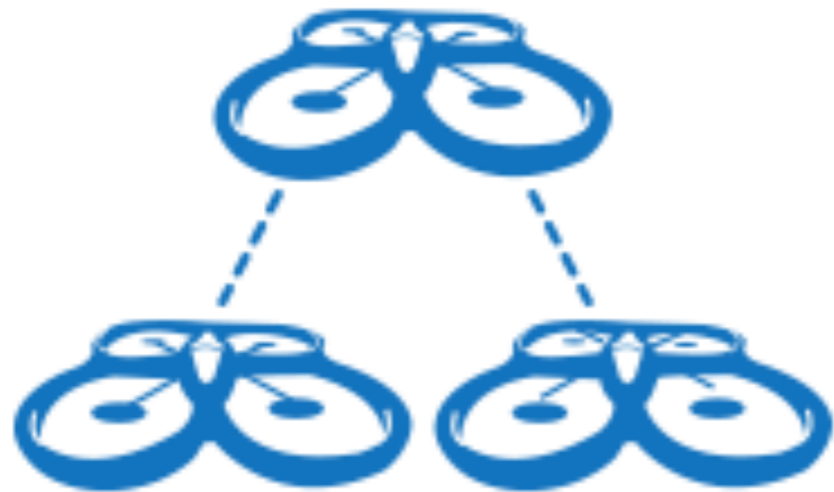
Grupos (swarms)

Grupos (swarms)

- Para operar grupos requiere una manera de coordinar muchos dispositivos al mismo tiempo.



Grupos (swarms)



- Para operar grupos requiere una manera de coordinar muchos dispositivos al mismo tiempo.
- Estos dispositivos deberán ser capaces de coordinar sus propias actividades, así como recibir comandos del centro de control.



API

API

- La integración con otros sistemas es crucial. Es necesario poder ejecutar comandos, así como recibir información.



API

- La integración con otros sistemas es crucial. Es necesario poder ejecutar comandos, así como recibir información.
- Se pueden utilizar protocolos basados en web, así como otros protocolos más *simples*.





Herramientas administrativas

Herramientas administrativas

- Así como un módem contiene una página para su administración, se requiere algún tipo de panel de control para los nuevos dispositivos de hardware.



Herramientas administrativas

- Así como un módem contiene una página para su administración, se requiere algún tipo de panel de control para los nuevos dispositivos de hardware.
- Esto es esencial para la administrar la configuración, así como poder identificar o resolver problemas.





Sintaxis declarativa

Sintaxis declarativa



- Escribir código que pueda describir lo que el sistema debe hacer, es el corazón de cualquier framework.



Sintaxis declarativa

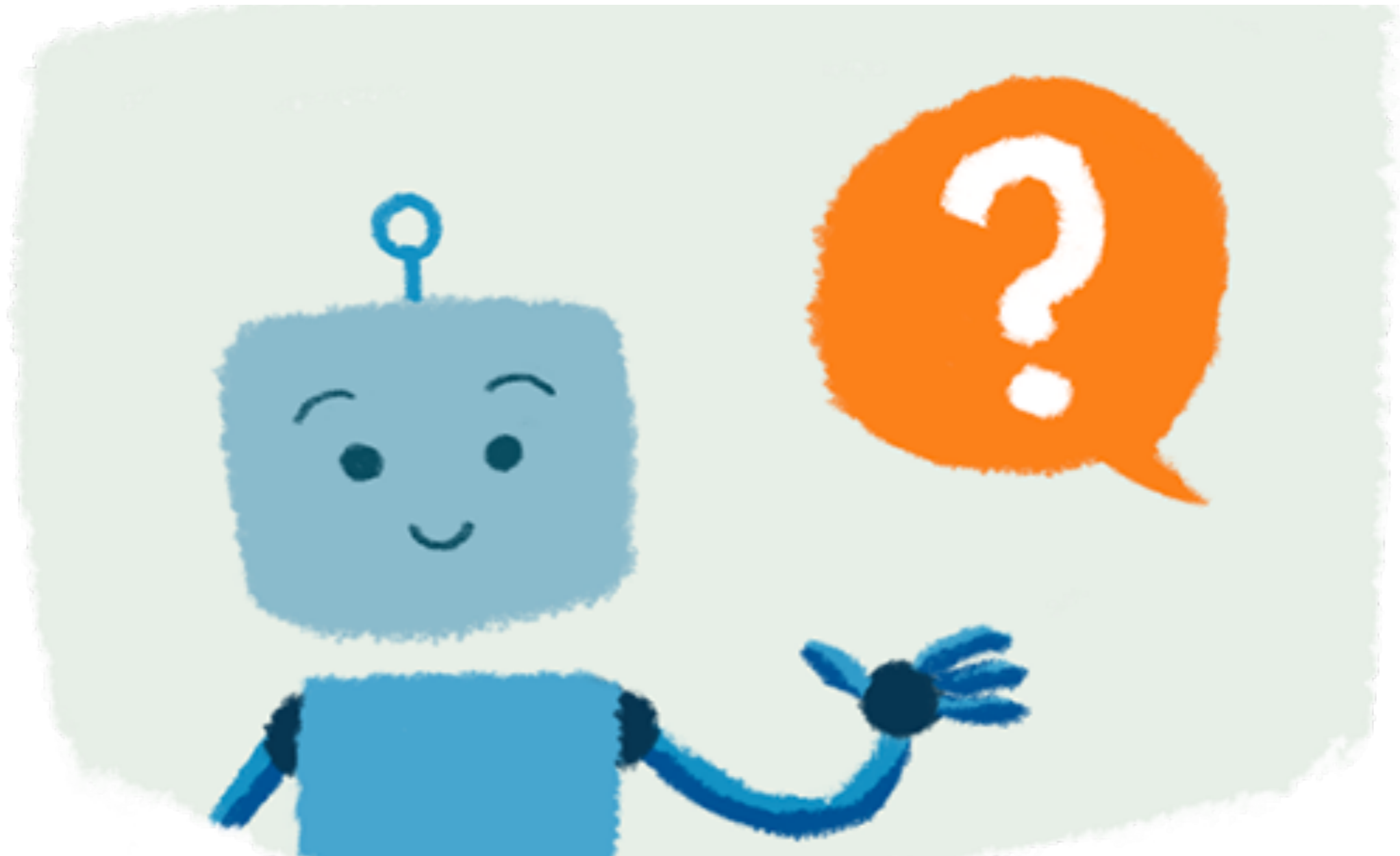


- Escribir código que pueda describir lo que el sistema debe hacer, es el corazón de cualquier framework.



- Investigaciones sobre inteligencia artificial y la interacción humano-maquina, han mostrado que es más fácil para los programadores el uso de la sintaxis declarativa. (Ej: DSL)





Documentación y soporte

Documentación y soporte

- La documentación es esencial para ayudar a los usuarios a realizar lo que requieren.



Documentación y soporte

- La documentación es esencial para ayudar a los usuarios a realizar lo que requieren.
- Se deben incluir descripciones sobre uso de las interfaces, así como múltiples ejemplos.



Documentación y soporte

- La documentación es esencial para ayudar a los usuarios a realizar lo que requieren.
- Se deben incluir descripciones sobre uso de las interfaces, así como múltiples ejemplos.
- Adicionalmente, es necesario algún tipo de soporte, aunque sea brindado por voluntarios, como ocurre generalmente en los proyectos de software libre.





Seguridad

Seguridad

- La autenticación y autorización deben ser consideradas, especialmente cuando la naturaleza de la información es sensible o ponga en riesgo la infraestructura.
- El manejo de actualizaciones de seguridad también es muy importante.



Seguridad

- La autenticación y autorización deben ser consideradas, especialmente cuando la naturaleza de la información es sensible o ponga en riesgo la infraestructura.



Conclusiones RobotOps

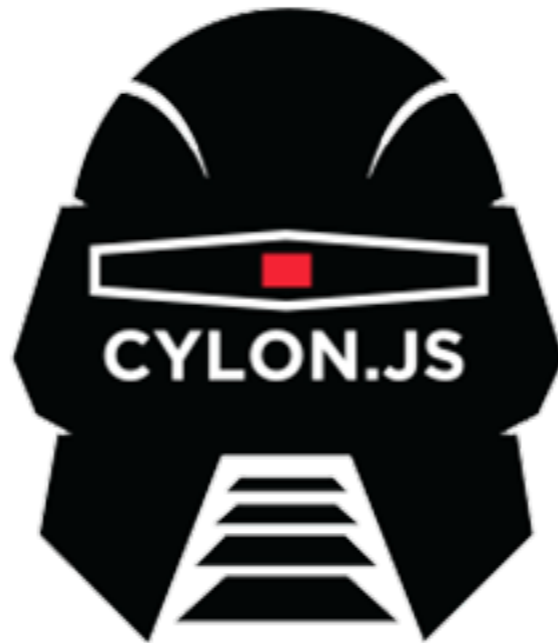
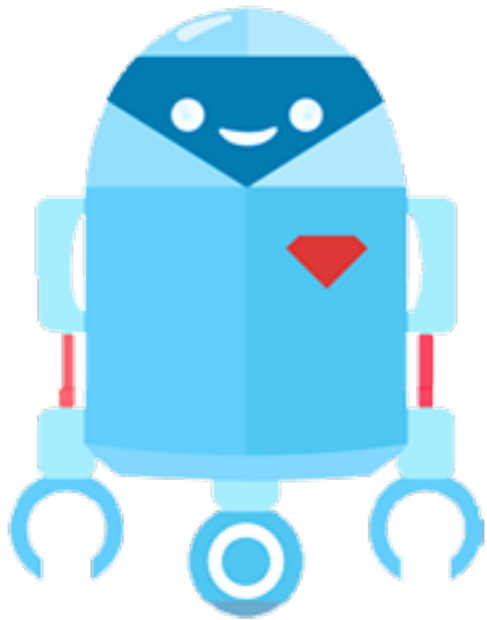
- Es esencial tomar en cuenta estos puntos para que la implementación del Internet de las cosas sea exitosa.



Conclusiones RobotOps

- Es esencial tomar en cuenta estos puntos para que la implementación del Internet de las cosas sea exitosa.
- Si trabajamos como industria, podemos acelerar la adopción, así como mejorar la experiencia de los usuarios.





Frameworks de robótica

RobotOps en funcionamiento

Frameworks de robótica

Frameworks de robótica

En **Hybrid Group**, hemos dedicado gran parte de nuestro tiempo pensando en estas y otras necesidades.



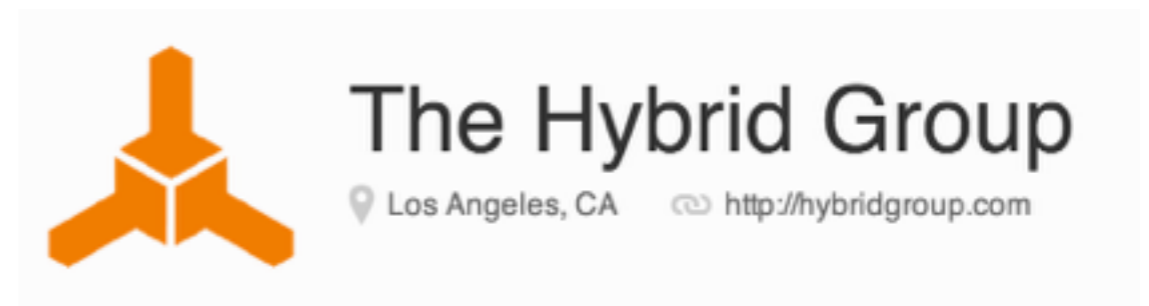
The Hybrid Group

Los Angeles, CA <http://hybridgroup.com>

Frameworks de robótica

En **Hybrid Group**, hemos dedicado gran parte de nuestro tiempo pensando en estas y otras necesidades.

Y hemos creado 3 frameworks de software libre:

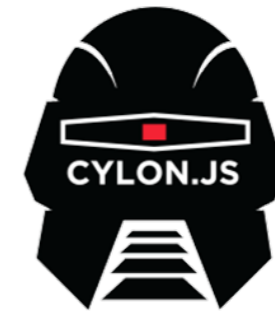
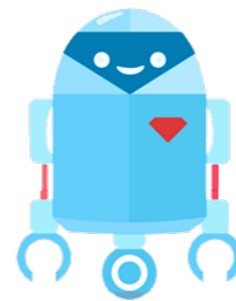


Frameworks de robótica

En **Hybrid Group**, hemos dedicado gran parte de nuestro tiempo pensando en estas y otras necesidades.

Y hemos creado 3 frameworks de software libre:

Cylon.js, **Artoo**, and **Gobot**



The Hybrid Group

Los Angeles, CA <http://hybridgroup.com>

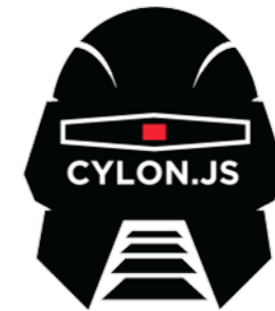
Frameworks de robótica

En **Hybrid Group**, hemos dedicado gran parte de nuestro tiempo pensando en estas y otras necesidades.

Y hemos creado 3 frameworks de software libre:

Cylon.js, **Artoo**, and **Gobot**

incorporando las reglas de **robotOps**

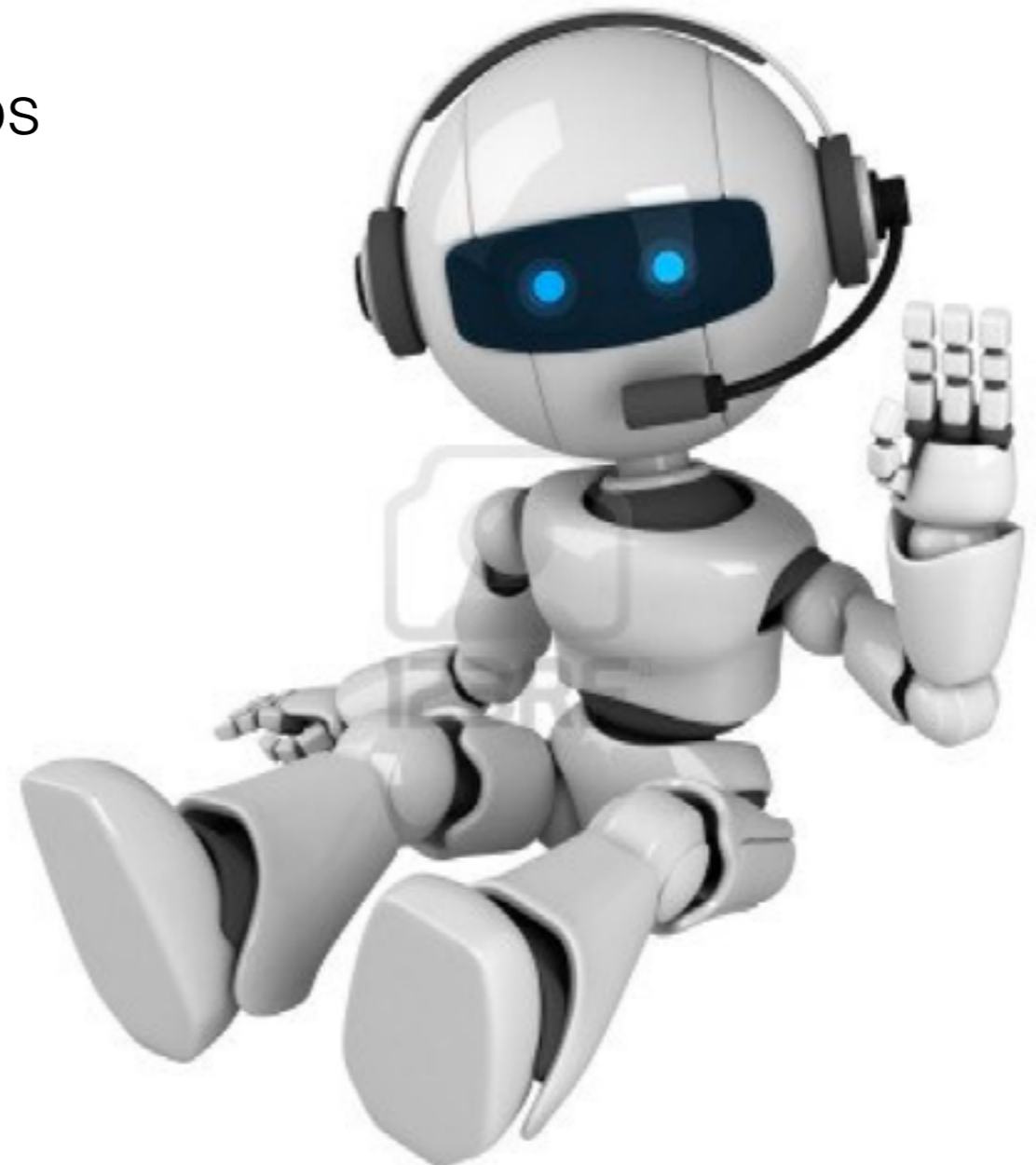


The Hybrid Group

Los Angeles, CA <http://hybridgroup.com>

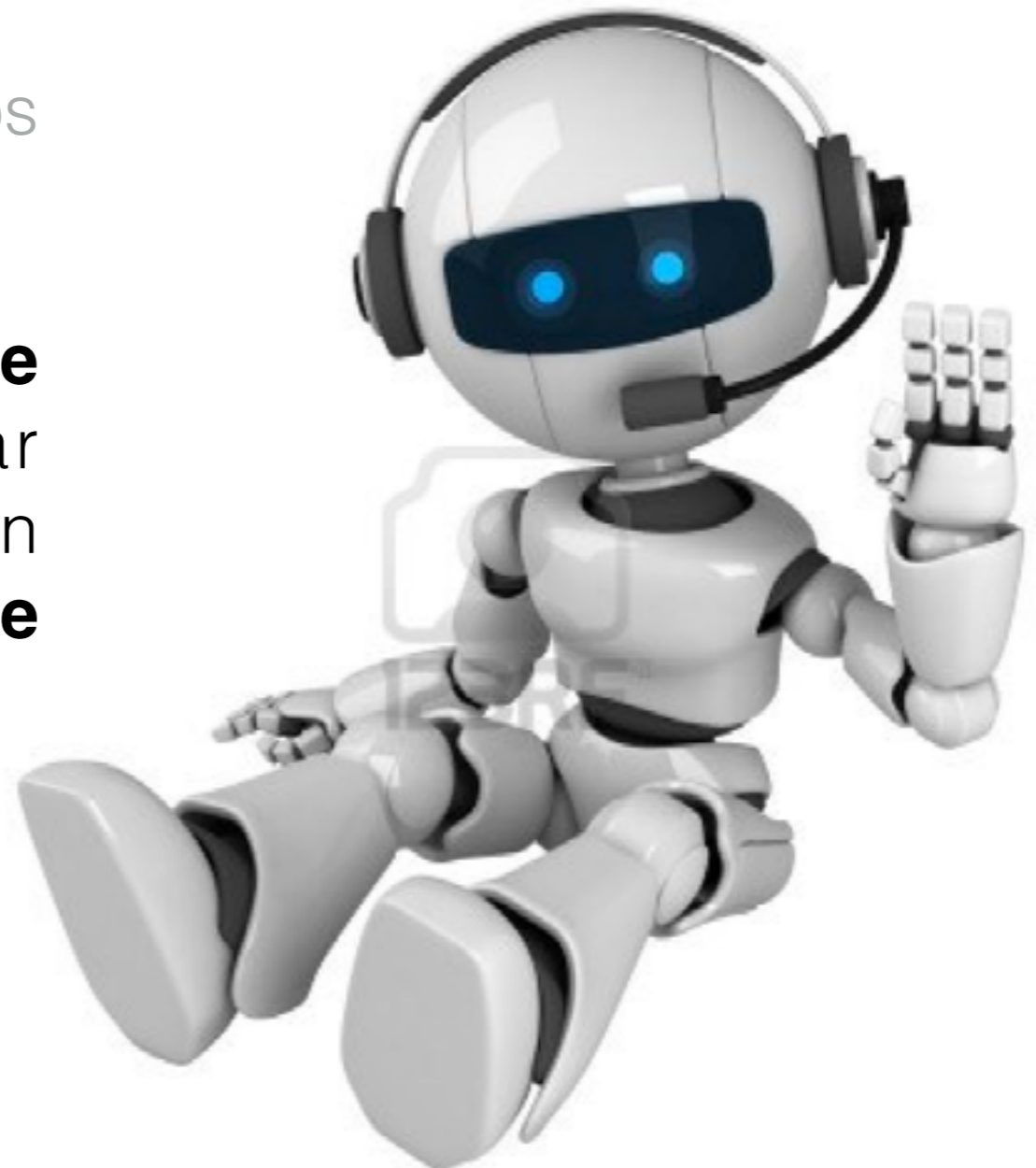
Frameworks de robótica

- Estos frameworks están escritos en **javascript**, **ruby** y **go**.



Frameworks de robótica

- Estos frameworks están escritos en **javascript**, **ruby** y **go**.
- Proveen una **interfaz simple** (pero poderosa) para crear soluciones que incorporen **múltiples dispositivos de hardware**

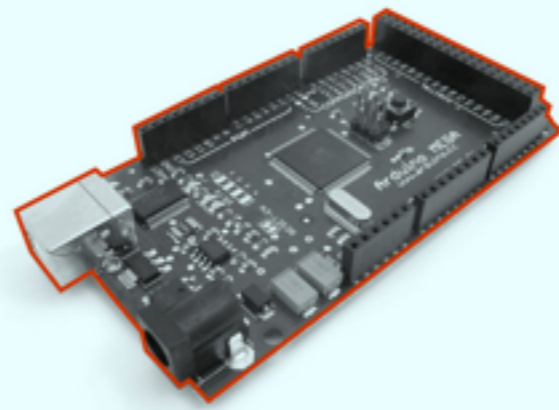




Dispositivos soportados



artoo Ardrone



artoo Arduino



artoo BeagleBone



artoo Crazyflie



artoo Digispark



artoo Joystick



artoo Keyboard



artoo Leap Motion



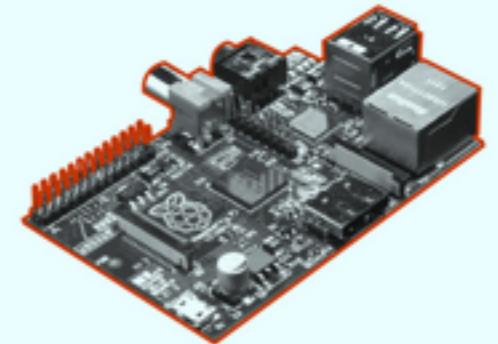
artoo Neurosky Mindwave



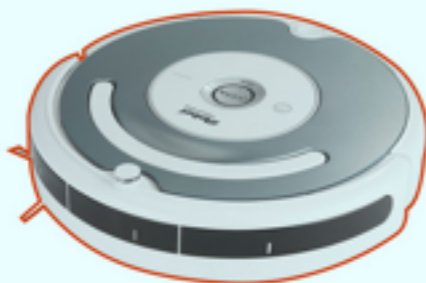
artoo OpenCV



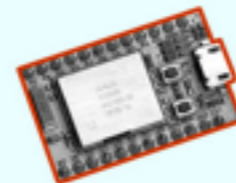
artoo Pebble



artoo Raspberry Pi



artoo Roomba



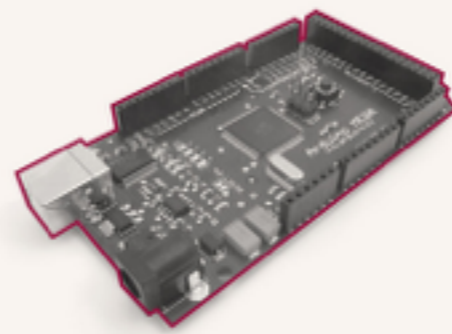
artoo Spark



artoo Sphero



CYLON.JS Ardrone



CYLON.JS Arduino



CYLON.JS BeagleBone



CYLON.JS Crazyflie



CYLON.JS Joystick



CYLON.JS Keyboard



CYLON.JS Digispark



CYLON.JS Force



CYLON.JS Leap Motion



CYLON.JS Neurosky Mindwave



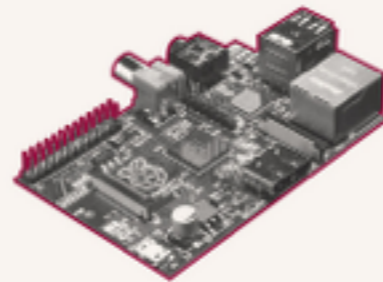
CYLON.JS OpenCV



CYLON.JS Pebble



CYLON.JS Rapiro



CYLON.JS Raspberry Pi



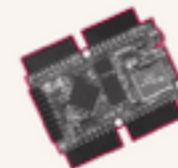
CYLON.JS Skynet



CYLON.JS Spark



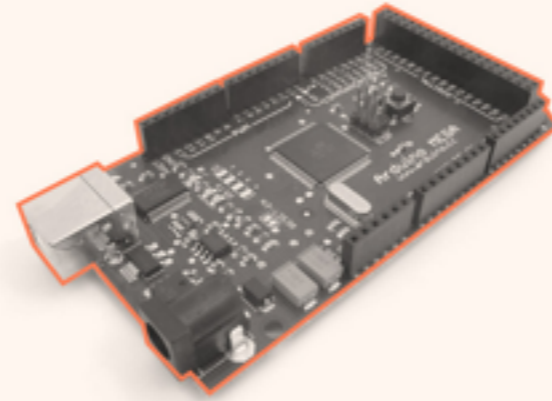
CYLON.JS Sphero



CYLON.JS Tessel



GOBOT Ardrone



GOBOT Arduino



GOBOT BeagleBone



GOBOT Digispark



GOBOT Joystick



GOBOT Leap Motion



GOBOT Neurosky Mindwave



GOBOT OpenCV



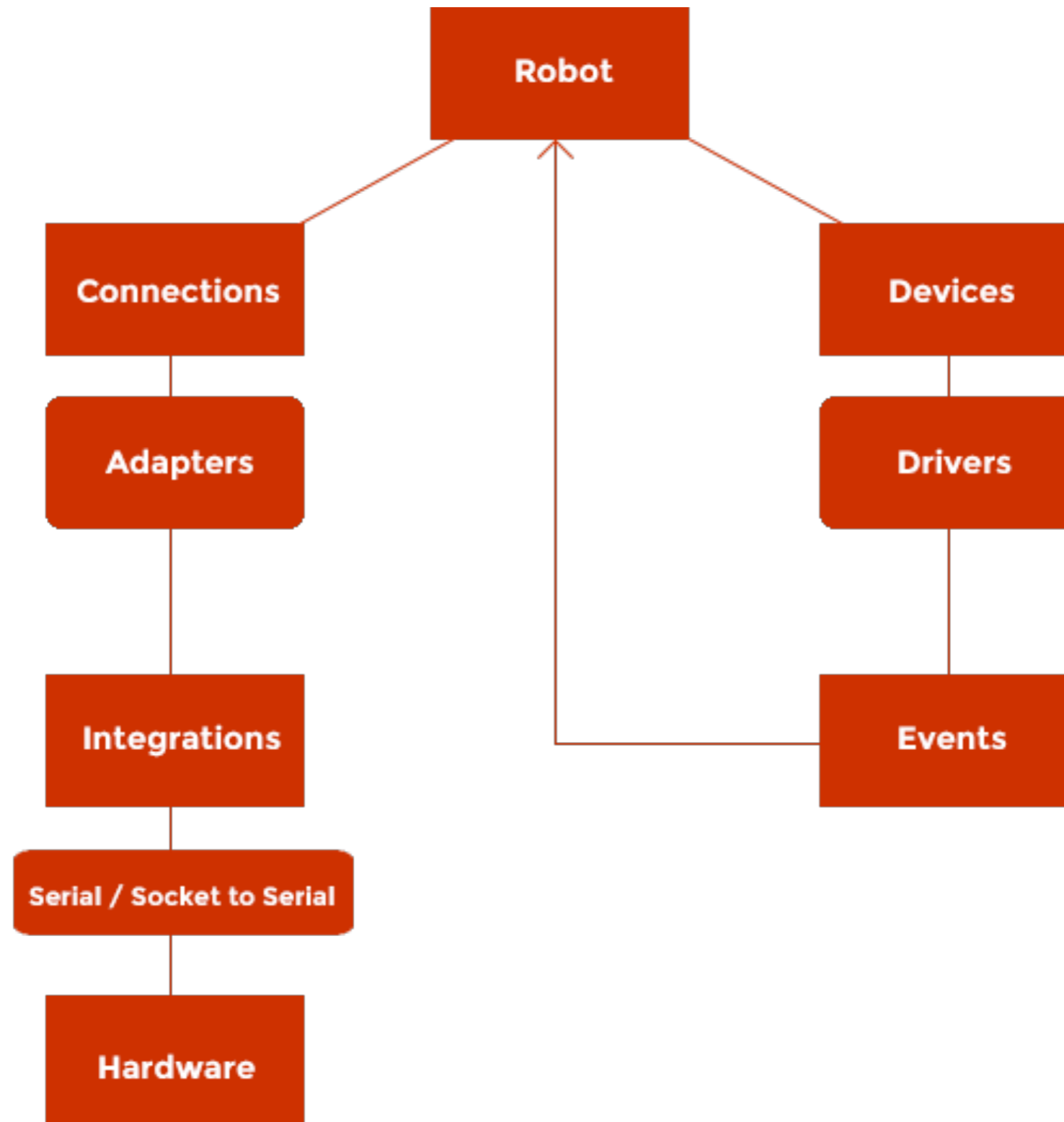
GOBOT Spark



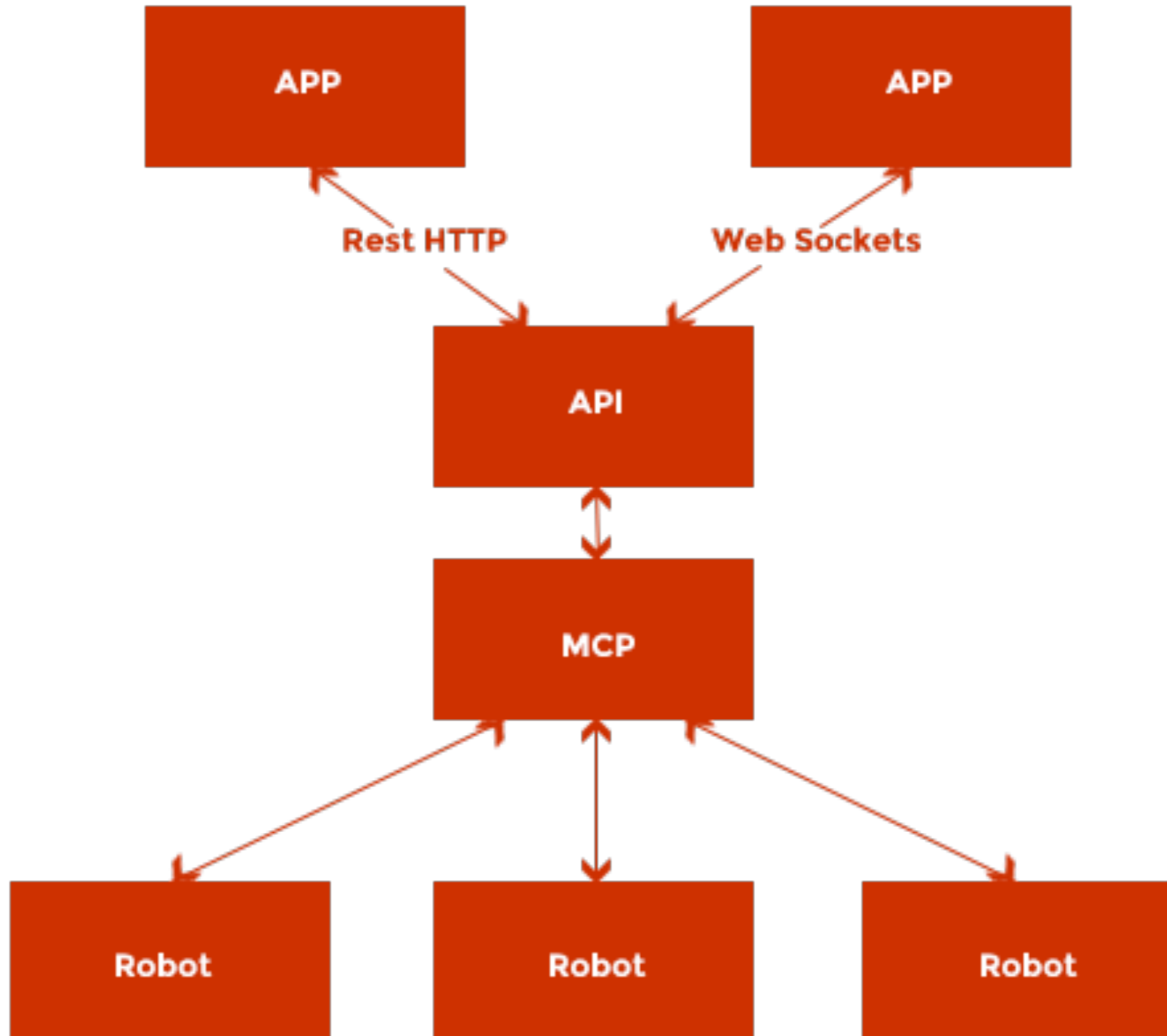
GOBOT Sphero

Arquitectura

Arquitectura

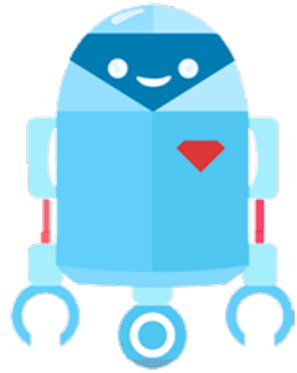


Arquitectura



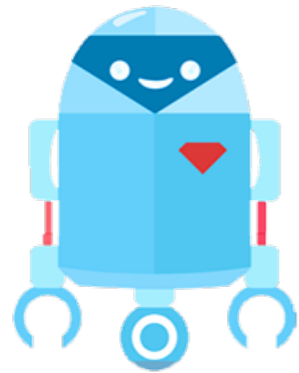
Instalación

Instalación

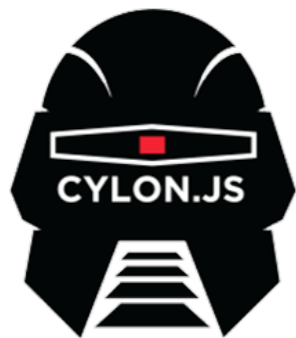


```
gem install artoo
```

Instalación

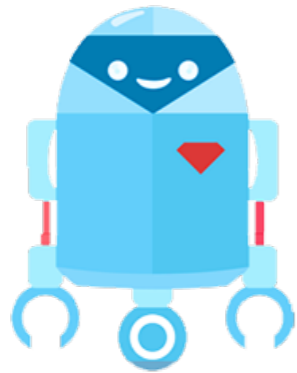


`gem install artoo`



`npm install cylon`

Instalación



```
gem install artoo
```



```
npm install cylon
```



```
go get -u github.com/hybridgroup/gobot
```

Ejemplos



Arduino + LED + Button

```
require 'artoo'

connection :arduino, :adaptor => :firmata, :port =>
  '/dev/ttyACM0'
device :led, :driver => :led, :pin => 13
device :button, :driver => :button, :pin => 2

work do
  on button, :push => proc {led.toggle}
end
```


Pebble + Sphero

```
require 'artoo'

connection :sphero, :adaptor => :sphero, :port => '127.0.0.1:1234'
device     :sphero, :driver  => :sphero

connection :pebble, :adaptor => :pebble
device     :watch,  :driver  => :pebble, :name => 'pebble'

api :host => '0.0.0.0', :port => '8080'

name 'pebble'

def move_forward
  p 'moving forward'
  sphero.roll 100, 0
  sleep 4
  sphero.stop
end

def move_backward
  p 'moving backward'
  sphero.roll 100, 180
  sleep 4
  sphero.stop
end
```

Pebble + Sphero

```
require 'artoo'

connection :sphero, :adaptor => :sphero, :port => '127.0.0.1:1234'
device     :sphero, :driver  => :sphero

connection :pebble, :adaptor => :pebble
device     :watch,  :driver  => :pebble, :name => 'pebble'

api :host => '0.0.0.0', :port => '8080'

name 'pebble'

def move_forward
  p 'moving forward'
  sphero.roll 100, 0
  sleep 4
  sphero.stop
end

def move_backward
  p 'moving backward'
  sphero.roll 100, 180
  sleep 4
  sphero.stop
end

def button_push(*data)
  unless data[1].nil?
    case data[1]
    when 'up' then
      move_forward
    when 'select' then
      sphero.set_color(rand(255),rand(255),rand(255))
    when 'down' then
      move_backward
    end
  end
end

work do
  on pebble, :button => :button_push
end
```

Parrot ARDrone

```
var Cylon = require('cylon');

Cylon.robot({
  connection: { name: 'ardrone', adaptor: 'ardrone', port: '192.168.1.1' },
  device: { name: 'drone', driver: 'ardrone' },

  work: function(my) {
    my.drone.takeoff();
    after(10..seconds(), function() { my.drone.land(); });
    after(15..seconds(), function() { my.drone.stop(); });
  }
}).start();
```

Leap motion + arduino

```
var Cylon = require('../..');

Cylon.robot({
  connections: [{ name: 'leapmotion', adaptor: 'leapmotion', port: '127.0.0.1:6437' },
    { name: 'arduino', adaptor: 'firmata', port: '/dev/ttyACM0' }],

  devices: [{ name: 'leapmotion', driver: 'leapmotion', connection: 'leapmotion' },
    { name: 'led', driver: 'led', pin: 13, connection: 'arduino' }],

  work: function(my) {
    my.leapmotion.on('frame', function(frame) {
      if (frame.hands.length > 0) {
        my.led.turnOn();
      } else {
        my.led.turnOff();
      }
    });
  }
}).start();
```

Sphero

```
package main

import (
    "github.com/hybridgroup/gobot"
    "github.com/hybridgroup/gobot-sphero"
)

func main() {

    spheroAdaptor := new(gobotSphero.SpheroAdaptor)
    spheroAdaptor.Name = "Sphero"
    spheroAdaptor.Port = "/dev/rfcomm0"

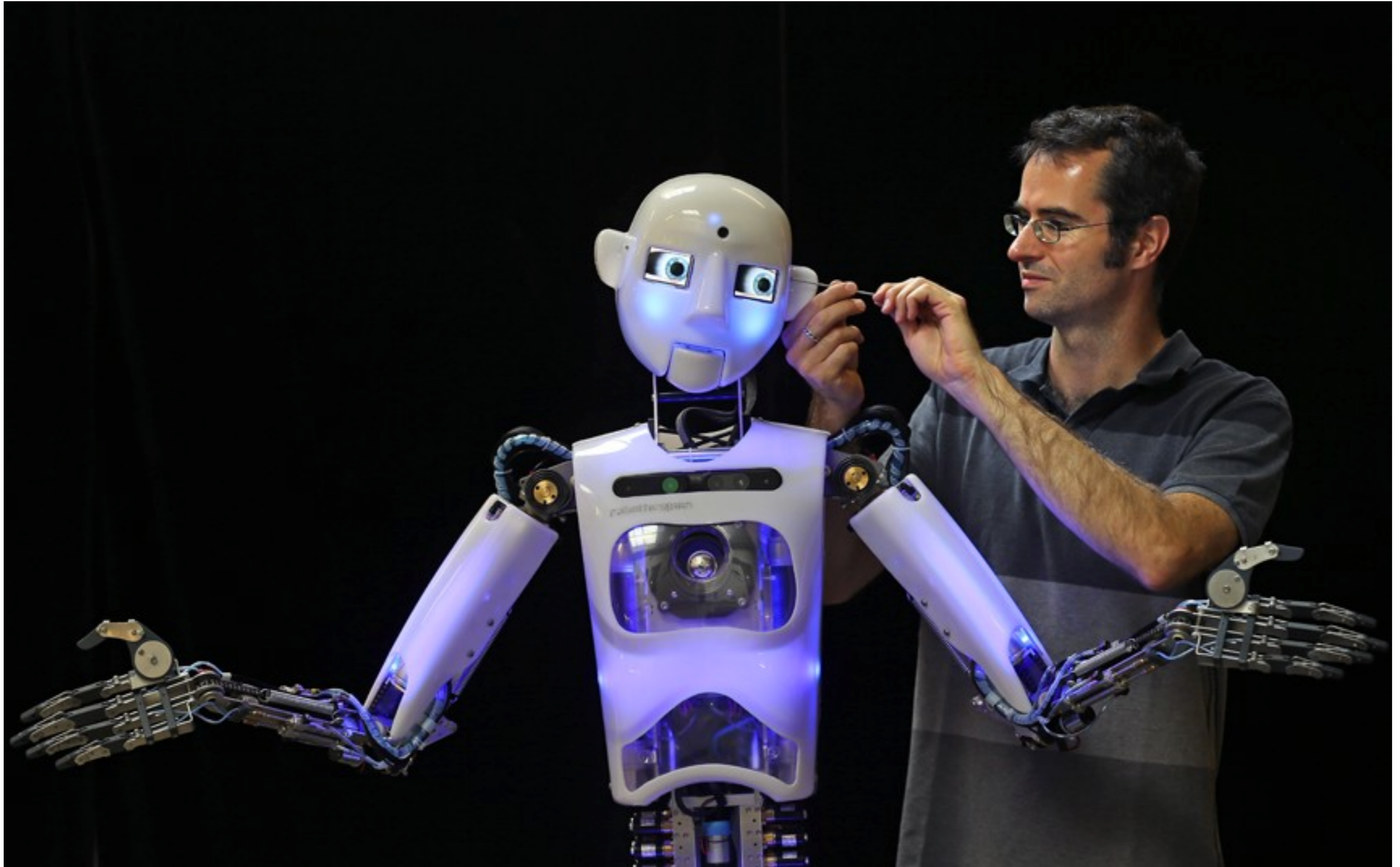
    sphero := gobotSphero.NewSphero(spheroAdaptor)
    sphero.Name = "Sphero"

    work := func() {
        gobot.Every("2s", func() {
            sphero.Roll(100, uint16(gobot.Rand(360)))
        })
    }

    robot := gobot.Robot{
        Connections: []gobot.Connection{spheroAdaptor},
        Devices:      []gobot.Device{sphero},
        Work:        work,
    }

    robot.Start()
}
```

+ RobotOps



API

The screenshot shows the Robeaux web interface in a browser window. The URL is `https://localhost:3000/#/robots/PingBot`. The interface has a header with the title "Robeaux" and navigation tabs for "robots" and "themes". Below the header, a "robot" tab is selected, showing the robot name "PingBot" and statistics: "Connections: 1" and "Devices: 2".

The main content area is divided into two columns: "Commands" and "Events".

Commands Panel:

- A dropdown menu is set to "ping".
- Fields for "name" and "value" are present, with "value" set to "string".
- A "run" button is visible.
- A "Results" section shows the output "pong".

Events Panel:

- A "Listeners" section lists "heartbeat" and "random", each with a "remove" button.
- An "event name" field is present with a "listen" button.
- An "Events" section displays a list of events:

```
heartbeat {"hello":"world"}
random {"random":22}
heartbeat {"hello":"world"}
heartbeat {"hello":"world"}
heartbeat {"hello":"world"}
```

At the bottom, there are panels for "device" and "connection":

- A green "device" panel for "ping" on "loopback".
- A "device" panel for "ping2" on "loopback".
- A "connection" panel for "loopback" on "/dev/ttyACM0".

Linea de comandos

```
$ ./gort
```

NAME:

```
gort - Command Line Utility for RobotOps
```

USAGE:

```
gort [global options] command [command options] [arguments...]
```

VERSION:

```
0.0.1
```

COMMANDS:

```
scan      Scan for connected devices on Serial, USB, or Bluetooth ports
```

```
arduino  Install avrdude, and upload sketches to your Arduino
```

```
bluetooth Scan, pair, unpair bluetooth devices. Establishes serial to Bluetooth connection.
```

```
help, h  Shows a list of commands or help for one command
```

GLOBAL OPTIONS:

```
--version, -v    print the version
```

```
--help, -h      show help
```


Test Driven Robotics

```
require 'artoo'

connection :sphero, :adaptor => :sphero, :port => '127.0.0.1:4560'
device :sphero, :driver => :sphero

def contact(*args)
  @contacts ||= 0
  @contacts += 1
  puts "Contact "
end

work do
  on sphero, :collision => :contact

  every(3.seconds) do
    sphero.roll 90, rand(360)
  end
end
```

Test Driven Robotics

```
require './test_helper'  
require './test_robot'  
  
describe 'sphero' do  
  let(:robot) { Artoo::MainRobot.new }  
  let (:start) { Time.now }  
  
  before :each do  
    Timecop.travel(start)  
    robot.work  
  end  
  
  after :each do  
    Timecop.return  
  end  
  
  it 'has work to do every 3 seconds' do  
    robot.has_work?(:every, 3.seconds).wont_be_nil  
  end  
end
```

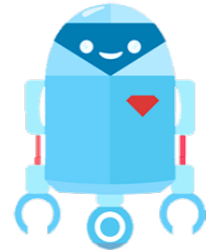
Test Driven Robotics

```
it 'receives collision event' do
  robot.expects(:contact)
  robot.sphero.publish("collision", "clunk")
  sleep 0.05
end

it 'must roll every 3 seconds' do
  Timecop.travel(start + 3.seconds) do
    robot.sphero.expects(:roll)
    sleep 0.05
  end
  Timecop.travel(start + 6.seconds) do
    robot.sphero.expects(:roll)
    sleep 0.05
  end
end
end
end
```

Documentación

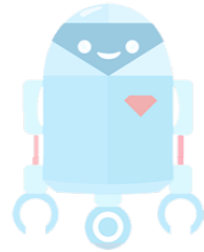
Documentación



Artoo

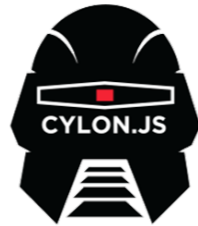
<http://artoo.io>

Documentación



Artoo

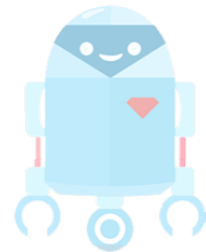
<http://artoo.io>



Cylon

<http://cylonjs.com>

Documentación



Artoo

<http://artoo.io>



Cylon

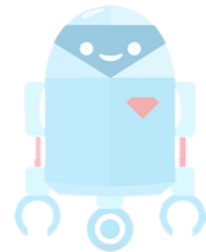
<http://cylonjs.com>



Gobot

<http://gobot.io>

Documentación



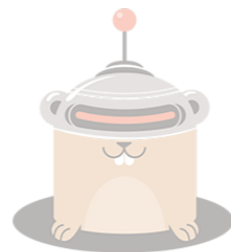
Artoo

<http://artoo.io>



Cylon

<http://cylonjs.com>



Gobot

<http://gobot.io>



RobotOps

<http://robotops.com>

Pasa la voz

Frameworks para **#losrobots**
@artooio @gobotio @cylonjs
via **@hybrid_group**



SG 
VIRTUAL
CONFERENCE
6ta edición

Re-evolución Robótica

Frameworks para **#losrobots**
@artooio @gobotio @cylonjs
via **@hybrid_group**

Presentado por:
Javier Cervantes Ponce
@solojavier