

Vitalinux y robótica

¿Se puede hacer robótica con Vitalinux (Linux)? Por supuesto que sí. En las siguientes páginas encontrarás la documentación relativa a la robótica usando Vitalinux.

Descubrirás que con Vitalinux viene ya lo hemos pensado para **facilitar el uso de los diferentes robots** en el aula y para que te puedas centrar en el robot y no en configurar e instalar aplicaciones o controladores.

Si necesitas crear un [punto de acceso para tus robots usando uno de los Vitalinux puedes consultar como hacerlo aquí](#).

- [ROBOTS LICITACIÓN ESCUELA 40](#)
 - [Cuadro resumen](#)
 - [Paquetes de Escuela 4.0 Agrupados](#)
 - [R3 Vincibot](#)
 - [R6 LEGO Spike](#)
 - [R8 Microbit](#)
 - [R9 Echidna](#)
 - [R10 STEAMAKERS ESP32](#)
 - [R12 Robots Makeblock mBot2 Cyberpi PROGRAMA mBlock 5-0 y mLink](#)
- [ROBOTS FUERA DE LA LICITACIÓN ESCUELA 40](#)
 - [PICO BRICKS](#)
 - [Arduino Cloud e Internet de las Cosas \(IoT\)](#)
 - [Arduino con Snap4Arduino](#)
 - [Lego Wedo y equipos BQ](#)
 - [EV3DEV + Open Roberta para LEGO Mindstorms EV3](#)
- [Si no tienes Vitalinux migasfree pero tienes Linux](#)

ROBOTS LICITACIÓN ESCUELA 40

Se utiliza la misma nomenclatura que [la licitación Escuela 4.0 oct25](#)

ROBOTS LICITACIÓN ESCUELA 40

Cuadro resumen

CUADRO RESUMEN ROBOTS LICITACIÓN ESCUELA 40

Se utiliza la misma nomenclatura que [la licitación Escuela 4.0 oct25\(opens new window\)](#)

ID	Robot	Vitalinux Play	Via web	Observaciones
R1	Talebot			
R2	Bluebot			
R3	Vincibot		SI	Necesita modificar reglas de udev
R6	Lego Spike			
R7	Cutebot			Se trata igual que la Micro:bit
R8	Micro:bit	NO	SI	
R9	Echidna	SI	NO	
R10	ESP32 StemMkr	SI (connector)	SI	Necesita connector para funcionar el programa via web
R11	SmartHome Mcr			Se trata igual que la Micro:bit
R12	mBot2	SI (mLink)	SI	Necesita mLink para funcionar el programa via web
R13	Photon			



ROBOTS LICITACIÓN ESCUELA 40

Paquetes de Escuela 4.0

Agrupados

¿Para qué sirve esto?

En Vitalinux se han preparado 4 paquetes que engloban el software necesario para trabajar con los materiales (robots, impresoras 3D...) que llegan a los centros con la licitación de la Escuela 4.0. Se organizan según la [asignación](#) de recursos del programa.

Primaria 1 `vx-dga-l-escuela40-primaria1`

- MatataCode Studio (editor para Vincibot / R3)

Primaria 2 `vx-dga-l-escuela40-primaria2`

- MakeCode (Micro:bit / R8)
- Arduinoblocks / Snap4Arduino (Echidna Black / R9)

Secundaria `vx-dga-l-escuela40-secundaria`

- MakeCode (Micro:bit / R8)
- mBlock (mBot2 CyberPi / R12)
- Arduinoblocks (ESP32 TDR Steam / R10)

Centros CRIE `vx-dga-l-escuela40-crie`

- PrusaSlicer (Impresora 3D / M1)
- Meshlab (Para trabajar con los mesh de los escaneadores 3D)

Nota: Los paquetes se instalan desde el Play de Vitalinux. Cada centro solo necesita informar al coordinador si quiere instalar un paquete para asignarle una instalación automática, o instalar el paquete manualmente que corresponde a su kit asignado.

Toda esta información hace referencia a la licitación de [Escuela 4.0](#)

ROBOTS LICITACIÓN ESCUELA 40

R3 Vincibot

Uso de VinciBot en Vitalinux

Vitalinux incorpora soporte para trabajar con el robot **VinciBot** dentro del marco de **Escuela 4.0**, permitiendo programarlo directamente desde el editor oficial de MatataStudio sin necesidad de instalar software adicional. Esta integración facilita al profesorado la preparación de actividades de robótica educativa de forma sencilla, estable y compatible con el equipamiento de los centros.

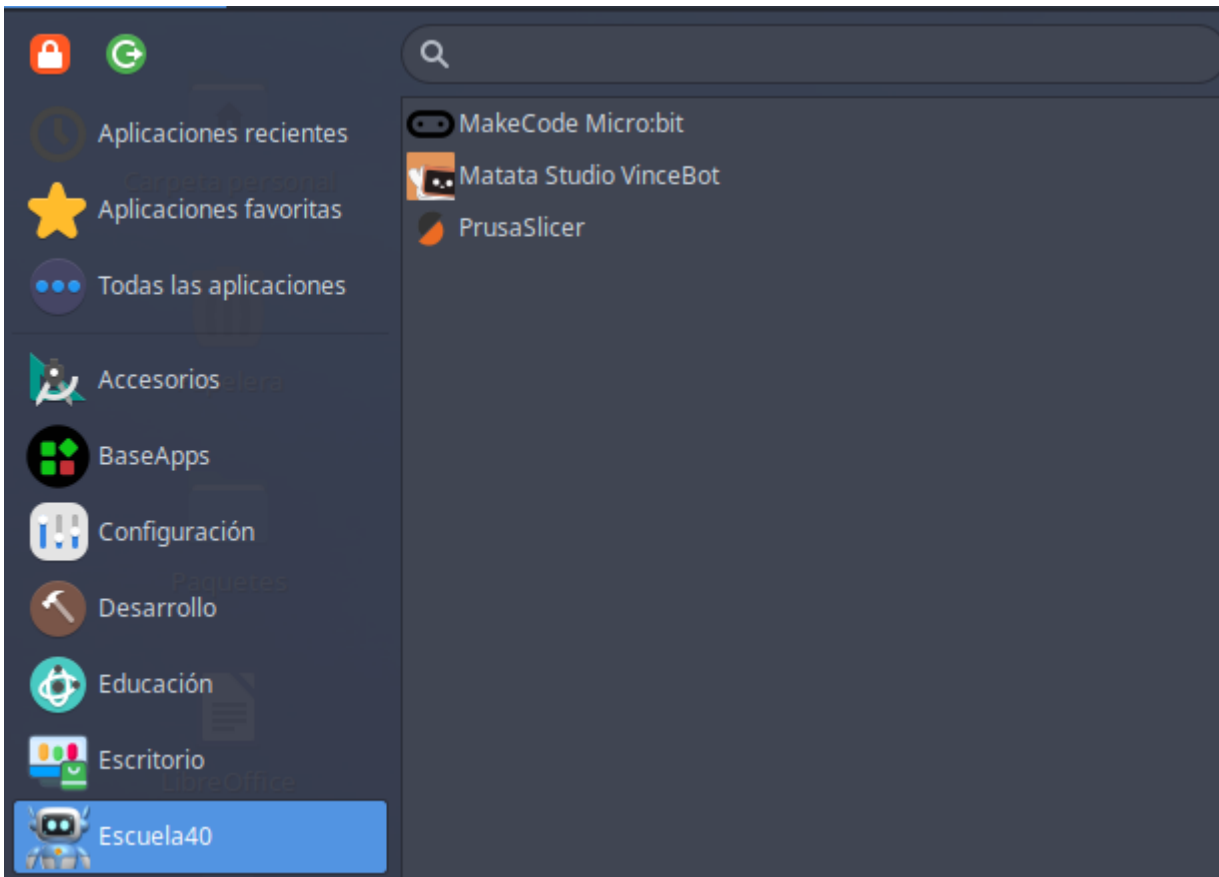
La programación de VinciBot se realiza mediante el editor oficial disponible en:

<https://vinci.matatastudio.com>

Mas información en el siguiente libro de CATEDU [Librería CATEDU - Vincibot](#)

El editor funciona directamente en el navegador, no requiere registro y es totalmente compatible con Vitalinux.

En Vitalinux, el acceso se encuentra dentro del menú **Escuela 4.0**, donde se incluye un enlace directo al editor.



Para que el profesorado pueda transferir programas al robot, es imprescindible:

- Que **VinciBot esté encendido**.
- Que esté **conectado al equipo mediante cable USB**.
- Que Vitalinux haya aplicado correctamente los **permisos udev** necesarios para acceder al dispositivo.

Si tienes migasfree, Vitalinux configura automáticamente estos permisos al instalar el editor, de modo que el robot puede comunicarse sin necesidad de acciones adicionales por parte del profesorado.

Métodos de carga del programa en el robot

El editor de VinciBot permite cargar programas directamente al robot siempre que esté encendido y conectado.

Carga directa desde el editor

Con el robot conectado por USB y encendido, el editor permite enviar el programa directamente.



Pasos generales:

- Encender VinciBot.
- Conectarlo al equipo mediante USB.
- Abrir el editor en <https://vinci.matatastudio.com>
- Crear o abrir un proyecto.
- Pulsar **Upload** (o el botón equivalente según idioma).
- Esperar a que el editor detecte el robot y complete la transferencia.

[IMAGENES A AÑADIR]

Actualización del firmware desde el editor

VinciBot permite actualizar su firmware directamente desde el editor oficial. Esta función es especialmente útil en entornos educativos, ya que garantiza que todos los robots del aula estén sincronizados y funcionando correctamente.

Cómo actualizar el firmware

1. Encender el robot.
2. Conectarlo al equipo mediante USB.
3. Acceder al editor: <https://vinci.matatastudio.com>
4. Abrir el menú de configuración del dispositivo.
5. Seleccionar **Actualizar firmware**.
6. Esperar a que el proceso finalice sin desconectar el robot.

El editor se encarga de todo el proceso, incluyendo la verificación de la versión instalada.

[IMAGENES A AÑADIR]

Permisos **udev** en Vitalinux

Para que el robot pueda comunicarse con el sistema, Vitalinux aplica automáticamente las reglas **udev** necesarias. Estas reglas permiten:

- Detectar el robot al conectarlo.
- Autorizar la comunicación USB sin privilegios de administrador.
- Garantizar que la carga de programas funcione en todos los equipos del centro.

Este proceso se realiza de forma transparente. En caso de que un robot no sea detectado, basta con:

- Desconectar y volver a conectar el cable USB.



- Comprobar que el robot está encendido.
- Reiniciar el navegador si fuera necesario.

ROBOTS LICITACIÓN ESCUELA 40

R6 LEGO Spike

OPCIÓN CONEXIÓN POR CABLE USB

Para programar este LEGO SPIKE entramos en la web <https://spike.legoeducation.com/> ([opens new window](#)) y nos pide elegir Basic o Prime, dependiendo del paquete a usar, los dos usan el mismo HUB

Y una vez dentro del reto en la ventana de programación pide conectarse al HUB.

Sub

Seguimos las instrucciones utilizando la conexión **por cable**:

Sub

Y al dar a **open** nos pide permiso para conectarse al puerto:

Sub

Y ya podemos jugar !!! <https://youtu.be/1yJ2oHeeq28>([opens new window](#))

OPCIÓN POR BLUETOOTH

Puedes solicitar en <https://soporte.vitalinux.educa.aragon.es/> un lanzador de "chrome" o "chromium" especial que abra el navegador con estas dos características habilitadas:

- *Web Bluetooth*
- *Experimental Web Platform features*

Puedes indicar en la solicitud qué nombre quieres poner al lanzador incluso puedes añadirnos un icono de un robot o similar para el lanzador para que los alumnos lo tengan fácilmente identificable y en qué aula (indica la etiqueta, por ejem SALASINF)

Si no nos indicas un icono le pondremos este:



Y de página de inicio iría a <https://spike.legoeducation.com/>

Con ese lanzador ya conecta de maravilla

ROBOTS LICITACIÓN ESCUELA 40

R8 Microbit

Uso de Micro:bit en Vitalinux

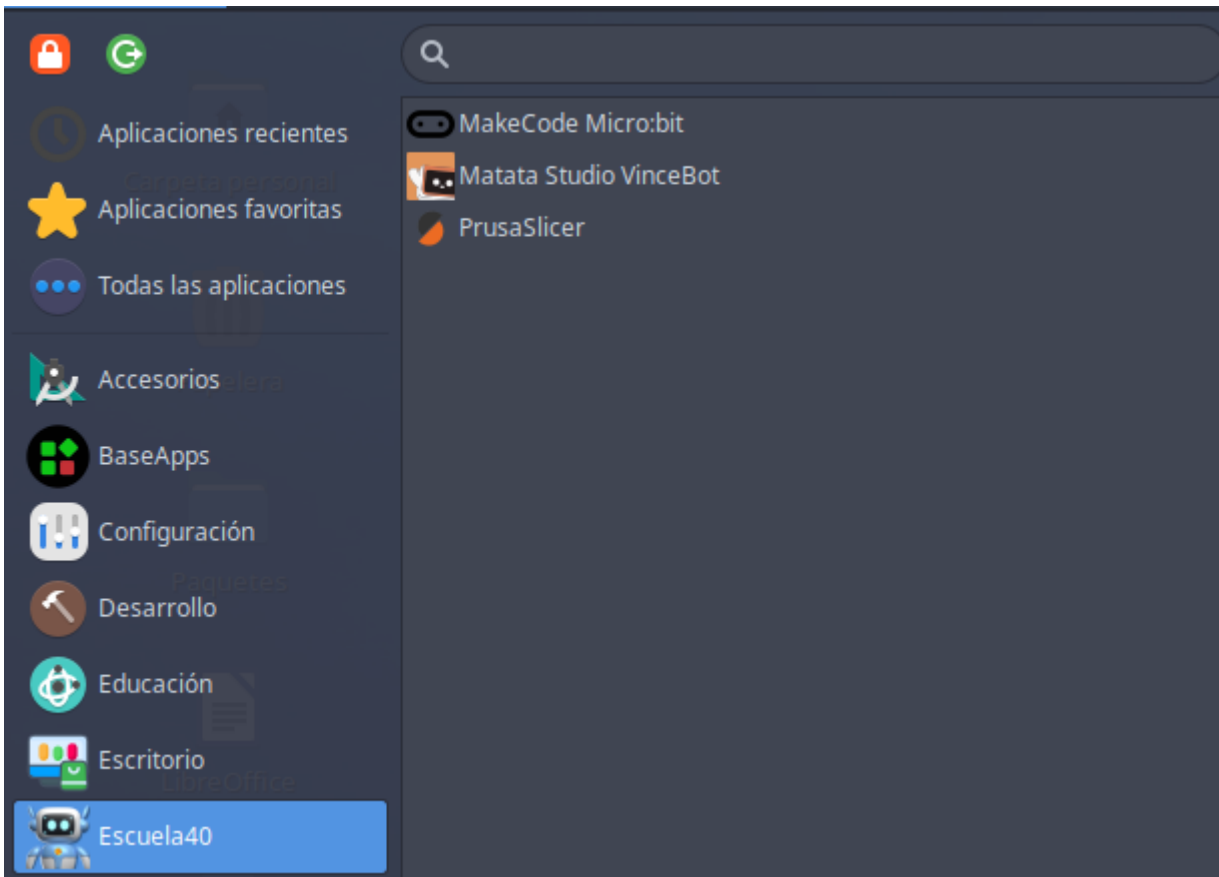
Vitalinux incorpora una integración específica para trabajar con placas **BBC Micro:bit** dentro del apartado **Escuela 4.0**, permitiendo programarlas sin instalar software adicional. Esta integración facilita al profesorado la preparación de actividades de programación, robótica y pensamiento computacional de manera sencilla y estable.

La programación de las placas Micro:bit se realiza mediante el editor oficial **MakeCode**, accesible desde:

<https://makecode.microbit.org>

El editor funciona directamente en el navegador, no requiere registro y es totalmente compatible con Vitalinux.

En Vitalinux, el acceso se encuentra dentro del menú **Escuela 4.0**, donde se incluye un enlace directo al editor.



Métodos de carga del programa en la placa

Carga directa desde el editor

Con la placa conectada por USB, MakeCode permite enviar el programa directamente a la Micro:bit desde el propio navegador.

Pasos generales:

- Conectar la placa al equipo mediante USB.
- Crear o abrir un proyecto en MakeCode.
- Pulsar **Descargar** y seleccionar **cargar directamente en la placa**.
- Esperar a que el editor detecte la Micro:bit y complete la transferencia.

Microsoft | micro:bit

Flashing Heart - Paso 2 de 6

Inicio sesión

Click on the **Basic** category in the Toolbox. Drag the **show Leds** block into the **forever** block. Then in the **show Leds** block, click on the squares to draw a heart design.

Caja de herramientas

Buscar...

Básico

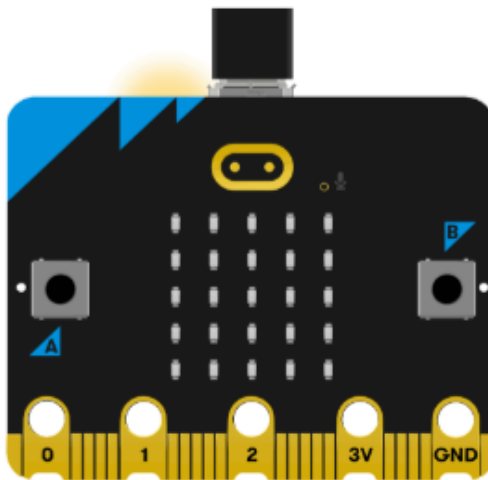
para siempre

mostrar LEDs

Descargar

Siguiente

1. Conecte su micro:bit a su computador



Siguiente

2. Empareja tu micro:bit con tu navegador



Presiona el botón de emparejar abajo.

Aparecerá una ventana en la parte superior de tu navegador.

Selecciona el dispositivo micro:bit y haz clic en Conectar.



Descargar como archivo



Emparejar

Pasos correspondientes a la carga directa desde el editor

Descarga del archivo `.hex` y copia manual

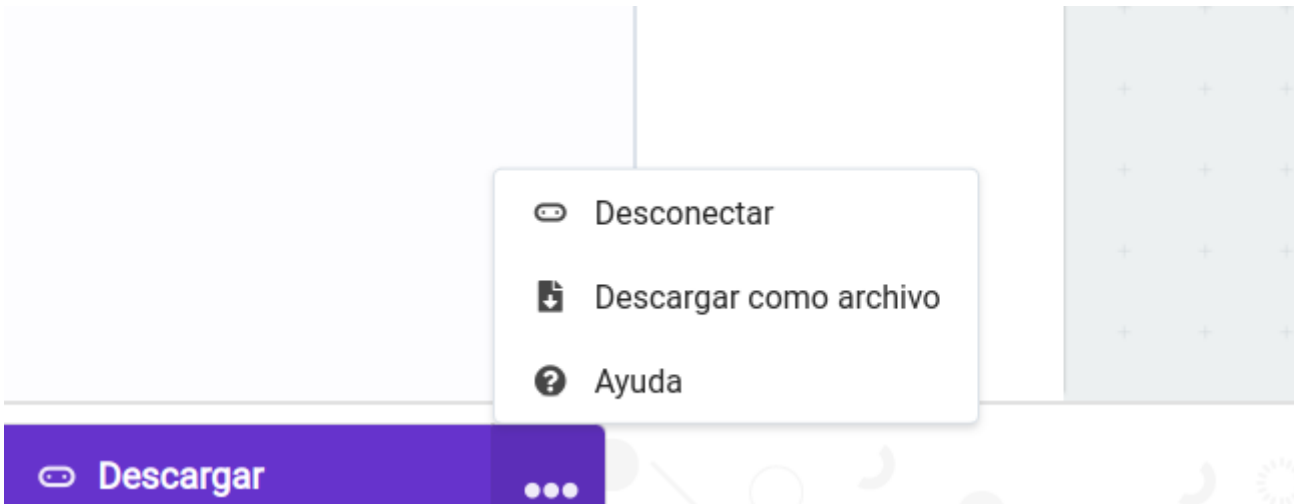
Como alternativa, es posible descargar el archivo compilado y copiarlo manualmente en la placa, que aparece en el sistema como un dispositivo de almacenamiento.

Pasos:

- Pulsar **Descargar** en MakeCode.
- Guardar el archivo `.hex`.
- Copiarlo en la unidad correspondiente a la Micro:bit.

(Imagen del proceso de descarga como archivo)

Este método es útil cuando la carga directa no está disponible o cuando se trabaja con varias placas.



En vez de descargar a la placa, se puede descargar como archivo

Comprobación y actualización del firmware

Para garantizar la compatibilidad con MakeCode y la carga directa, la placa debe tener un firmware **superior a la versión 0249**.

Cómo comprobar la versión

1. Conectar la Micro:bit al equipo.
2. Abrir la unidad que aparece en el sistema.
3. Consultar el archivo **DETAILS.TXT**.
4. Buscar la línea donde aparece la versión del firmware.

En el ejemplo mostrado, la placa tiene la versión **0241**, por lo que requiere actualización.

Sub

(Imagen del archivo DETAILS.TXT con la versión)

Cómo actualizar el firmware

En la página oficial de MakeCode se ofrece un procedimiento sencillo para actualizar la placa si la versión es inferior a la recomendada.

Enlace: <https://makecode.microbit.org>



ROBOTS LICITACIÓN ESCUELA 40

R9 Echidna

1. Instalamos el software desde **VitalinuxPlay**



2. Conectamos la placa Echidna en el puerto USB
3. Ejecutamos **EchidnaML** (lo tienes en la carpeta "Educación"), el programa detecta la placa y funciona perfectamente

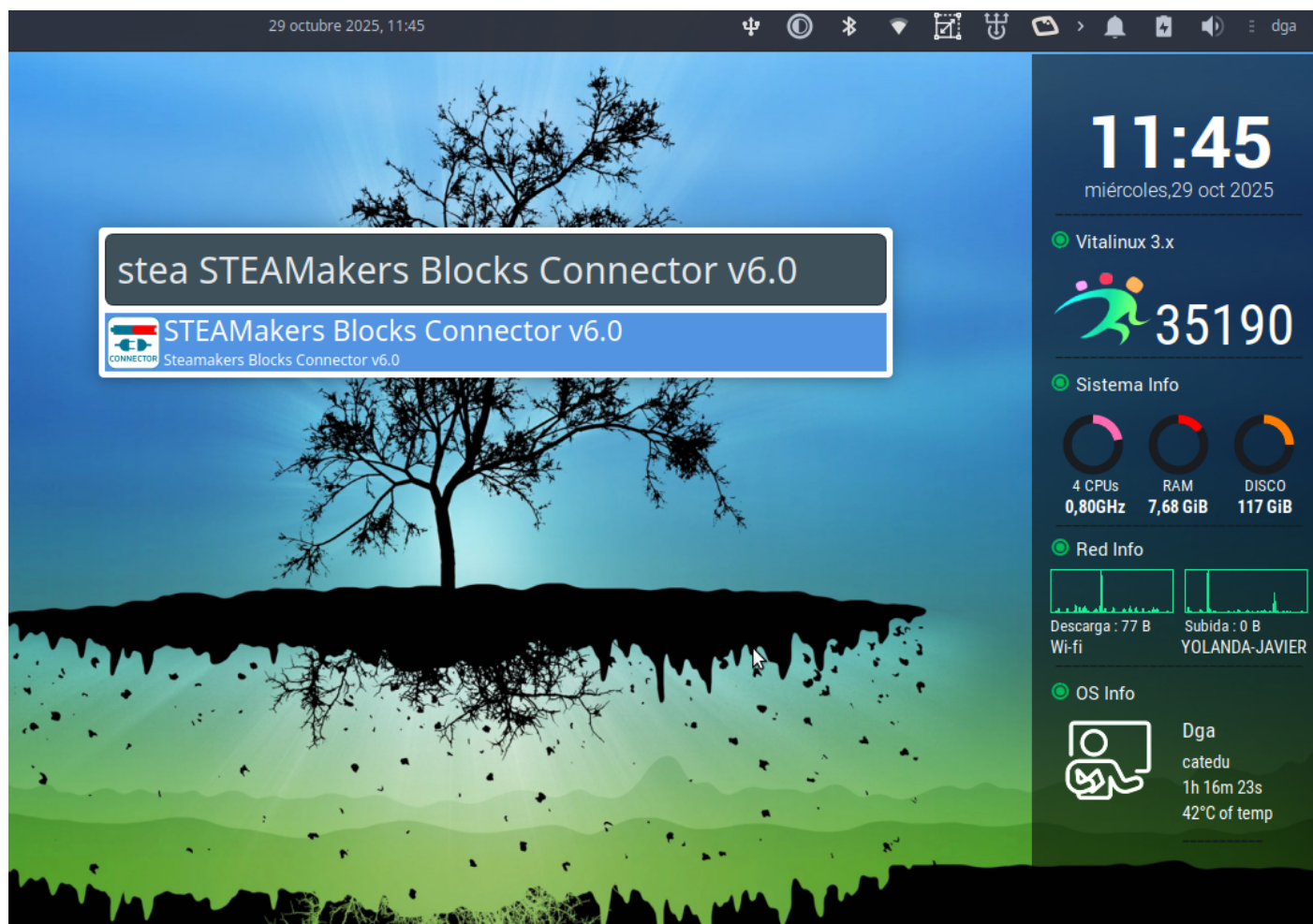
ROBOTS LICITACIÓN ESCUELA 40

R10 STEAMAKERS ESP32

STEAMAKERS BLOCKS es una aplicación online que lo apoya Innovadidactic y que nos permite conectarnos con diferentes placas:

- Arduino
- ESP32
- Estas mismas placas con el escudo ImaginaSTEAM ...

Necesita la aplicación **Connector**. Podemos instalarlo desde **Vitalinux Play**. Sirve para comunicar la aplicación online con la placa que conectamos en nuestro equipo.





Se nos abrirá automáticamente el navegador con la página abierta de la aplicación online

<https://www.steamakersblocks.com>(opens new window)

ArduinoBlocks register

Una vez dentro de la aplicación y registrados, solo habremos de seleccionar el puerto USB en el que tengamos conectada nuestra placa y a funcionar, creamos el programa en bloques y **Subir**.

ArduinoBlocks demo

ROBOTS LICITACIÓN ESCUELA 40

R12 Robots Makeblock mBot2 Cyberpi PROGRAMA mBlock 5- 0 y mLink

mBlock 5-0 y mLink puede utilizarse aparte de para programar mBot2, para muchas otras placas:

- Cyberpi
- mBot1
- Echidna [mejor utilizar EchidnaML](#)
- Arduino UNO....

En SO Linux como Vitalinux es necesario :

1. Instalar mLink Este programa está disponible en **Vitalinux Play**. Es un programa que tiene que estar funcionando para que el navegador se comuniqué con la placa
2. Utilizar **la versión online** disponible en [https://ide.mblock.cc/\(opens new window\)](https://ide.mblock.cc/(opens new window))

Luego lo primero instala **mLink** desde **Vitalinux Play** :

Sube

Una vez instalado, seguir estos sencillos pasos:

1. Conecta la placa al ordenador
2. Arranca el **mLink** instalado previamente y **no lo cierres, minimízalo**.
3. Arranca un navegador compatible con Chrome y accede a la [web https://ide.mblock.cc](https://ide.mblock.cc) [\(opens new window\)](#). Puede ser Chromium también.

En esta página web <https://ide.mblock.cc> [\(opens new window\)](#) le damos a conectar y elegir el puerto USB (suele ser el último, ver imagen)

Sube

Una vez conectado, ya proceder como normalmente.



Notas: Al ejecutar mLink, saldrá una ventana de comandos **QUE NO TENEMOS QUE CERRAR** dejarla minimizada. **OJO PRIMERO MLINK Y LUEGO MBLOCK**

```
Start mlink: Running...
```

```
Version: 1.2.0
```

Copied!

**** Notas mBot Con pincho Bluetooth ****

Hemos probado inalámbricamente con el pincho Bluetooth (no lo incorpora la licitación Escuela 40), y trabajar **en vivo** como **cargar** y funciona correctamente.

lsub

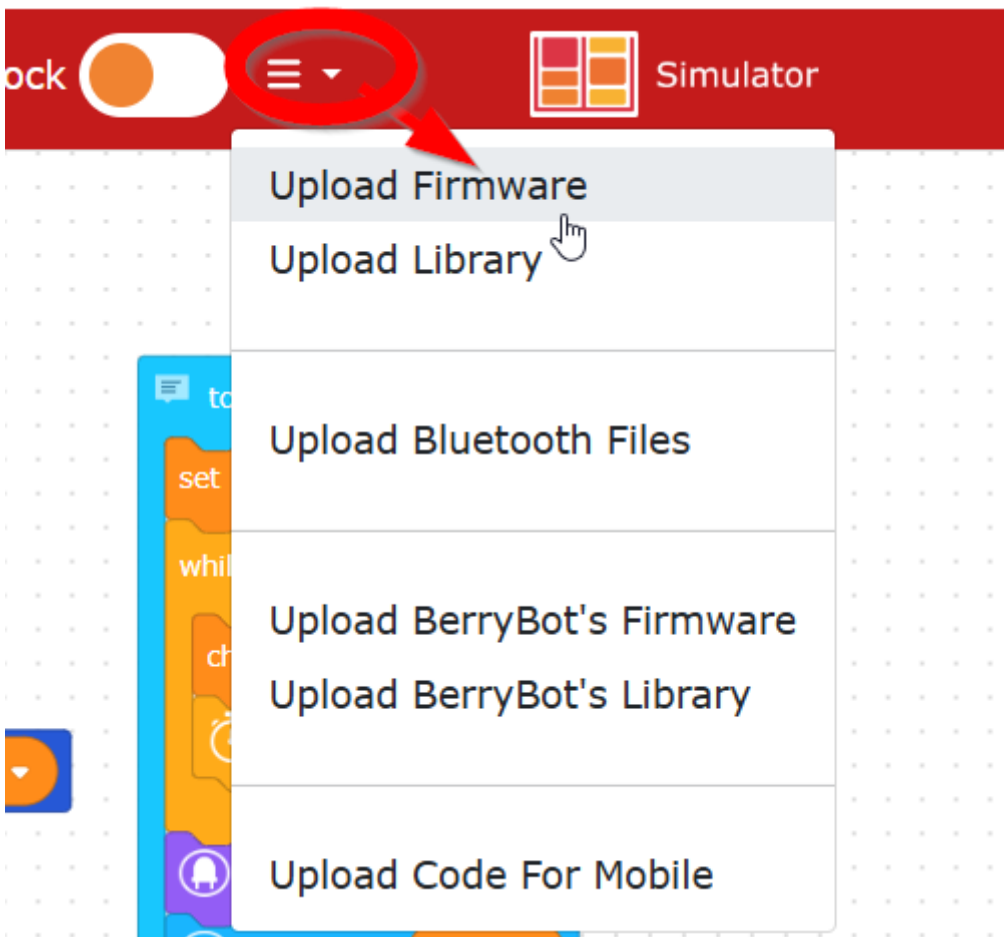
ROBOTS FUERA DE LA LICITACIÓN ESCUELA 40

ROBOTS FUERA DE LA LICITACIÓN ESCUELA 40

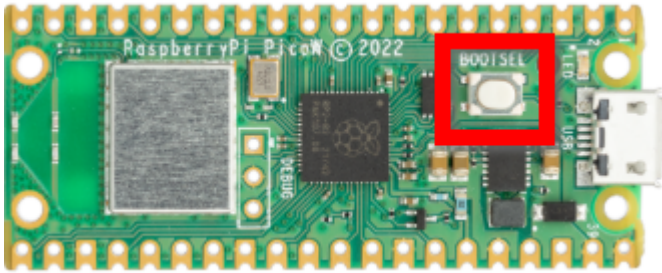
PICO BRICKS



1. Entramos en la página <http://rbt.ist/ide>(opens new window)
2. Descargamos el Firmware



3. Ponemos PicoBricks en modo Boodloader
 - Apretamos el botón BOOTSEL
 - **MIENTRAS LO TENEMOS APRETADO** lo conectamos al USB
 - Aparecerá una nueva unidad, ya podemos soltar el botón Bootsel



4.- El fichero descargado en el paso 2 lo grabamos en la nueva unidad

Sub

5. Pasado unos minutos, desconecta de USB y vuelve a conectar y **ya puedes programar**, al darle RUN te pedirá que te conectes con Pico bricks, elegir el puerto USB correspondiente (ver imagen)

Sub

ROBOTS FUERA DE LA LICITACIÓN ESCUELA 40

Arduino Cloud e Internet de las Cosas (IoT)

Arduino además de hardware, y el software Arduino IDE para PC, también pone a disposición de las personas que quieran registrarse una Aplicación web con los siguientes servicios:

- Arduino Cloud
- IoT Cloud: para trabajar IoT con placas de arduino.
- Web Editor: igual que el Arduino IDE pero en versión web
- Manager for Linux

Arduino World

Desde vitalinux siempre recomendamos usar la versión de Escritorio del Arduino IDE para evitar problemas de conectividad y para evitar registrar al alumnado. No obstante, por si alguien quiere trabajar en Arduino Cloud (por ejemplo en los proyectos de IoT es necesario utilizar la aplicación IoT Cloud, que solo está disponible desde la aplicación web) es necesario previamente descargar un software en el equipo llamado **Arduino Create Agent**.

Hemos habilitado en Vitalinux Play una forma sencilla de instalarlo en un equipo individual. Se instala de igual forma que cualquier otro programa desde el Vitalinux Play, en la flecha de arriba a la derecha.

Arduino Agent Play

Una vez instalado, hay que lanzar la aplicación. Se instala en el menú Sonido y Video, si bien es mucho más fácil usar Albert y empezar a escribir el nombre. Arduino Agent Run

En ese momento nos aparecerá un icono abajo a la derecha, junto al reloj. Desde allí podremos activarlo y pausarlo para cuando lo necesitemos en los proyectos.

Arduino Agent app

Como siempre, si queréis una instalación masiva de forma remota en varios equipos de vuestro centro, podéis canalizarla a través de soporte.

ROBOTS FUERA DE LA LICITACIÓN ESCUELA 40

Arduino con Snap4Arduino

Snap4Arduino es un programa libre, heredero del anterior Scratch For Arduino (S4A) que sirve para programar placas con un lenguaje de bloques de forma sencilla. Posee versión online y offline. La versión offline se puede descargar desde Vitalinux Play. Para poder controlar la placa simplemente tendremos que cargar previamente en ella el **StandardFirmata**, un firmware disponible en los Ejemplos de la aplicación Arduino IDE. Por lo tanto será necesario en nuestro equipo también habernos descargado el Arduino IDE.

arduino IDE

Una vez cargado ese firmware en la placa, podremos usar Snap4Arduino para controlarla, y lo interesante es que dispone de la opción de ver el funcionamiento en vivo, sin necesidad de subir el programa a la placa.

arduino IDE

ROBOTS FUERA DE LA LICITACIÓN ESCUELA 40

Lego Wedo y equipos BQ

LEGO WeDo

Lego WeDo 2.0 (bluetooth) no tiene el software desarrollado para funcionar en Linux, por lo que **no podremos usar el software específico desarrollado por Lego para funcionar con sus actividades**. Sin embargo **puede funcionar con Scratch (web u offline) bajo Linux** usando un conector desarrollado por la comunidad: Scratch Link. Se puede instalar en Vitalinux mediante Vitalinux Play o de forma centralizada.

Scratch Link permite conectar hardware para interactuar con los proyectos de Scratch. [Web e instrucciones del proyecto para Linux \(opens new window\)](#) para uso por ejemplo con bluetooth o la placa microbit.

Lego WeDo 1.0 ya no es funcional debido a que la solución que había se basaba en usar Scratch2Offline (no instalable en Vitalinux 3) y en una app para chrome/chromium(s2bot 4 scrat) que igualmente ha dejado de soportado para Chrome/Chromium y que solo funcionaría con Scratch 2.0.

Equipos BQ

Hay que tener en cuenta que BQ ha dejado de existir como empresa y ya no hay continuidad de sus productos (hardware o software).

BitBloq - Web2Board

La web ha dejado de funcionar y ya no se puede usar bitbloq online. Sólo quedaría usar la versión offline pero solo se puede instalar para vitalinux 2.x y no sabemos si funcionará.

Impresoras 3D Witbox Go! de BQ

BQ proporciona el software Zetup para trabajar con dichas impresoras, pero **NO FUNCIONA en Vitalinux 3.x**, al no tener una versión actualizada del mismo.



Como dicho software se debe descargar e instalar a mano desde la web del fabricante, hemos paquetizado el mismo para poder instalarlo directamente desde Synaptic (o el gestor de paquetes que uses), pero sobre todo para poder instalarlo de forma desatendida en aquellos equipos que nos indiqueis. El paquete se llama: **vx-dga-l-zetup-bq** y lo puedes encontrar en Vitalinux Play

ROBOTS FUERA DE LA LICITACIÓN ESCUELA 40

EV3DEV + Open Roberta para LEGO Mindstorms EV3

Este manual puede tener alguna inconsistencia. Se ha probado en pocos bloques EV3 por lo que no se puede garantizar que funcionen con todos.

¿Para qué sirve este manual?

El software oficial de LEGO para EV3 dejará de tener soporte el 31 de julio de 2026. Este manual explica cómo instalar **EV3DEV**, un sistema alternativo basado en Linux, y cómo conectarlo a **Open Roberta Lab**, el editor de programación por bloques en web, para seguir usando los robots en clase sin interrupciones.

Lo que necesitas antes de empezar

- El bloque EV3 (el "ladrillo")
- Una tarjeta **microSD** de al menos 4 GB (recomendable una MicroSD A1 o superior, máximo 32 GB)
- Un lector de tarjetas microSD (muchos portátiles lo traen incorporado; si no, un adaptador USB vale)
- Un ordenador con conexión a internet
- Un cable USB mini-B (el mismo que se usa para cargar el EV3)

1: Descargar las herramientas

1.1 Descargar Balena Etcher

Etcher es el programa que usaremos para "grabar" el sistema en la tarjeta SD. Es gratuito y muy sencillo de usar. Existe en el Play de Vitalinux como programa a descargar. Si estas en otro

entorno:

1. Ve a <https://etcher.balena.io>
2. Descarga la versión para tu sistema operativo (Windows, macOS o Linux)
3. Instálalo siguiendo los pasos habituales

Importante: Descarga la versión **Etcher v1.7** o anterior. Las versiones más nuevas tienen incompatibilidades conocidas con las imágenes de EV3DEV.

1.2 Descargar la imagen de EV3DEV

1. Ve a la página oficial de descargas: <https://www.ev3dev.org/downloads>
2. Busca la sección "**LEGO MINDSTORMS EV3**"
3. Haz clic en el enlace de **ev3dev-stretch snapshot images**
4. Descarga el archivo más reciente (tendrá extensión `.img.xz`). No hace falta descomprimirlo.

2. Flashear (grabar) la tarjeta SD

"Flashear" significa grabar el sistema EV3DEV en la tarjeta microSD para que el robot pueda arrancarlo.

1. Introduce la tarjeta microSD en el lector de tu ordenador
2. Abre **Balena Etcher**
3. Haz clic en "**Flash from file**" y selecciona el archivo `.img.xz` que descargaste
4. Haz clic en "**Select target**" y elige tu tarjeta microSD

Asegúrate de seleccionar la tarjeta correcta. Etcher borrará todo lo que haya en el dispositivo elegido. Si tienes dudas, desconecta cualquier otro USB o disco externo antes de este paso.

5. Haz clic en "**Flash!**" y espera a que termine (puede tardar varios minutos)
6. Cuando aparezca el mensaje de éxito, extrae la tarjeta del ordenador con seguridad

3. Arrancar el EV3 con EV3DEV

1. Con el EV3 **apagado**, introduce la tarjeta microSD en la ranura que hay en la parte inferior o lateral del bloque.
2. Enciende el EV3 pulsando el botón central
3. Verás aparecer en la pantalla el texto "**ev3dev**" durante el arranque
4. El primer arranque tarda un poco más de lo habitual (es normal). Cuando los LEDs se pongan en verde y aparezca el menú principal, el sistema está listo.

Las credenciales por defecto son: usuario `robot`, contraseña `maker`. Las necesitarás en el siguiente paso.

4. Conectar el EV3 a la red

Para comunicarse con Open Roberta, el EV3 necesita estar conectado a internet. Hay dos opciones:

Opción A — Cable USB (más sencilla, pero no soportada por todos los bloques EV3):

1. Conecta el cable USB entre el ordenador y el puerto USB mini-B del EV3
2. En la pantalla del EV3, ve a **Wireless and Networks** → **USB** → **Connect**
3. El ordenador debería detectar automáticamente el dispositivo (en Windows puede pedir instalar un controlador la primera vez; acéptalo)

Opción B — WiFi:

Necesitas un adaptador WiFi USB compatible con EV3DEV (La mayoría de antenas WIFI USB estan soportadas). Conéctalo al puerto USB del EV3 y configura la red desde **Wireless and Networks** → **Wi-Fi**.

5. Activar el servicio Open Roberta en el EV3

Este paso solo es necesario **la primera vez**. El servicio de Open Roberta no viene activado por defecto para ahorrar memoria.



1. Conéctate al EV3 por SSH desde un terminal del ordenador, si lo conectaste por WIFI, en el bloque puedes consultar la IP que tiene asignada:

```
ssh robot@ev3dev.local  
ssh robot@<IP-DEL-BLOQUE>
```

La contraseña es `maker`.

2. Una vez dentro, ejecuta estos dos comandos (uno tras otro):

```
sudo systemctl unmask openrobertalab  
sudo systemctl start openrobertalab.service
```

3. A partir de ahora el servicio de open roberta arrancará automáticamente cada vez que enciendas el EV3. No hará falta repetir este paso.

6. Conectar con Open Roberta Lab

6.1 Obtener el código de emparejamiento en el EV3

1. En el menú del EV3, navega hasta "**Open Roberta Lab**" usando los botones del bloque
2. Selecciona "**Connect to public server**"
3. El EV3 mostrará un **código de 4 cifras** en pantalla. Apúntalo, lo necesitarás enseguida.

6.2 Conectar desde el navegador

1. En el ordenador, abre el navegador y ve a <https://lab.open-roberta.org>
2. Haz clic en el botón "**Conectar**" (arriba a la derecha)
3. Elige el robot "**EV3dev**" en la lista
4. Introduce el **código de 4 cifras** que aparece en la pantalla del EV3
5. Si la conexión es correcta, el EV3 emitirá un pitido y en el navegador aparecerá el robot como conectado

7. Programar y ejecutar un programa

1. En Open Roberta Lab, crea tu programa con bloques (el entorno es muy similar al software original de LEGO)
2. Cuando quieras ejecutarlo, haz clic en el botón **"Ejecutar"** (el triángulo de play)
3. El código se envía al EV3 automáticamente y el robot lo ejecuta

Para detener un programa con bucle infinito: pulsa simultáneamente los botones **Intro** + **Abajo** en el EV3. Si eso no funciona, mantén pulsado el botón **Atrás** durante 1 segundo.

Resumen del flujo de trabajo en clase

Encender EV3 con SD ev3dev → Conectar USB o WiFi →
Seleccionar "Open Roberta Lab" en menú EV3 → Abrir lab.open-roberta.org →
Introducir código → Programar con bloques → Ejecutar

Solución de problemas frecuentes

Problema	Posible solución
El EV3 no arranca con ev3dev	Comprobar que la SD está bien insertada. Repetir el flasheo.
El ordenador no reconoce el EV3 por USB	Reiniciar la conexión en el menú del EV3. En Windows, revisar que se instalan los controladores.
No aparece la opción "Open Roberta Lab" en el menú, o indica que no esta disponible.	Repetir el paso 5 (activar el servicio por SSH).
Open Roberta no acepta el código	El código caduca en pocos minutos. Volver al menú del EV3 y generar uno nuevo.
El programa no llega al robot	Comprobar que el EV3 aparece como "conectado" en la web antes de ejecutar.

Referencias y recursos

- Página oficial de EV3DEV: <https://www.ev3dev.org>
- Open Roberta Lab: <https://lab.open-roberta.org>
- Balena Etcher: <https://etcher.balena.io>

Manual elaborado por el equipo de soporte Vitalinux · Mayo 2026

Si no tienes Vitalinux migasfree pero tienes Linux

Puedes saltarte éste paso preliminar si usas Vitalinux, migasfree lo hará por ti...

En Lego, mbot, cyberpi, ESP32, Arduino... la comunicación con el mismo se hace a través de un dispositivo de interfaz humana (también llamado HID). Por ejemplo en el caso de Lego WeDo con un cable usb y en el caso de Cyberpi con un emisor inalámbrico (también usb).

Podemos ver los usb conectados con:

```
lsusb
```

Copied!

En la siguiente captura podemos observar la salida de los comandos indicados antes y después de insertar el cable del robot LEGO. Podemos ver que su IdVendor es 0694 y que su idProduct es 0003.

lsusb

Para poder leer y escribir en dicho dispositivo (mandar órdenes al robot o leer datos del mismo), Linux crea un archivo de tipo carácter en /dev, pero podemos observar que sólo tiene permisos root: Vemos dos dispositivos, uno el teclado/ratón y otro que corresponde a la conexión con el robot

centre|thumb|Vemos dos dispositivos, uno el teclado/ratón y otro que corresponde a la conexión con el robot

Para ello, simplemente tenemos que indicarle a udev que nos cree dicho dispositivo con los permisos necesarios:

Editar un archivo nuevo:

```
sudo nano /lib/udev/rules.d/99-hidraw.rules
```

Copied!

Y en ese fichero ponemos la línea

```
ATTRS{idVendor}=="0694", ATTRS{idProduct}=="0003", SUBSYSTEMS=="usb", ACTION=="add",  
MODE="0666", GROUP="plugdev"
```

Copied!

Así sucesivamente con los diferentes robots que queramos conectar.

Nota: Muchos robots de placa electrónica, como el ESP32, Cyberpi, Arduino, etc.. en el listado lsub no aparece el nombre como tal, sino el chip de interface CH340 o UART que traduce el protocolo USB al TTL de la placa. La manera de localizar quien es quien es ejecutar lsub antes y después de conectar el robot

Un fichero que contiene todos, y que iremos actualizando sería este para 99-hidraw.rules :

=====

```
# Solucion generica: para todos los dispositivos de interfaz humana\  
# KERNEL=="hidraw*", SUBSYSTEM=="hidraw", MODE="0664", GROUP="plugdev"\  
# Lego Wedo\  
ATTRS{idVendor}=="0694", ATTRS{idProduct}=="0003", SUBSYSTEMS=="usb", ACTION=="add",  
MODE="0666", GROUP="plugdev"\  
# mBot\  
ATTRS{idVendor}=="0416", ATTRS{idProduct}=="ffff", SUBSYSTEMS=="usb", ACTION=="add",  
MODE="0666", GROUP="plugdev"\  
# Cyberpi mBot2\  
ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="7523", SUBSYSTEMS=="usb", ACTION=="add",  
MODE="0666", GROUP="plugdev"\  
# Lego Spike  
ATTRS{idVendor}=="0694", ATTRS{idProduct}=="0009", SUBSYSTEMS=="usb", ACTION=="add",  
MODE="0666", GROUP="plugdev"\  
# mbot pincho 2.4 UART Windbond  
ATTRS{idVendor}=="0416", ATTRS{idProduct}=="ffff", SUBSYSTEMS=="usb", ACTION=="add",  
MODE="0666", GROUP="plugdev"\  
# microbit  
ATTRS{idVendor}=="0d28", ATTRS{idProduct}=="0204", SUBSYSTEMS=="usb", ACTION=="add",  
MODE="0666", GROUP="plugdev"
```

Copied!



=====

Por último quedaría reiniciar el servicio udev

```
sudo service udev restart
```

Copied!

Si antes de realizar el proceso estaba conectado el robot, tendremos que desconectarlo y volverlo a conectar para que los cambios surtan efecto, quedando algo parecido a esto:

lsub