

ROBOTS LICITACIÓN ESCUELA 40

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

- [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](#)

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			



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](#)

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.

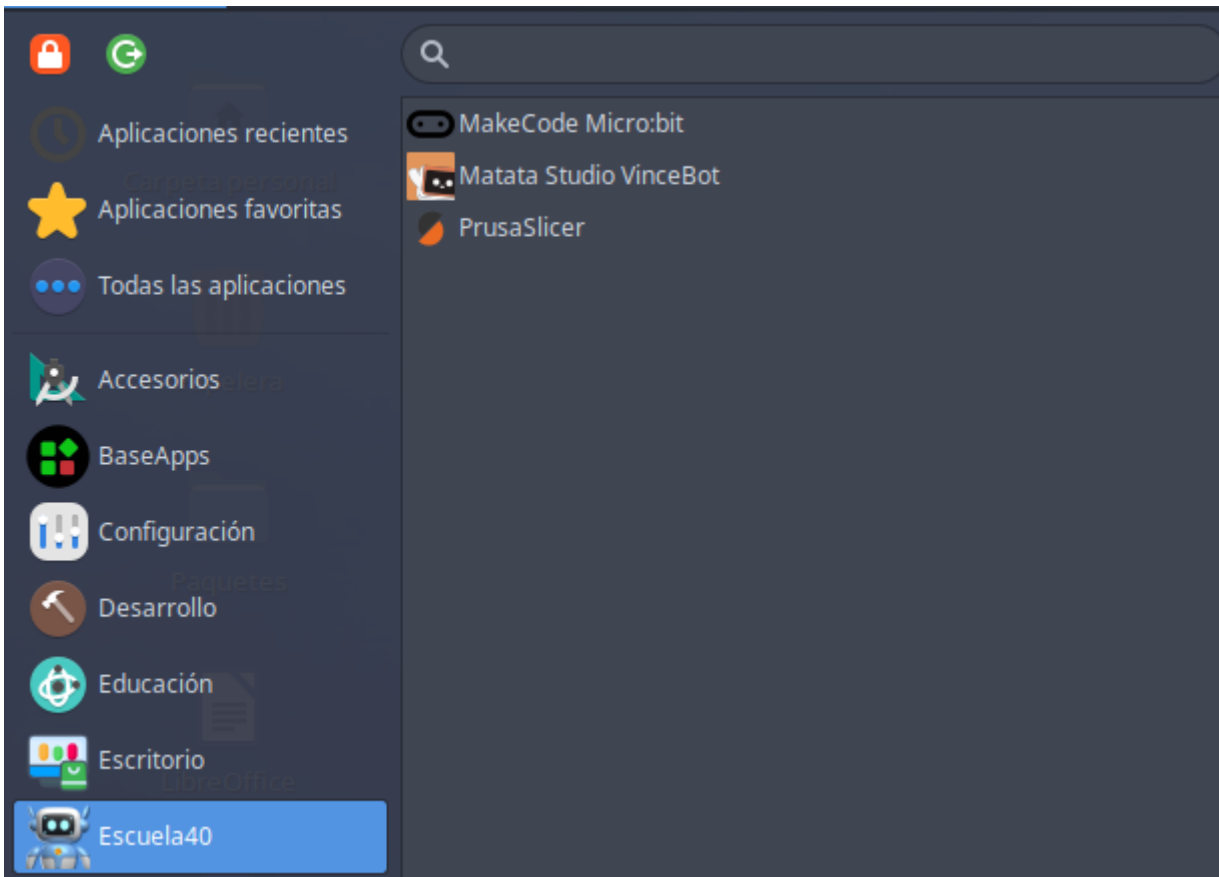
La programación de VinciBot se realiza mediante el editor oficial disponible en:

<https://vinci.matatastudio.com>

Más 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.

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

R8 Microbit

Uso de Micro:bit en Vitalinux

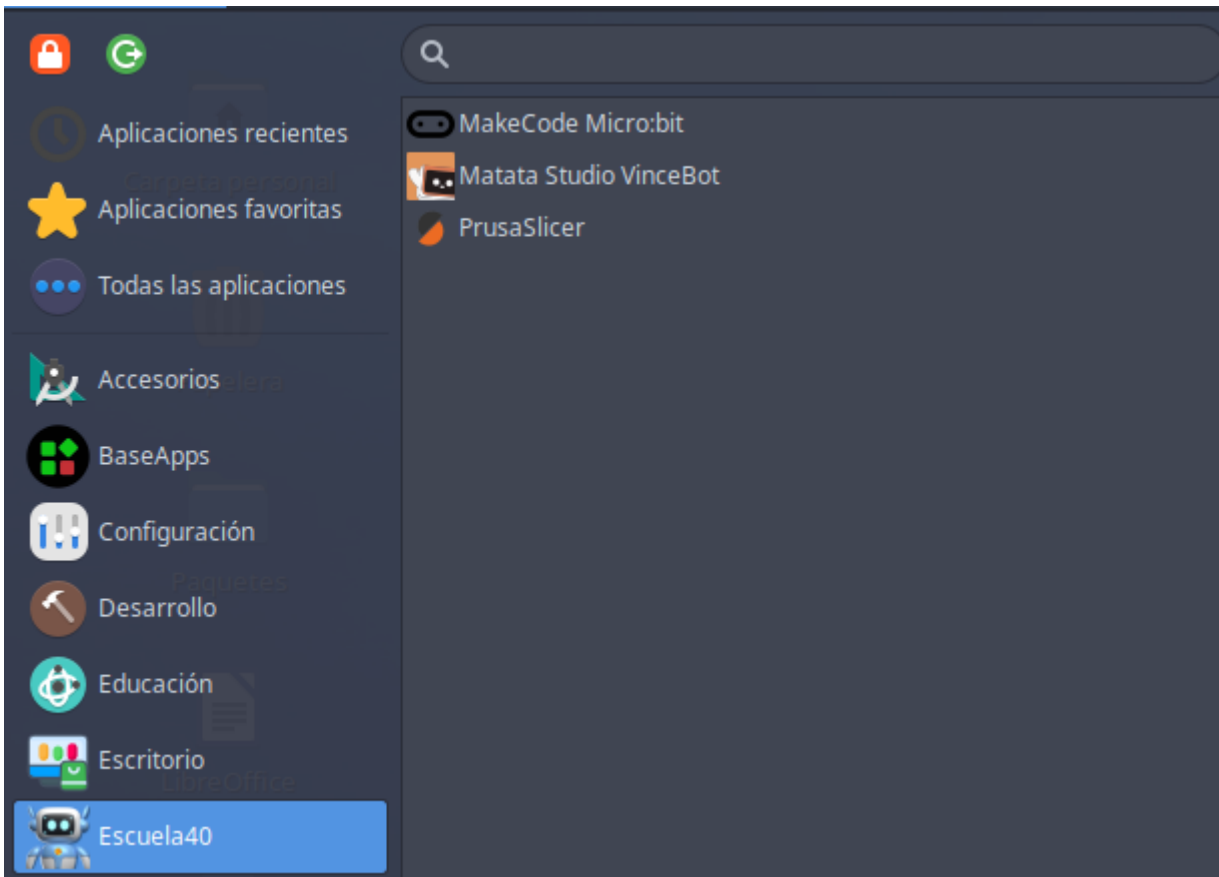
Vitalinux incorpora una integración específica para trabajar con placas **BBC Micro:bit** dentro del apartado **Escuela 4.0**, permitiendo programarlas sin instalar software adicional. Esta integración facilita al profesorado la preparación de actividades de programación, robótica y pensamiento computacional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



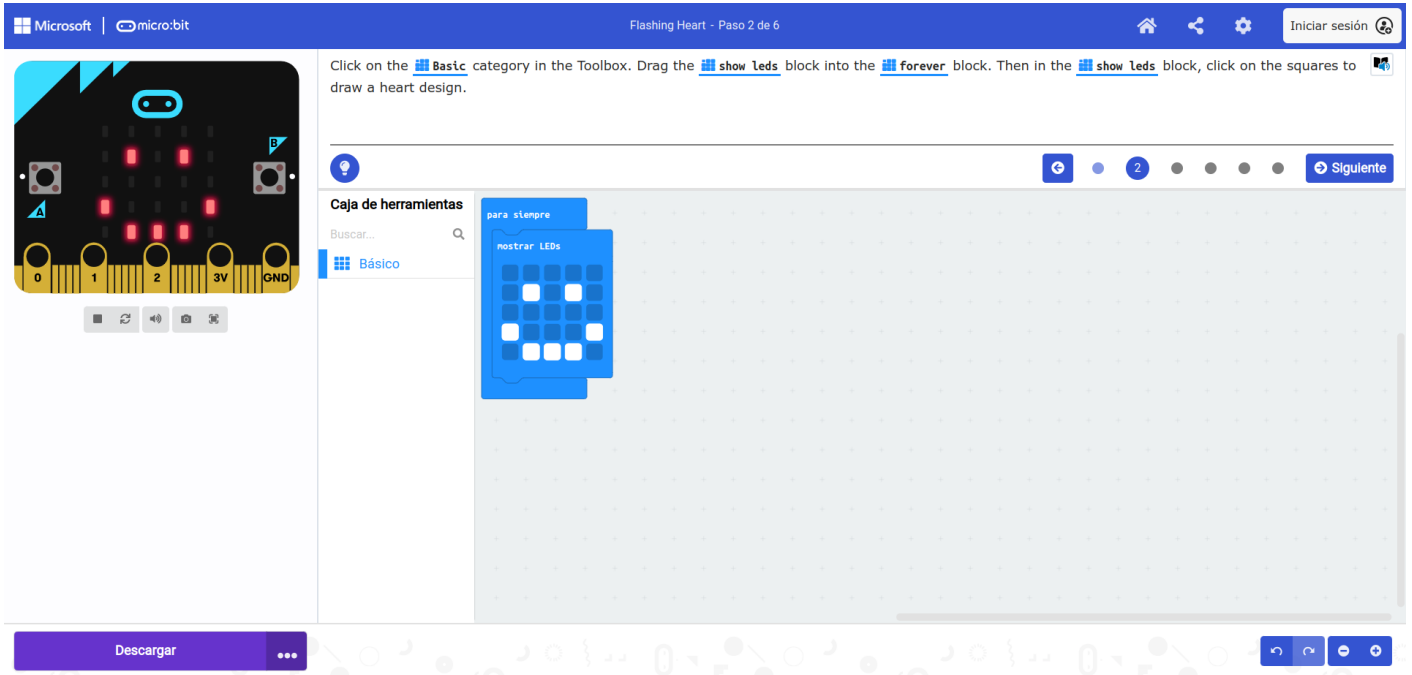
Métodos de carga del programa en la placa

Carga directa desde el editor

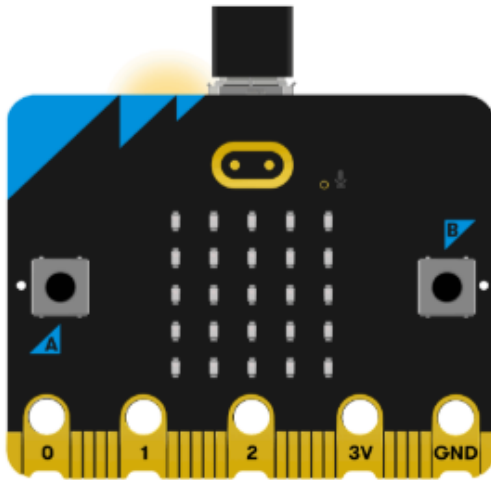
Con la placa conectada por USB, MakeCode permite enviar el programa directamente a la Micro:bit desde el propio navegador.


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.



1. Conecte su micro:bit a su computador



 **Siguiente**

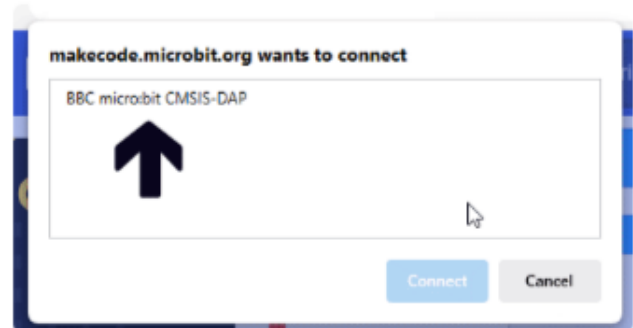
2. Empareja tu micro:bit con tu navegador



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

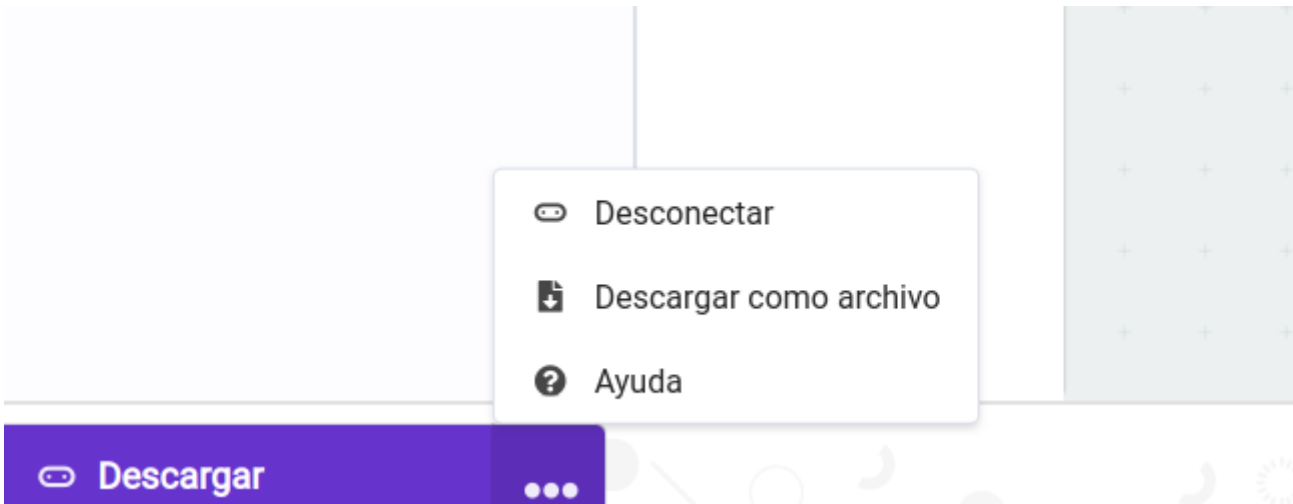
Como alternativa, es posible descargar el archivo compilado y copiarlo manualmente en la placa, que aparece en el sistema como un dispositivo de almacenamiento.

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>



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

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\(opens new window\)](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

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 comunique con la placa
2. Utilizar **la versión online** disponible en <https://ide.mblock.cc/>[\(opens new window\)](#)

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

lsub

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)

lsub

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