



WORKSHOP ORACLE ACADEMY

10:10h a 11:10h

Programando el robot Finch en Java con una Raspberry Pi

Ponente: Jordi Ariño, Software Developer Manager at PUE



Programming the Finch Robot in Java





Jordi Ariño
Software Developer Manager at PUE
Instructor oficial de Oracle

jordi.arino@pue.es @jordiAS2K











Designed for Computer Science Education



Introducción

Programming the Finch Robot in Java



Finch es un pequeño **robot** diseñado para el **ámbito educativo**, con el objetivo de acercar el **coding** a los estudiantes desde una edad temprana, mientras aprenden a programar jugando.

Este robot ofrece soporte para más de una docena de lenguajes de programación -Java, Python,...-, así como para entornos gráficos de programación por bloques que permiten interactuar fácilmente con sus componentes y sensores.







Programming the Finch Robot in Java



En este **workshop** obtendrás una **visión práctica** de cómo interactuar mediante Java –y algún otro entornocon los distintos componentes y sensores disponibles en el robot *Finch*.

Además veremos como integrar una **Raspberry Pi** con el robot *Finch,* para que éste se vuelva un dispositivo mucho más potente y autónomo.

Esta actividad se enmarca dentro del programa educativo **Oracle Academy.**





Proyecto: Oracle Academy



Ofrece una solución completa de recursos para la educación en TIC, con el objetivo de ayudar a los estudiantes a prepararse para su futuro profesional.

PUE es el **partner** escogido por *Oracle* para la difusión y gestión de su iniciativa *Oracle Academy* en España.

129 centros participan activamente en este proyecto y reciben soporte activo por parte de PUE para la correcta implementación de los recursos docentes que tienen a su disposición.

www.pue.es/oracle-academy





Oracle Academy: Cursos







Java Fundamentals



Java Programming



Oracle Application
Express – Application
Development
Foundations



Database Foundations



Database Design and Programming with SQL



Programming with PL/SQL





Oracle Academy: Workshops



Getting Started with Java Using Alice



Solve it with SQL



Creating Java Programs with Greenfoot



Programming the Finch Robot





Oracle Academy: Workshops



Getting Started with Java Using Alice



Solve it with SQL



Creating Java Programs with Greenfoot



Programming the Finch Robot









Designed for Computer Science Education



El robot Finch



Dimensiones y peso





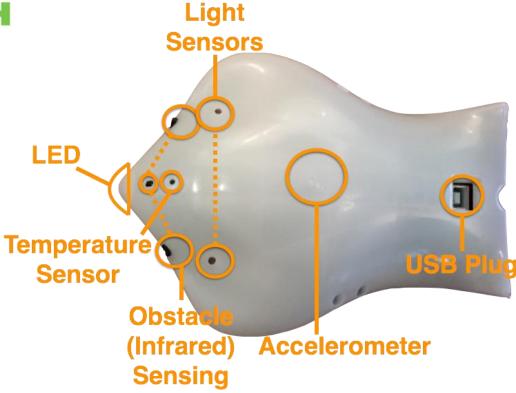
Finch mide 17.8cm desde el pico a la cola, 14cm de ala a ala y 7.6cm de alto.

Su peso aproximado es de 400g.





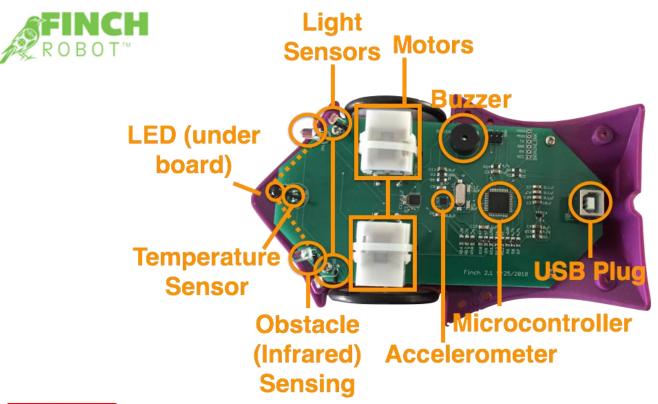




















Protocolo de comunicación

El protocolo de comunicación de Finch con el computador es mediante una conexión USB.

Se clasifica como un **dispositivo USB HID** -la misma categoría que los mouses y teclados-.

Como dispositivo USB HID, Finch hace uso de drivers genéricos por lo que no es necesario instalar drivers adicionales.

Finch viene de serie con un **cable USB A to B** de 4.5 metros de largo y no necesita ningún tipo de batería ni fuente de alimentación.







LED: Emisión de luz

Integrado un LED a todo color incrustado en su "pico".

Se trata de un **LED RGB** que permite mostrar cualquier color ajustando la intensidad o tonalidad de los diferentes componentes R, G y B.

Para cada componente RGB se pueden configurar 256 valores de tonalidad o intensidad (0-255).









Buzzer: Emisión de sonidos

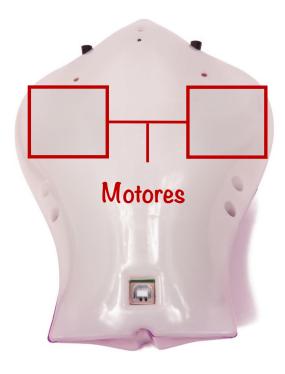
Integrado un *buzzer* o timbre eléctrico (zumbador) capaz de reproducir sonidos con frecuencias entre 100Hz y 10KHz.

Además, lleva integrado un componente software *-FreeTTS-* que permite reproducir frases sintetizadas.









Motores y ruedas

Integrados dos pequeños motores de engranaje que permiten que se pueda mover y girar sobre si mismo mediante dos ruedas.

Las ruedas están ajustadas a presión en los ejes del motor.

La velocidad máxima que puede alcanzar *Finch* es de 38 cm por segundo aprox.









Detectores de obstáculos

Integrados **dos sensores infrarrojos** que permiten detectar obstáculos delante de él a una distancia de 5cm aprox.

Los sensores son digitales y actúan como **parachoques virtuales**.

No proporcionan información sobre la distancia a un obstáculo, sino que simplemente registran si existe un obstáculo-.







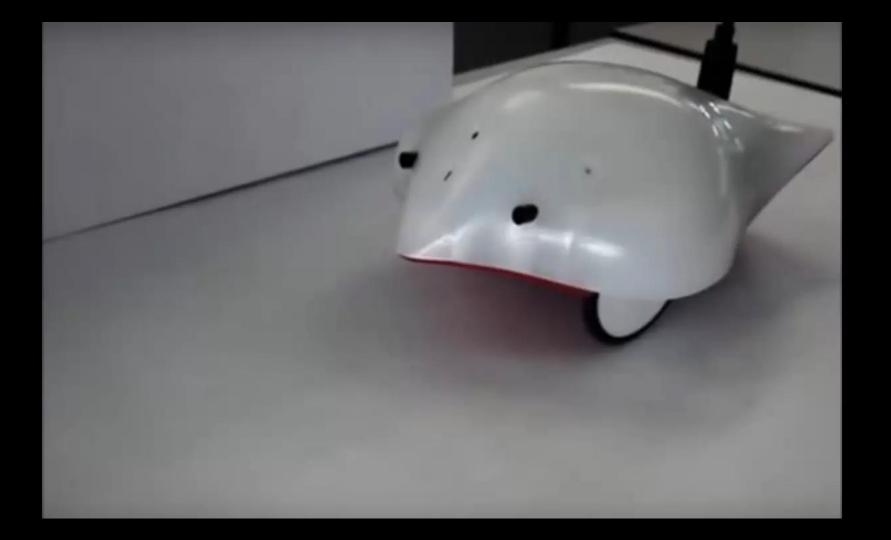
Detectores de obstáculos

Objetos muy estrechos u objetos compuestos de ciertos plásticos negros pueden no registrarse como obstáculos.

Se utilizan dos LEDs infrarrojos y un receptor de infrarrojos para detectar obstáculos que ofrece una forma eficaz y de bajo costo para detectar la mayoría de los obstáculos, pero es limitada.









Sensor de temperatura

Integrado un **termistor** o sensor de temperatura por resistencia que permite **detectar la temperatura ambiente**.

También puede determinar la temperatura de un objeto, si el objeto se pone en contacto con el sensor.

Este sensor tiene una precisión de $\pm 2^{\circ}$ C.









Sensores de luz

Integrados **dos fotoresistores** que permiten detectar los **niveles de luz ambiental**.

Estos sensores son analógicos con 8 bits de resolución y pueden diferenciar fácilmente los días soleados y nublados y el brillo de varias fuentes de luz interiores.

Estos sensores se usan comúnmente como dispositivos de entrada o para permitir que Finch se aleje o acerque de las fuentes u orígenes de luz.









Acelerómetro

Integrado un **acelerómetro** de tipo Freescale MMA7660FC MEMS de 3 ejes para detectar la aceleración.

Permite detectar la **dirección de la gravedad**, y así saber cómo está orientado Finch (plano en el suelo, vertical, volteado, etc).

También es posible detectar picos de aceleración causados por golpear o agitar a Finch.

El sensor puede detectar aceleraciones de $\pm 1,5$ Gs.











Designed for Computer Science Education



Programando a Finch



Java, Python, C, C++, C#, VB, Scala, Javascript, Snap!, Scratch,...

¿Cómo programar a Finch?

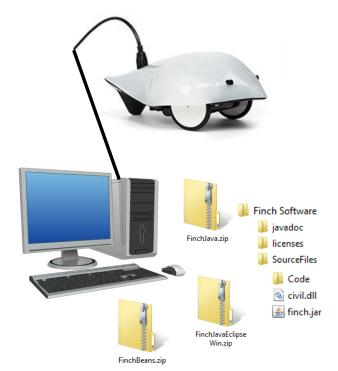
Finch funciona con Windows (XP, Vista, 7, 8 y 10), Mac OS 10.5 y superior, y la mayoría de las distribuciones de Linux.

Soporte para más de una docena de lenguajes de programación así como para entornos gráficos de programación por bloques que permiten interactuar fácilmente con sus componentes y sensores.

Snap!	0011 0 0100	Middle School		■ € △ ⊙	(4)
Scratch	Elementary			■ 6 0 0	
Python			High School+	# £ \$ 0	
Processing				€ € Å	
Greenfoot				≡ € ∆	







Java

```
Finch myFinch = new Finch();

myFinch.saySomething ("Hello World");
myFinch.setLED(0,255,255);
myFinch.sleep(2000);

myFinch.quit();
```

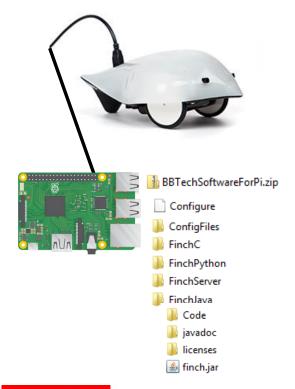
Online javadoc: http://finchrobot.com/javadoc

Finch no es compatible con Java 5 o versiones anteriores de Java









Java

```
Finch myFinch = new Finch();

myFinch.saySomething ("Hello World");
myFinch.setLED(0,255,255);
myFinch.sleep(2000);

myFinch.quit();
```

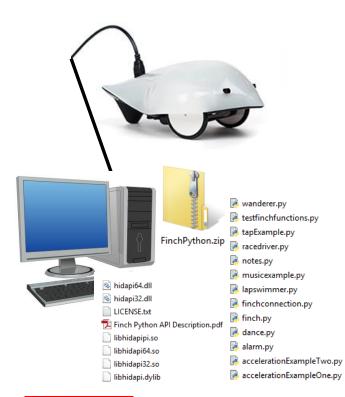
Online javadoc: http://finchrobot.com/javadoc

Finch no es compatible con Java 5 o versiones anteriores de Java









Python

from finch import Finch
import time

finch = Finch()

finch.wheels(0.5,-0.5)
time.sleep(0.75)
finch.wheels(0,0)

time.sleep(1)

finch.wheels(0.5,0)
time.sleep(0.75)
finch.wheels(0,0)

for the Finch
for sleep()

declare the Finch object

first turn

pause

second turn

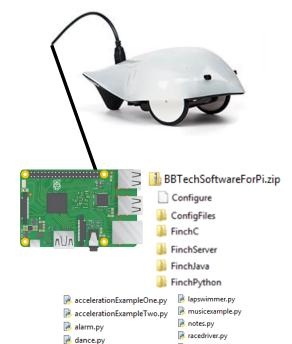




tapExample.py

wanderer.py

testfinchfunctions.py



Python

from finch import Finch
import time

finch = Finch()

finch.wheels(0.5,-0.5)
time.sleep(0.75)
finch.wheels(0,0)

time.sleep(1)

finch.wheels(0.5,0)
time.sleep(0.75)
finch.wheels(0,0)

for the Finch
for sleep()

declare the Finch object

first turn

pause

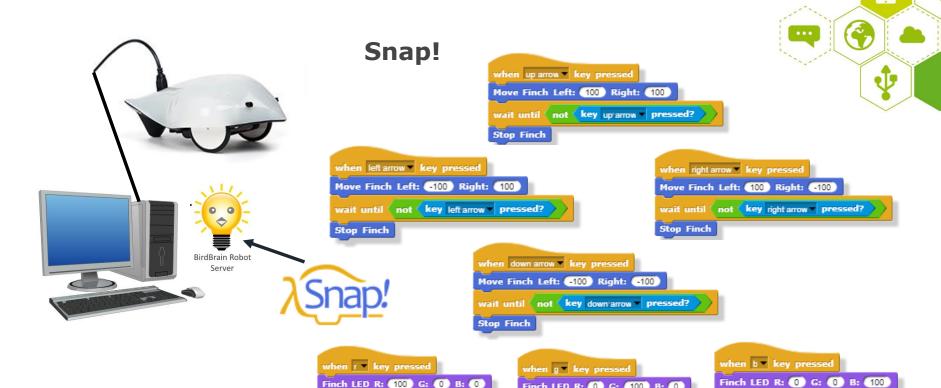
second turn



finch.py

finchconnection.py



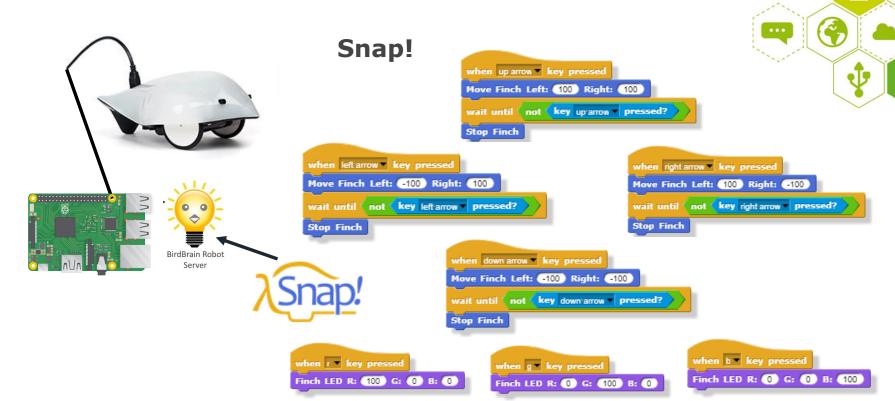


Finch LED R: 0 G: 100 B: 0

Finch LED R: 100 G: 0 B: 0













Scratch

```
when up arrow ▼ key pressed
                             Move Finch left: 100 right: 100
                              wait until not key up arrow ▼ pressed?
                             Move Finch left: 0 right: 0
when left arrow ▼ key pressed
                                                               when right arrow ▼ key pressed
Move Finch left: -100 right: 100
                                                               Move Finch left: 100 right: -100
 wait until not key left arrow pressed?
                                                               wait until not key right arrow pressed?
Move Finch left: 0 right: 0
                                                               Move Finch left: 0 right: 0
                               when down arrow key pressed
                               Hove Finch left: -100 right: -100
                               wait until not key down arrow pressed?
                               Hove Finch left: 0 right: 0
when r v key pressed
                                         when g v key pressed
Finch LED color R: 100 G: 0 B: 0
                                         Finch LED color R: 0 G: 100 B: 0
```

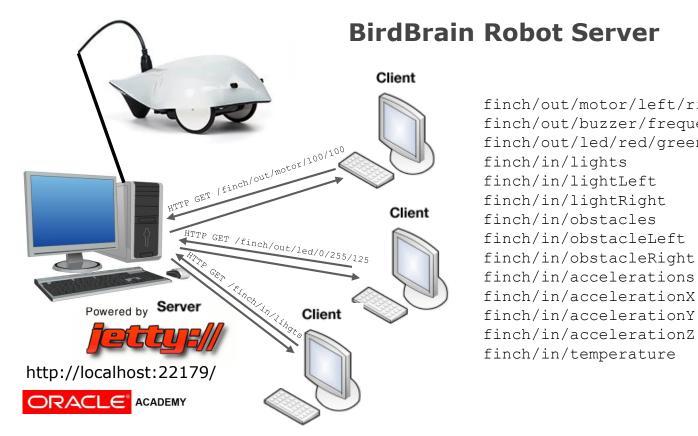


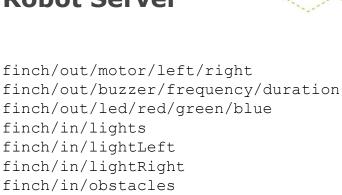




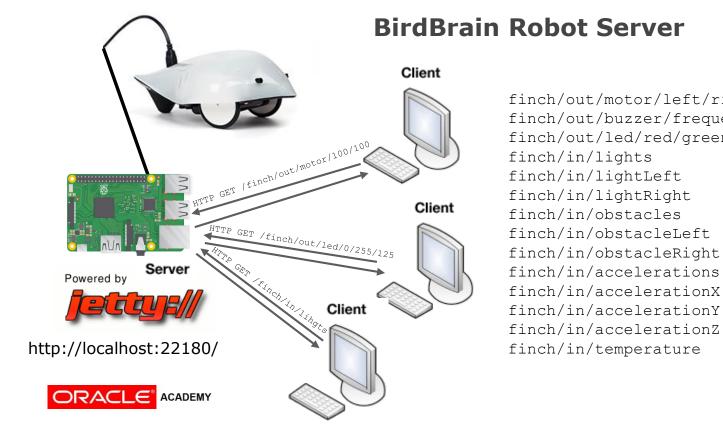
Finch LED color R: 0 G: 0 B: 100

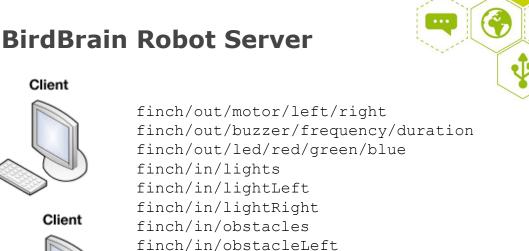
when b v key pressed

















Designed for Computer Science Education

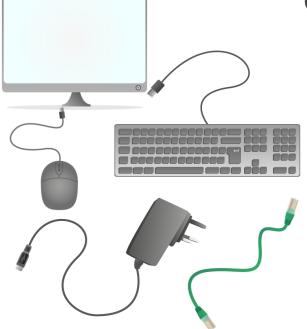


Finch con una Raspberry Pi

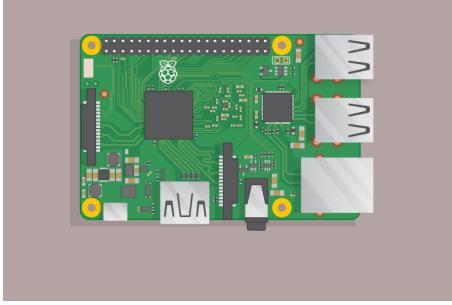








Componentes necesarios

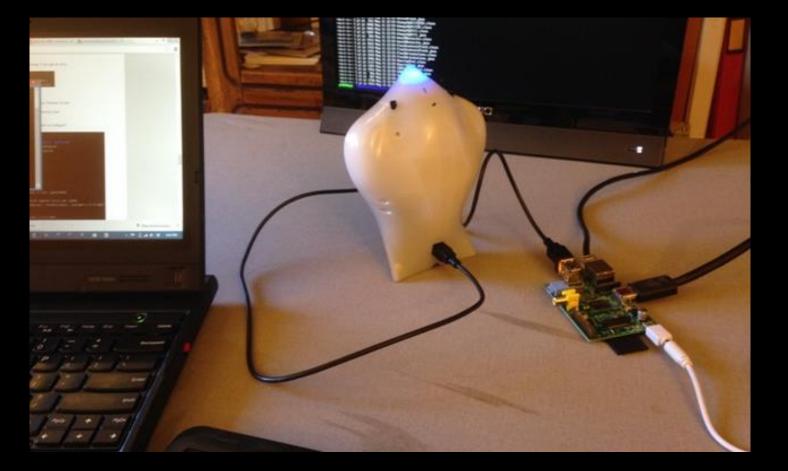




RASPBIAN

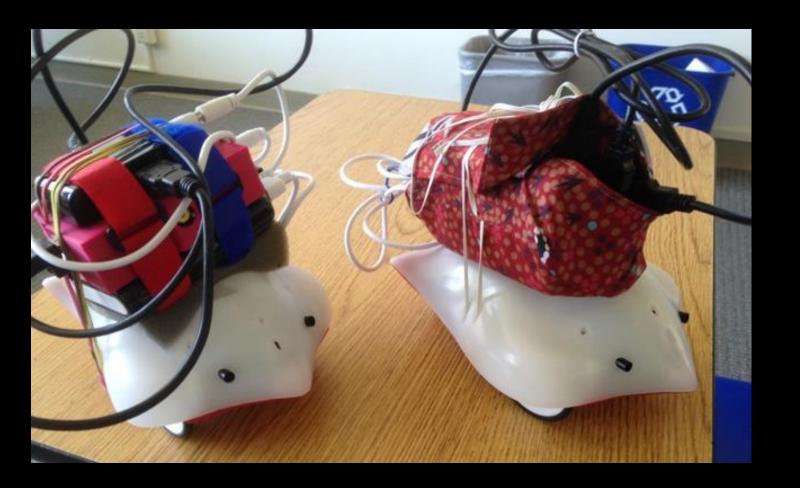






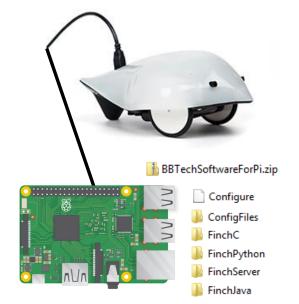






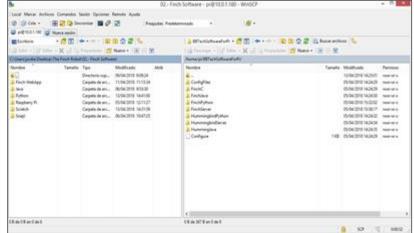








WinSCP

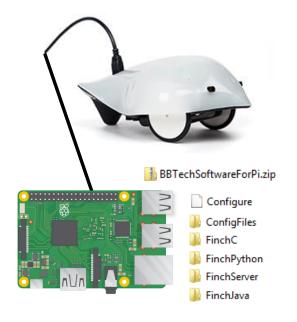






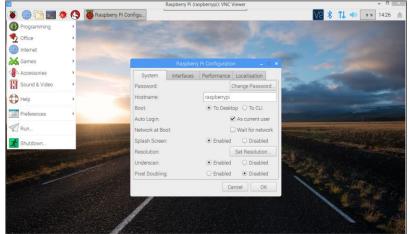








REALVIC

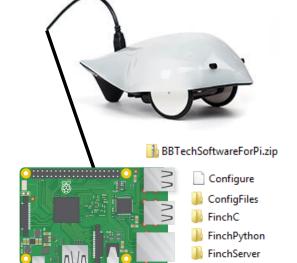




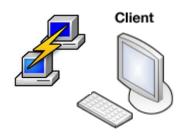




Conexión remota a Raspberry Pi



FinchJava



PuTTY

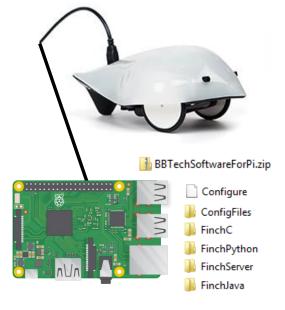
login as: pi
pi@10.0.1.180's password:
Linux raspberrypi 4.9.80-v7+ #1098 SMP Fri Mar 9 19:11:42 GMT 2018 armv71

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
Last login: Fri Apr 13 09:35:24 2018 from 10.0.1.225
pi@raspberrypi:~ \$





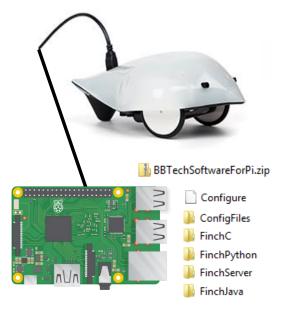


Primeros pasos con Finch

```
oi@raspberrypi:~/BBTechSoftwareForPi $ chmod +x Configure
pi@raspberrypi:~/BBTechSoftwareForPi $ 1s -1
otal 36
drwxr-xr-x 2 pi pi 4096 Apr 5 14:24 ConfigFiles
-rwxr-xr-x 1 pi pi 367 Apr 5 14:24 Configure
drwxr-xr-x 3 pi pi 4096 Apr 5 14:24 FinchC
drwxr-xr-x 5 pi pi 4096 Apr 5 14:24 FinchJava
drwxr-xr-x 2 pi pi 4096 Apr 5 14:24 FinchPython
drwxr-xr-x 3 pi pi 4096 Apr 5 14:24 FinchServer
drwxr-xr-x 2 pi pi 4096 Apr 5 14:24 HummingbirdPython
drwxr-xr-x 3 pi pi 4096 Apr 5 14:24 HummingbirdServer
drwxr-xr-x 5 pi pi 4096 Apr 5 14:24 HummingJava
```





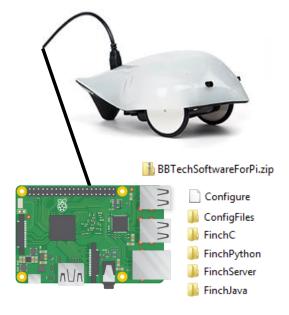


Primeros pasos con Finch

```
i@raspberrypi:~/BBTechSoftwareForPi $ sudo ./Configure
 opying libhidapi32.so to /usr/lib/
 opying libhidapi-jni.so to /usr/lib/
 opying finch and hummingbird udev rules
 eading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
 libusb-1.0-doc
he following NEW packages will be installed:
 libusb-1.0-0-dev libusb-1.0-doc
 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Weed to get 244 kB of archives.
After this operation, 1,711 kB of additional disk space will be used.
o you want to continue? [Y/n] Y
Get:1 http://ftp.cica.es/mirrors/Linux/raspbian/raspbian stretch/main armhf libusb-1.0-0-dev armhf 2:1
Get:2 http://ftp.cica.es/mirrors/Linux/raspbian/raspbian stretch/main armhf libusb-1.0-doc all 2:1.0.2
Fetched 244 kB in 0s (451 kB/s)
Selecting previously unselected package libusb-1.0-0-dev:armhf.
(Reading database ... 124889 files and directories currently installed.)
Preparing to unpack .../libusb-1.0-0-dev 2%3a1.0.21-1 armhf.deb ...
 npacking libusb-1.0-0-dev:armhf (2:1.0.21-1) ...
Selecting previously unselected package libusb-1.0-doc.
Preparing to unpack .../libusb-1.0-doc 2%3a1.0.21-1 all.deb ...
Inpacking libusb-1.0-doc (2:1.0.21-1) ...
Setting up libusb-1.0-doc (2:1.0.21-1) ...
Setting up libusb-1.0-0-dev:armhf (2:1.0.21-1) ...
```





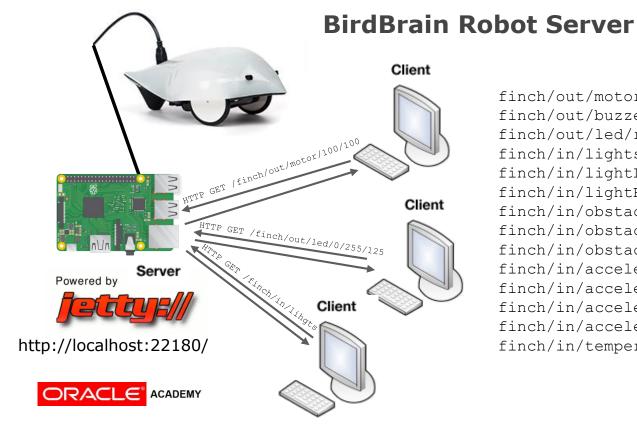


Primeros pasos con Finch

```
pi@raspberrypi:~/BBTechSoftwareForPi/FinchServer $ java -jar FinchServer.jar
Connecting to Finch...this may take a few seconds...
Finch server started!
A Finch was found!
2018-04-13 09:35:49.997:INFO:oejs.Server:jetty-8.y.z-SNAPSHOT
2018-04-13 09:35:50.689:INFO:oejs.AbstractConnector:Started SelectChannelConnect
or@0.0.0.0:22180
The IP address of this device is /10.0.1.180 - ignore the leading /
Press ENTER to quit.
```









finch/in/temperature



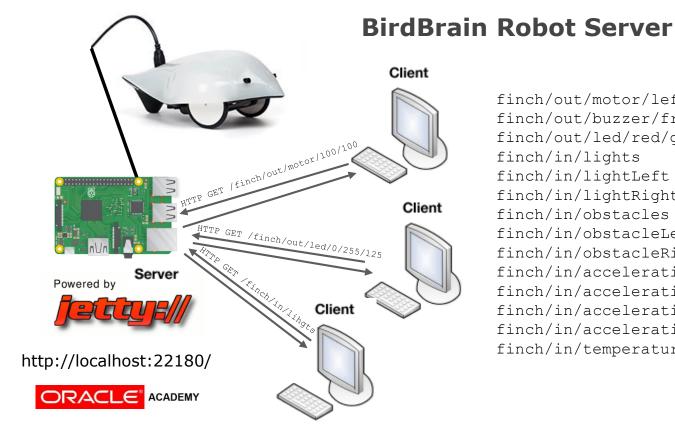




Designed for Computer Science Education



Hands-On Labs





finch/out/motor/left/right finch/out/buzzer/frequency/duration finch/out/led/red/green/blue finch/in/lights finch/in/lightLeft finch/in/lightRight finch/in/obstacles finch/in/obstacleLeft finch/in/obstacleRight finch/in/accelerations finch/in/accelerationX finch/in/accelerationY finch/in/accelerationZ finch/in/temperature















Finch dice: Escribe tu mensaje aquí...

Acceso a modo piano



Java Finch API



```
myFinch.setLED(255, 0, 0, 1000);
```

```
myFinch.setLED(255, 0, 0);
```

red: valores entre 0-255 green: valores entre 0-255 blue: valores entre 0-255

duration: duración en milisegundos (1s = 1000ms)









Java Finch API

```
myFinch.buzz(252, 1000);
myFinch.sleep(1000);
```

```
myFinch.buzzBlocking(252, 1000);
```

Reproduce un tono a la frecuencia especificada (Hz) y durante la duración especificada (ms) en el buzzer interno de Finch.

El método buzz() es no bloqueante por lo que si llama a dos métodos buzz() seguidos sin una pausa, sólo se escuchará el segundo tono.









Java Finch API

```
myFinch.saySomething("hello");
myFinch.saySomething("hello", 5000);
```

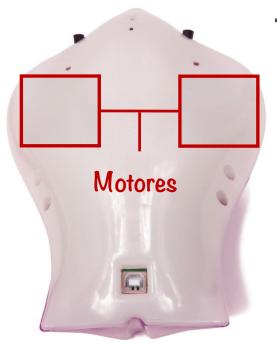
Toma el texto de pasado como parámetro y lo sintetiza en un archivo de sonido para reproducirlo en los altavoces del PC.

La duración indicada permite retrasar la ejecución del programa durante un número de milisegundos después de la emisión de la sintetización.

FreeTTS 1.2.3 - A speech synthesizer written entirely in the JavaTM







Java Finch API



```
myFinch.setWheelVelocities(100, 100);
myFinch.sleep(1000);
```

```
myFinch.setWheelVelocities(100, 100, 1000);
```

```
myFinch.stopWheels();
```

Los valores válidos actuales oscilan entre -255 y 255.

Si timeToHold es positivo, este método bloquea la ejecución del programa durante la cantidad de tiempo especificado por timeToHold y, a continuación, detiene las ruedas una vez transcurrido el tiempo.







Java Finch API

```
double value = myFinch.getTemperature();
boolean hot = myFinch.isTemperature(30);
```

Devuelve la lectura de la temperatura actual obtenida mediante el sensor de temperatura integrado.

El valor devuelto está en grados Celsius.

Para obtener en grados Fahrenheit, multiplique el número por 1,8 y luego agregue 32.









Java Finch API

```
int 11 = myFinch.getLeftLightSensor();
int 12 = myFinch.getRightLightSensor();
int[] light = myFinch.getLightSensors();
```

Los valores válidos oscilan entre 0 y 255, donde los valores más altos indica que el sensor detecta más luz que los valores más bajos.









Java Finch API

```
boolean o = myFinch.isObstacle();
boolean ol = myFinch.isObstacleLeftSide();
boolean or = myFinch.isObstacleRightSide();
boolean[] olr = myFinch.getObstacleSensors();
```







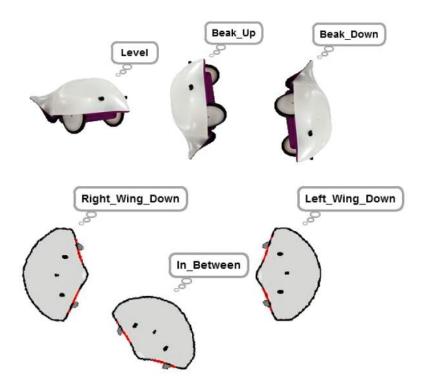
Java Finch API

```
boolean b1 = myFinch.isBeakDown();
boolean b2 = myFinch.isBeakUp();
boolean b3 = myFinch.isFinchLevel();
boolean b4 = myFinch.isFinchUpsideDown();
boolean b5 = myFinch.isLeftWingDown();
boolean b6 = myFinch.isRightWingDown();
boolean b7 = myFinch.isTapped();
boolean b8 = myFinch.isShaken();
```

















iMuchas gracias!

jordi.arino@pue.es @jordiAS2K











¿Alguna pregunta?

www.pue.es/oracle-academy educacion@pue.es











BARCELONA - MADRID www.pue.es

iGracias!

- **#PUEDAY18**
- educacion@pue.es
- 93 206 02 49













...... Networking cisco Academy