

Rover Marciano Con Arduinoblocks e Internet de las cosas IoT

Se trabaja con NodeMCU que es equivalente en instrucciones al Arduino, pero con conexión Wifi, luego con el servicio Blink podremos manejar nuestro rover desde el móvil y leer sus sensores con suma facilidad.

- [0 Inicio](#)
- [Pensamiento computacional](#)
- [Robótica y accesibilidad](#)
- [1 Introducción](#)
 - [1.2 Material](#)
 - [1.3 Conexiones](#)
- [2 ARDUINOBLOCKS](#)
 - [Introducción](#)
 - [Crear cuenta](#)
 - [Cuentas alumnado](#)
 - [ArduinoBlocks connector](#)
 - [Empezando](#)
 - [Intermitente](#)

- [¿Qué es Arduino?](#)
- [Software del Arduino](#)
- [Sensores](#)
- [Sensores del rover Arduino](#)
- [Actuadores y otras salidas](#)
- [Motores en el rover Arduino](#)

- [3 BLYNK](#)
 - [Internet de las cosas IoT](#)
 - [Qué es Blynk](#)
 - [Blynk legacy y Blynk IoT](#)

- [4 BLYNK LEGACY](#)
 - [Conseguir la App BLYNK LEGACY](#)
 - [Crear cuentas](#)
 - [Entrando en el Blynk Legacy: El panel de control](#)
 - [En la APP](#)
 - [Poner el TOKEN en ARDUINOBLOCKS](#)
 - [Encender LED](#)
 - [Encender gradualmente LED](#)
 - [Medir la luz del LDR](#)
 - [Medir Temperatura y Humedad con el DHT22](#)
 - [Medir el campo magnético](#)
 - [A por los motores](#)
 - [A divertirnos con un Joystick](#)
 - [Proyecto final](#)

- [5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL](#)
 - [Servidor local Blynk](#)
 - [OJO ESTO NO TIENES QUE HACERLO](#)
 - [Raspberry: Qué hay que hacer para empezar](#)

- [Raspberry: Poner un Sistema Operativo](#)
 - [Raspberry: Conectarlo a la red](#)
 - [Raspberry: Conexión SSH](#)
 - [Raspberry: Cambiar usuario y contraseña](#)
 - [Raspberry: Apagar](#)
 - [Raspberry VNC](#)
 - [Raspberry: Instalar Java 8](#)
 - [Raspberry: Instalar y ejecutar el servidor Blynk local](#)
 - [En Windows Instalar Java 11](#)
 - [En Windows Instalar Blynk local](#)
 - [En Windows Ejecutar Blynkserver](#)
 - [En Linux](#)
-
- [Créditos](#)

0 Inicio

Objetivos

1. Conocer el Internet de las cosas **IoT** controlando robot tipo rover con sensores con una APP.
2. Explorar por encima la potencialidad de la programación por bloques [ArduinoBlocks](#)
3. Conocer la posibilidad de crear APPs de forma inmediata y sencilla para el control de hardware con [Blink](#)
4. Adentrarnos un poco en el mundo maker y descubrir sus ventajas

Contenidos

1. La electrónica del rover: NodeMCU y sensores
2. Programación por bloques [ArduinoBlocks](#) para controlar la placa **NodeMCU**.
3. Manejo de la [App Blink](#) para controlar el rover por wifi y con el móvil.
4. AVANZADO: Creación de un **servidor local BLINK con Raspbery**



Pensamiento computacional

¿Dónde se encaja este robot? ¿se puede comparar este robot con otros robots de otros cursos que hacemos desde CATEDU?

Esta es la hoja de ruta que proponemos, no se tiene que tomar al pie de la letra, pero intenta ayudar al profesorado que tenga una visión global de tanta oferta:

Como se puede ver ROVER CON ARDUINO tiene la ventaja de tener un precio razonable, y dentro del rango de programación en bloques del Arduino en primaria.

Guía orientativa

https://docs.google.com/presentation/d/e/2PACX-1vQHiZvv1cGHet7eXVy-QcECY4Lj0k0l7ntDi8MevRWHQX-9myA0bfR5l0fMeuGZkWD0Hw-Ob-MGoco_/embed?start=trueloop=true&delayms=3000

Tenemos un **grupo Telegram Robótica Educativa en Aragón,**

<https://t.me/roboticaeducativaaragon>



Robótica y accesibilidad

1.- Introducción

Durante mucho tiempo la robótica fue patrimonio de personas y/o instituciones con alta capacidad económica (podían adquirir las placas con microcontroladores comerciales) y capacidad intelectual (podían entender y programar el funcionamiento de las mismas) siempre dentro de los límites establecidos por las marcas comerciales y lo que pudieran “desvelar” de su funcionamiento, vigilando siempre que la competencia no “robara” sus secretos y “copiara” sus soluciones.

Todo esto saltó por los aires en torno a 2005 con la irrupción de un grupo de profesores y estudiantes jóvenes, que decidieron romper con esta dinámica, tratando de poner a disposición de su alumnado microcontroladores económicamente accesibles y que les permitieran conocer su funcionamiento, sus componentes, e incluso replicarlos y mejorarlos. Nació **Arduino** y el concepto de **Hardware Open Source**. Detrás de este concepto se encuentra la **accesibilidad universal**. En un proyecto Open Source todo el mundo puede venir, ayudar y contribuir, minimizando barreras económicas e intelectuales.

Arduino traslada al hardware un concepto ya muy conocido en el ámbito del software, como es el **software open source o software libre**.

[opensource.png](#)

Software libre

Cuando los desarrolladores de software terminan su creación, tienen múltiples posibilidades de ponerlo a disposición de las personas, y lo hacen con condiciones específicas especificadas en una licencia. Esta licencia es un contrato entre el creador o propietario de un software y la persona que finalmente acabará utilizando este software. Como usuarios, es nuestro deber conocer las condiciones y permisos con las que el autor ha licenciado su producto, para conocer bajo qué condiciones podemos instalar y utilizar cada programa.

Existen muchas posibilidades de licencias: software privativo, comercial, freeware, shareware, etc.. Nos centraremos aquí en la de software libre.

GNU (<https://www.gnu.org>) es una organización sin ánimo de lucro que puso una primera definición disponible de lo que es software libre: Software libre significa que los usuarios del software tienen libertad (la cuestión no es el precio). Desarrollaron el sistema operativo GNU

para que los usuarios pudiesen tener libertad en sus tareas informáticas. Para GNU, el software libre implica que los usuarios tienen las cuatro libertades esenciales:

1. ejecutar el programa.
2. estudiar y modificar el código fuente del programa.
3. redistribuir copias exactas.
4. distribuir versiones modificadas.

En otras palabras, el software libre es un tipo de software que se distribuye bajo una licencia que **permite a los usuarios utilizarlo, modificarlo y distribuirlo libremente**. Esto significa que los usuarios tienen libertad de ejecutar el software para cualquier propósito, de estudiar cómo funciona el software y de adaptarlo a sus necesidades, de distribuir copias del software a otros usuarios y de mejorar el software y liberar las mejoras al público.

El software libre se basa en el principio de la libertad de uso, y no en el principio de la propiedad. Esto significa que los usuarios tienen la libertad de utilizar el software de la manera que deseen, siempre y cuando no violen las condiciones de la licencia. El software libre es diferente del software propietario, que es el software que se distribuye con restricciones en su uso y modificación. El software propietario suele estar protegido por derechos de autor y solo se puede utilizar bajo los términos y condiciones especificados por el propietario del software.

Recomendamos la visualización de este [video](#) para entender mejor el concepto.

<https://www.youtube.com/embed/nIDVZ816zol>

Más adelante, entorno a 2015, en Reino Unido, surgiría también la placa **BBC Micro:bit**, con la misma filosofía de popularizar y hacer accesible en este caso al alumnado de ese país la programación y la robótica. También hablaremos de ella.

2.- ARDUINO o LA ROBÓTICA ACCESIBLE

Arduino es una **plataforma de hardware y software libre**.

Hardware libre

Esto significa que tanto la placa Arduino como el entorno de desarrollo integrado (IDE) son de código abierto. Arduino permite a los usuarios utilizar, modificar y distribuir tanto el software

como el hardware de manera libre y gratuita, siempre y cuando se respeten las condiciones de las licencias correspondientes.

El hardware libre es un tipo de hardware cuya **documentación y diseño están disponibles de manera gratuita y libre** para su modificación y distribución. Esto permite a los usuarios entender cómo funciona el hardware y adaptarlo a sus necesidades, así como también crear sus propias versiones modificadas del hardware.

Arduino surge como solución al **elevado precio de los microcontroladores** allá por el año 2005. En el ámbito de la educación, los microcontroladores solo se utilizaban en la etapa universitaria, y su coste era tan elevado que muchos proyectos de fin de carrera se quedaban únicamente en prototipos virtuales ya que las universidades no podían proveer a cada estudiante con un microprocesador, contando además que en el propio proceso de experimentación lo más habitual era que una mala conexión hiciera que se rompieran. Otro **gran inconveniente era la dificultad de la programación**. Cada fabricante entregaba su manual de programación, lo que hacía que de unos a otros no hubiera un lenguaje estándar, y la consecuente dificultad de interpretación. Además, su programación era a bajo nivel en lenguaje máquina. Generar una simple PWM requería una ardua y minuciosa secuenciación que podía llevar varias horas hasta conseguir el resultado deseado. Por este motivo, el enfoque de Arduino desde el principio fue ser Open Source tanto en hardware como en software. El desarrollo del hardware fue la parte más sencilla. Orientado a educación, sufre algunas modificaciones frente a los microprocesadores existentes para hacer más fácil su manejo y accesibilidad a cualquier sensor o actuador. El mayor esfuerzo se entregó en todas las líneas de código que hacían posible que ya no hubiera que programar a bajo nivel gracias al IDE de Arduino que incluía bibliotecas y librerías que estandarizaban los procesos y hacían tremendamente sencillo su manejo. Ahora el alumnado para mover un motor, ya no tenía que modificar las tramas de bits del procesador una a una, sino que bastaba con decir que quería moverlo en tal dirección, a tal velocidad, o a equis grados.

Acabábamos de pasar de unos costes muy elevados y una programación muy compleja a tener una **placa accesible, open source y de bajo coste** que además hacía muy **accesible su programación y entendimiento**, características fundamentales para su implantación en educación, hasta tal punto que su uso ya no era exclusivo de universidades, sino que se extiende a la educación secundaria.

[arduinosecundaria.png](#)

Este hecho es fundamental para el desarrollo del Pensamiento Computacional en el aula observándose que su accesibilidad y beneficios son tales, que alcanzan a **centros con alumnado de toda tipología** como la aplicación del pensamiento computacional y robótica en aulas con alumnos de necesidades especiales. Una vez más, aparece el concepto de accesibilidad asociado a

esta filosofía Open Source.

A este respecto, recomendamos la lectura de [este interesante blog](#), que tiene por título: "ROBOTIQUEAMOS..." Experiencia de aproximación a la robótica en Educación Especial (CPEE ÁNGEL RIVIÈRE). También recomendamos los trabajos robótica en Educación Especial (CPEE ÁNGEL RIVIÈRE): <http://zaragozacpeeangelriviere.blogspot.com/search/label/ROB%C3%93TICA>

[blogRobotiqueamos.jpg](#)

Igualmente, la aparición de Arduino supone una gran facilidad para la aplicación de la robótica y la programación en la atención temprana, donde son numerosas sus aplicaciones desde ayudar a mitigar el déficit de atención en jóvenes autistas, hasta ayudar a socializar a los alumnos con dificultades para ello, o ayudar a alumnos de altas capacidades a desarrollar sus ideas.

Por otro lado su accesibilidad económica lo ha llevado a popularizarse en países de **todo el mundo**, especialmente en aquellos cuyos sistemas educativos no disponen en muchas ocasiones de recursos suficientes, lo que supone en la práctica una **democratización del conocimiento y superación de brecha digital**.

Filosofía del Arduino ver vídeo

Arduino y su IDE son la primera solución que aparece en educación con todas las ventajas que hemos enumerado, y esto hace que todos los nuevos prototipados y semejantes tengan algo en común, siempre son compatibles con Arduino

Para entender bien la filosofía de Arduino y el hardware libre, os recomendamos este documental de 30 minutos. [Arduino the Documentary](#)

<https://player.vimeo.com/video/18390711?h=b5844e7753>

Scratch: software libre para el desarrollo del pensamiento computacional



Scratch es un lenguaje de programación visual desarrollado por el grupo Lifelong Kindergarten del MIT Media Lab. Scratch es un software libre. Esto significa que está disponible gratuitamente para todos y que se distribuye bajo una licencia de software libre, la Licencia Pública General de Massachusetts (MIT License). Esta licencia permite a los usuarios utilizar, modificar y distribuir el software de manera libre, siempre y cuando se respeten ciertas condiciones. Entre otras cosas, la licencia de Scratch permite a los usuarios utilizar el software para cualquier propósito, incluyendo fines comerciales. También permite modificar el software y distribuir las modificaciones, siempre y cuando se incluya una copia de la licencia y se indique que el software ha sido modificado. En resumen, Scratch es un software libre que permite a los usuarios utilizar, modificar y distribuir el software de manera libre y gratuita, siempre y cuando se respeten las condiciones de la licencia. De hecho, gracias a que está licenciado de esta forma, han surgido decenas de variaciones de Scratch para todo tipos de propósitos, eso sí, siempre educativos y relacionados con las enseñanzas de programación y robótica

3. BBC micro:bit y la Teoría del Cambio

BBC micro:bit, a veces escrito como Microbit o Micro Bit, es un pequeño ordenador del tamaño de media tarjeta de crédito, creado en 2015 por la BBC con el fin de promover el desarrollo de la robótica y el pensamiento computacional entre la población escolar del Reino Unido. Actualmente su uso está extendido entre 25 millones de escolares de 7 a 16 años de más de 60 países.

[e74cc3a97963070daee67213f9ccf5268388bd01-790x635.webp](https://microbit.org) Tarjeta BBC micro:bit V1. Fuente: <https://microbit.org>. CC BY-SA 4.0.

Aunque el proyecto fue iniciado por la BBC, su desarrollo fue llevado a cabo por 29 socios tecnológicos de primera línea. Por ejemplo, la implementación del Bluetooth integrado en la tarjeta corrió a cargo de la fundación propietaria de la marca, Bluetooth SIG, una asociación privada sin ánimo de lucro.

El hardware y el software resultantes son 100% abiertos, y están gestionados por una fundación sin ánimo de lucro que comenzó a funcionar en el año 2016, la [Micro:bit Educational Foundation](#). La fundación basa sus actuaciones en su Teoría del Cambio,

Teoría del cambio y más sobre microbit

Teoría del cambio puede resumirse en tres principios:

- El convencimiento de que la capacidad de comprender, participar y trabajar en el mundo digital es de vital importancia para las oportunidades de vida de una persona joven.
- La necesidad de emocionar y atraer a las personas jóvenes por medio de BBC micro:bit, especialmente a las que podrían pensar que la tecnología no es para ellas.
- Diversificar a los estudiantes que eligen las materias STEM a medida que avanzan en la escuela y en sus carreras, para hacer crecer una fuente diversa de talento, impulsando la equidad social y contribuyendo a crear una tecnología mejor.

Para desarrollar sus principios, la fundación trabaja en tres líneas de acción:

- El desarrollo de hardware y software que contribuyan a despertar el entusiasmo en las personas jóvenes hacia la tecnología y hacia las oportunidades que presenta.
- La creación de recursos educativos gratuitos y fáciles de usar que permitan al profesorado enseñar de forma atractiva y creativa.
- La colaboración con entidades asociadas que compartan una misma visión para ofrecer programas educativos de alto impacto en todo el mundo.

Uno de los objetivos de la Micro:bit Educational Foundation es llegar a 100 millones de escolares en todo el mundo.

En correspondencia con las líneas de acción y con los principios expuestos, el sistema resultante es muy económico: tanto las placas como los accesorios producidos por terceras empresas tienen un precio muy contenido. Además, dado el carácter abierto del proyecto, están disponibles algunos clones totalmente compatibles, como Elecrow Mbits o bpi:bit. Estos clones son incluso más potentes y económicos que la placa original.

El universo micro:bit destaca por su **alta integración de software y hardware**: basta un clic de ratón para cargar las librerías necesarias para que funcione cualquier complemento robótico, como sensores, pantallas, tarjetas de Internet de las Cosas, robots, casas domóticas, etc.

La programación de la placa se realiza desde un ordenador a través de un navegador cualquiera, estando disponibles **12 lenguajes de programación**. De nuevo, por ser un sistema abierto, existen múltiples soluciones de programación, aunque las más común es [MakeCode](#).

[MakeCode.png](#) Captura de pantalla del editor MakeCode, <https://makecode.microbit.org/#>.



El sitio web MakeCode permite programar con bloques y también en Python y en Java, traduciendo de un lenguaje a otro instantáneamente. No se necesita ningún registro en la plataforma para poder programar.

Los programas también pueden guardarse descargados en el ordenador compilados en código de máquina. Al subir de nuevo el programa al editor, se realiza una decompilación automática al lenguaje de bloques, Python o Java. Los programas guardados en código de máquina se pueden cargar directamente en micro:bit, que en el escritorio de un ordenador se maneja como una simple unidad de memoria USB.

MakeCode contiene además múltiples recursos como tutoriales, vídeos, fichas de programación, cursos para el profesorado, ejemplos y propuestas de proyectos y experimentos, todo ello en varios idiomas y clasificado por edades desde los 7 años.

Otra solución muy usada para programar micro:bit es [MicroPython](#), creada por Python Software Foundation, otra organización sin ánimo de lucro.

[MicroCode](#) permite que los más pequeños, a partir de los 6 años de edad, programen micro:bit mediante un sistema de fichas dispuestas en líneas de acción. Están disponibles un tutorial introductorio en 20 idiomas, una guía del usuario y muchos ejemplos. El proyecto es de código abierto.

Micro:bit también es programable en **Scratch** con sólo añadir una extensión al editor.

Todos los entornos de desarrollo descritos disponen de un simulador de micro:bit, por lo que ni siquiera resulta necesario disponer de una tarjeta física para aprender a programar.

Una vez realizada la programación, la placa y sus complementos pueden funcionar desconectados del ordenador por medio de un cargador de móvil, una batería externa o un simple par de pilas alcalinas.

Versiones y características de micro:bit

A pesar de su pequeño tamaño, micro:bit es un sistema potente. Existen dos versiones de la placa. La más moderna, llamada micro:bit V2, tiene las siguientes características:

- Procesador de 64 MHz.



- 512 KB de RAM Flash y 128 KB de RAM.
- Matriz de 5 x 5 LED rojos.
- Dos pulsadores mecánicos y un tercer pulsador de apagado y reset.
- Un pulsador táctil.
- Micrófono y altavoz.
- Acelerómetro y brújula.
- Sensores de luz y de temperatura.
- Comunicación con otras placas por Bluetooth de bajo consumo.
- Alimentación a 3 V o por USB.
- 25 pines de entradas y salidas para conectar motorcitos, sensores, placas de Intenet de las Cosas, robots y, en general, cualquier otro tipo de accesorio.
- 200 mA de intensidad de corriente disponibles en las salidas para alimentar accesorios.

4.- LA IMPORTANCIA DEL OPEN SOURCE / CÓDIGO ABIERTO EN EDUCACIÓN

La creación, distribución, modificación y redistribución del hardware y software libre así como su utilización, están asociados a una serie de valores que deberían ser explicados en la escuela a nuestros alumnos para dar una alternativa a la versión mercantilista de que cualquier creación es creada para obtener beneficios económicos.

En GNU, pusieron especial énfasis en la difusión del software libre en colegios y universidades, promoviendo una serie de valores fundacionales:

Valores GNU

Compartir

El código fuente y los métodos del hardware y software libre son parte del conocimiento humano. Al contrario, el hardware software privativo es conocimiento secreto y restringido. El código abierto no es simplemente un asunto técnico, es un asunto ético, social y político. Es una cuestión de derechos humanos que la personas usuarias deben tener. La libertad y la cooperación son valores esenciales del código abierto. El sistema GNU pone en práctica estos valores y el principio del compartir, pues compartir es bueno y útil para el progreso de la humanidad. Las escuelas deben enseñar el valor de compartir dando ejemplo. El hardware y software libre favorece la educación pues permite compartir conocimientos y herramientas.



Responsabilidad social

La informática, electrónica, robótica... han pasado a ser una parte esencial de la vida diaria. La tecnología digital está transformando la sociedad muy rápidamente y las escuelas ejercen una influencia decisiva en el futuro de la sociedad. Su misión es preparar al alumnado para que participen en una sociedad digital libre, mediante la enseñanza de habilidades que les permitan tomar el control de sus propias vidas con facilidad. El hardware y el software no debería estar bajo el poder de un desarrollador que toma decisiones unilaterales que nadie más puede cambiar.

Independencia

Las escuelas tienen la responsabilidad ética de enseñar la fortaleza, no la dependencia de un único producto o de una poderosa empresa en particular. Además, al elegir hardware y software libre, la misma escuela gana independencia de cualquier interés comercial y evita permanecer cautiva de un único proveedor. Las licencias de hardware y software libre no expiran

Aprendizaje

Con el open source los estudiantes tienen la libertad de examinar cómo funcionan los dispositivos y programas y aprender cómo adaptarlos si fuera necesario. Con el software libre se aprende también la ética del desarrollo de software y la práctica profesional.

Ahorro

Esta es una ventaja obvia que percibirán inmediatamente muchos administradores de instituciones educativas, pero se trata de un beneficio marginal. El punto principal de este aspecto es que, por estar autorizadas a distribuir copias de los programas a bajo costo o gratuitamente, las escuelas pueden realmente ayudar a las familias que se encuentran en dificultad económica, con lo cual promueven la equidad y la igualdad de oportunidades de aprendizaje entre los estudiantes, y contribuyen de forma decisiva a ser una escuela inclusiva.

Calidad

Estable, seguro y fácilmente instalable, el software libre ofrece una amplia gama de soluciones para la educación.

Para saber más

En los años 90, era realmente complicado utilizar un sistema operativo Linux y la mayoría de la cuota del mercado de los ordenadores personales estaba dominada por Windows. Encontrar drivers de Linux para el hardware que tenía tu equipo era casi una quimera dado que las principales compañías de hardware y de software no se molestaban en crear software para este sistema operativo, puesto que alimentaba la independencia de los usuarios con respecto a ellas mismas.

Afortunadamente, y gracias a la creciente presión de su comunidad de usuarios, estas situaciones pertenecen al pasado, y las compañías fabricantes de hardware han tenido que variar el rumbo. Hoy en día tenemos una gran cantidad de argumentos en los que nos podemos basar para dar el salto hacia cualquier sistema operativo basado en Linux. Tal y como podemos leer en educacionit.com, podemos encontrar las siguientes ventajas:

- Es seguro y respeta la privacidad de los usuarios: Aunque hay compañías linuxeras, como Oracle, Novell, Canonical, Red Hat o SUSE, el grueso de distribuciones y software Linux está mantenido por usuarios y colectivos sin ánimo de lucro. De esta forma, podemos confiar en que una comunidad que tiene detrás millones de usuarios, pueda validar el código fuente de cualquier de estas distribuciones, asegurándonos la calidad de las mismas, compartir posibles problemas de seguridad, y sobre todo, estar bien tranquilos con la privacidad y seguridad de nuestros datos e información personal, aspecto que debería ser crítico y determinante a la hora de trabajar con los datos de menores de edad en las escuelas y colegios.
- Es ético y socialmente responsable: La naturaleza de Linux y su filosofía de código abierto y libre hace posible que cualquier usuario con conocimientos pueda crear su propia distribución basada en otras o probar las decenas de versiones que nos podemos encontrar de una distribución Linux. Este es el caso de Ubuntu por ejemplo. Gracias a esta democratización de los sistemas operativos, incluso han podido aparecer en nuestras vidas nuevos dispositivos basados en software y hardware libre como Arduino y Raspberry Pi.
- Es personalizable: el código abierto permite su estudio, modificación y adaptación a las necesidades de los diferentes usuarios, teniendo así no un único producto sino una multiplicidad de distribuciones que satisfacen las necesidades de los diferentes colectivos a los que se dirijan. Especialmente útiles son las distribuciones educativas libres, que pueden ser adaptadas a las necesidades de las escuelas.
- Está basado en las necesidades de los usuarios y no en las de los creadores de hardware y software
- Es gratis. La mayoría de las distribuciones Linux son gratuitas y de libre descarga

- Es fácil de usar. Una de las barreras que durante años ha evitado a muchos usar Linux es su complejidad. Las distribuciones orientadas al consumo doméstico cumplen los estándares de simplicidad y necesidades que cualquier usuario sin conocimientos de tecnología pueda necesitar. El entorno gráfico es sencillo, intuitivo, e incluso se puede customizar para que se pueda parecer a los más conocidos como Windows y MacOS. Además, vienen con la mayoría de aplicaciones que cualquier usuario puede necesitar: ofimáticas, edición de audio y vídeo y navegación por Internet.
- Es suficiente. Tiene su propio market de aplicaciones. Como el resto de sistemas operativos ya sea para ordenadores o dispositivos móviles, también podemos encontrar un lugar único donde poder descargar cientos de aplicaciones para todos los gustos y necesidades.

Por estas razones, el software libre se ha expandido por toda la comunidad educativa en los últimos años de manera exponencial. Un buen ejemplo de lo que estamos hablando es **Bookstack**, este sistema de edición de contenidos para cursos que utiliza Aularagón así como el uso de **Moodle** como plataforma de enseñanza y aprendizaje. En cuanto a sistema operativo para ordenadores, en Aragón disponemos de nuestra propia distribución Linux: Vitalinux EDU. Tal y como podemos leer desde su página web: **Vitalinux EDU (DGA)** es la distribución Linux elegida por el Gobierno de Aragón para los centros educativos. Está basada en Vitalinux, que se define como un proyecto para llevar el Software Libre a personas y organizaciones facilitando al máximo su instalación, uso y mantenimiento. En concreto Vitalinux EDU (DGA) es una distribución Ubuntu (Lubuntu) personalizada para Educación, "tuneada" por los requisitos y necesidades de los propios usuarios de los centros y adaptada de forma personalizada a cada centro y a la que se ha añadido una aplicación cliente Migasfree. De ésta forma, obtenemos:

1. Un **Sistema Ligero**. Permite "revivir" equipos obsoletos y "volar" en equipos modernos. Esto garantiza la sostenibilidad de un sistema que no consume recursos de hardware innecesariamente ni obliga a la sustitución del hardware cada poco tiempo en esa espiral de obsolescencia programada en la que se ha convertido el mercado tecnológico.
2. **Facilidad en la instalación y el uso** del sistema mediante programas personalizados.
3. Un Sistema que **se adapta al centro** y/o a cada aula o espacio, y no un centro que se adapta a un Sistema Operativo.
4. **Gestión de equipo y del software de manera remota** y desatendida mediante un servidor Migasfree.
5. **Inventario** de todo el hardware y software del equipo de una forma muy cómoda.
6. Soporte y apoyo de una **comunidad** que crea, comparte e innova constantemente.



1 Introducción

1 Introducción

1.2 Material

MATERIAL

El precio de este kit es aproximadamente [50€](#)

El Kit de préstamo de Aulargón y en este presupuesto **no incluye la Raspberry** del apartado 4 Avanzado para montar un servidor Blynk local

El kit de préstamo incluye todo este material ya montado

<https://docs.google.com/spreadsheets/d/e/2PACX->

[1vS8HWZ4aCeZcHxtyiAZuvF63jOnPjccLIWO2GYEuB8PjXXIWHIIPILGmwHAgAKJHSFQWrMAu5zVqXM](https://docs.google.com/spreadsheets/d/e/2PACX-1vS8HWZ4aCeZcHxtyiAZuvF63jOnPjccLIWO2GYEuB8PjXXIWHIIPILGmwHAgAKJHSFQWrMAu5zVqXM)
[F/pubhtml?gid=0single=true&widget=true&headers=false](#)



NodeMCU



Es una placa de desarrollo basado en el **SoC** ([System on a chip](#)). ESP8266 e incorpora la comunicación Wifi que tanto falta en el Arduino.

Desde que salió el ESP01 como primer módulo del ESP8266, se ha evolucionado mucho, mejorando la popularidad, potencialidad y bajando los precios. Actualmente (2020) se fabrica el ESP32. Ver

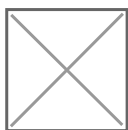
[CURSO ESP32 EN EL AULA.](#)



Nosotros utilizamos en el Rover el NodeMCU basado en el ESP12E conocida como **NodeMCU** que apareció en el 2014 que se programaba con [Lua](#).

Con el paso del tiempo esta placa se programa en varios lenguajes, siendo la más interesante desde el punto pedagógico que se pueda programar en el didáctico **entorno Arduino**.

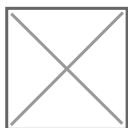
Existen disparidad en los fabricantes, en este curso se va a utilizar la generación segunda de 8 pines



De [Luis Llamas](#) CC-BY-NC-SA

Para más información recomendamos la página de Luis Llamas (<https://www.luisllamas.es/esp8266-nodemcu/>)

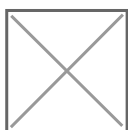
Shield motor para NodeMCU



Esta shield permite conectar motores en los 4 primeros terminales azules con el NodeMCU que se coloca en el zócalo. Para saber cómo funcionan los motores con esta shield [ver aquí](#)

También tiene unos machos para las entradas y salidas digitales del NodeMCU y uno para la entrada analógica. Esto lo vemos en [conexiones](#).

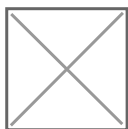
Chasis





Se elige este chasis por su bajo coste y con tracción 4x4 (en Marte no hay carreteras asfaltadas ☐☐).

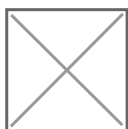
DHT22



Créditos: CC-BY-NC-SA [Luis Llamas](#)

Es un sensor que mide de forma sencilla tanto la temperatura como la humedad. Este sensor aparece en el [curso de Aularagón Arduino con código](#)

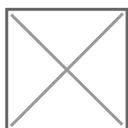
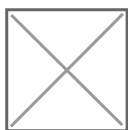
Su conexión es sencilla :



Créditos: CC-BY-NC-SA [Luis Llamas](#)

Para saber más recomendamos la página de [Luis Llamas](#)

Sensor efecto Hall



De Desconocido - [Popular Science Monthly Volume 64, Dominio público](#)

[Edwin Helbert Hall](#) descubrió en 1879 que en presencia de un campo magnético, un conductor que conduzca una corriente se le producía un campo eléctrico porque las cargas eléctricas se desviaban de su trayectoria principal, nuestro sensor simplemente mide ese campo eléctrico:



De [Luis Llamas](#) CC-BY-NC

El sensor tiene un led de color rojo que indica que hay una lectura de campo magnético.

Nuestro rover analizará si las rocas que hay cerca de él tienen ferritas midiendo su componente magnético. Una aproximación tosca de los rovers actuales pero sirve para acercar el mundo de la medición remota al alumnado :

<https://www.youtube.com/embed/NphWPvi9cy4>

Nuestro robot detectará si hay un imán cerca o no .

https://www.youtube.com/embed/ZqYQaWH_yE4

Sensor de luz LDR.



El LDR es una resistencia que varía con la luz. La conexión interna de este sensor analógico hace que **cuanto más luz menos valor nos proporciona** por lo tanto va al revés. Esto será importante para la configuración de nuestros programas, por ejemplo el medidor "Gauge" de la APP Blink de este vídeo va de 1023 a 0 para medir correctamente la cantidad de luz:



https://www.youtube.com/embed/n6m_8mGBwXY

1 Introducción

1.3 Conexiones

Conexiones motores y batería

Este es el esquema de conexión de los motores y de la batería con el Shield :



Una alternativa a la conexión de la batería es utilizar un PowerBank y diréctamente al NodeMCU como podemos ver en la foto :



Otra posibilidad es utilizar baterías 18650 [ver](#) pero ojo con usar estas baterías [ver+](#)

Conexiones NodeMCU sensores con la Shield.

El sensor de efecto Hall y el sensor DHT22 lo conectaremos a las entradas digitales, nosotros hemos elegido

- DHT22 a D5
- Efecto Hall al D6

Es recomendable no usar D4 ni D0 pues los utilizaremos como Led

El sensor de luz LDR al ser analógico, lo conectaremos en la única entrada analógica que tiene esta shield





2 ARDUINOBLOCKS

2 ARDUINOBLOCKS

Introducción

Esto no pretende ser un tutorial exhaustivo de STEAMAKERSBLOCKS, sino una guía rápida. STEAMAKERSBLOCKS es un programa que tiene muchas posibilidades. Si quieres saber más sobre ARDUINOBLOCKS tutoriales, ejemplos, foro.... te recomendamos

<http://arduinoblocks.didactronica.com/> o el libro [Arduino blocks - libros y tutoriales](#)

IMPORTANTE SABERLO: TENEMOS UN CHAT DE Ayuda en STEMAKERSBLOCKS

hay chat de **Telegram** con una comunidad de profesores y técnicos de la empresa que apoya Arduinoblocks donde puedes encontrar proyectos, enlaces interesantes y lo más importante: Puedes preguntar tus dudas o problemas

https://t.me/innovadidactic_comunidad

logogrupotelegramarduinoblocks.jpg

¿Por qué una programación con bloques?

Arduino se programa en lenguaje C++ (con algunas variaciones para simplificarlo). Para programar normalmente se utiliza el IDE ("Integrated Development Environment"/"Entorno de Desarrollo Integrado") de Arduino, que permite escribir el código, compilar el programa (crear el programa binario para el procesador Arduino) y grabarlo en la placa Arduino a través del puerto USB. El IDE de Arduino se puede descargar [desde la web oficial](#). Es totalmente libre (José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

2022-04-13 12_45_21-PROYECTO 00 CONOCEMOS ARDUINO. - PDF-XChange Viewer.png

(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

Sin embargo pensando en edades más tempranas se han desarrollado formas más sencillas e intuitivas de programar Arduino como son los **lenguajes de programación por bloques**. De todos estos lenguajes cabe destacar **STEAMAKERSBLOCKS**.

Gracias a este lenguaje visual podemos programar las placas Arduino sin necesidad de escribir ni una sola línea de código, de esta forma podemos empezar a realizar proyectos con Arduino de una forma muy rápida y a edades más tempranas. La única desventaja es que el lenguaje por código

tiene todo el potencial que requiere la programación de un experto.

Mismo programa creado con el IDE de Arduino en C++ (imagen de la izquierda) y con Arduinoblocks (imagen de la derecha).

(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

[2022-04-13 12_47_24-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

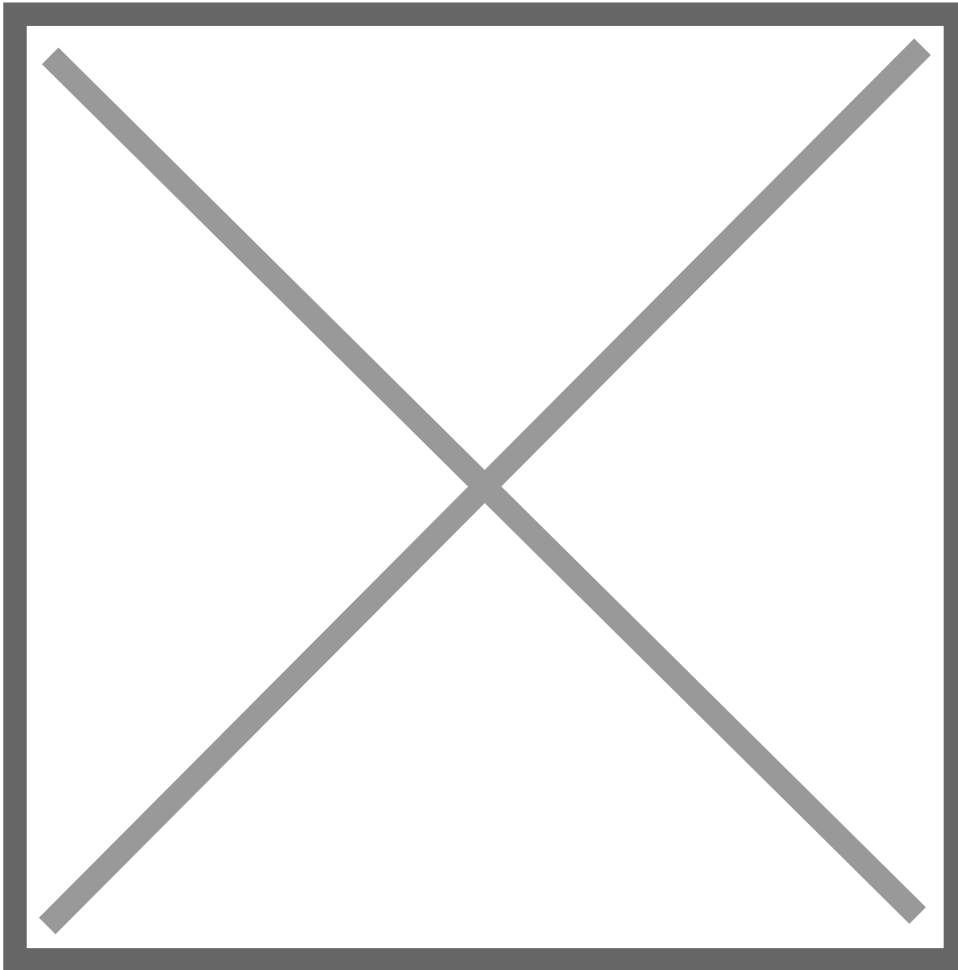
Para trabajar con Arduinoblocks debemos ir a su página web <http://www.arduinoblocks.com/> desde cualquier navegador y para cualquier sistema operativo (Windows, Linux, Mac). *(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)*

STEAMAKERSBLOCKS (antes Arduinoblocks)



2025-10-22 10_31_17-.png antes

STEMAKERSBLOCKS es un programa creado por el profesor Juanjo López. Gracias a su entorno gráfico facilita la programación de placas Arduino a todos los niveles. Esta herramienta permite programar a personas sin conocimientos previos de programación, pero su versatilidad y potencia es tan grande que expertos programadores también pueden utilizarlo. *(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)*



De [Juan José López Almendros](#) CC-BY-SA

La programación en **STEMAKERSBLOCKS** se realiza con bloques al estilo AppInventor o Scratch, *se puede utilizar a partir de 8 años*. No tenemos que escribir líneas de código y no nos permitirá unir bloques incompatibles evitando así posibles errores de sintaxis. La plataforma **STEMAKERSBLOCKS** genera, compila y sube el programa a la placa Arduino por medio de la conexión USB. Una vez subido el programa, la placa el Arduino no necesitará de la conexión al PC para funcionar pudiendo alimentarla con baterías o una fuente de alimentación para que funcione de forma autónoma.

STEMAKERSBLOCKS actualmente funciona con todos los navegadores de última generación: Mozilla Firefox, Google Chrome, Opera, Safari,... (José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

Por otro lado, tal y como se describe en la [Wiki](#) de **Vitalinux**, **STEMAKERSBLOCKS** funciona perfectamente con todos los sistemas operativos, pudiendo ser fácilmente instalable en equipos individuales y a nivel de centro dentro del soporte de **Vitalinux**.



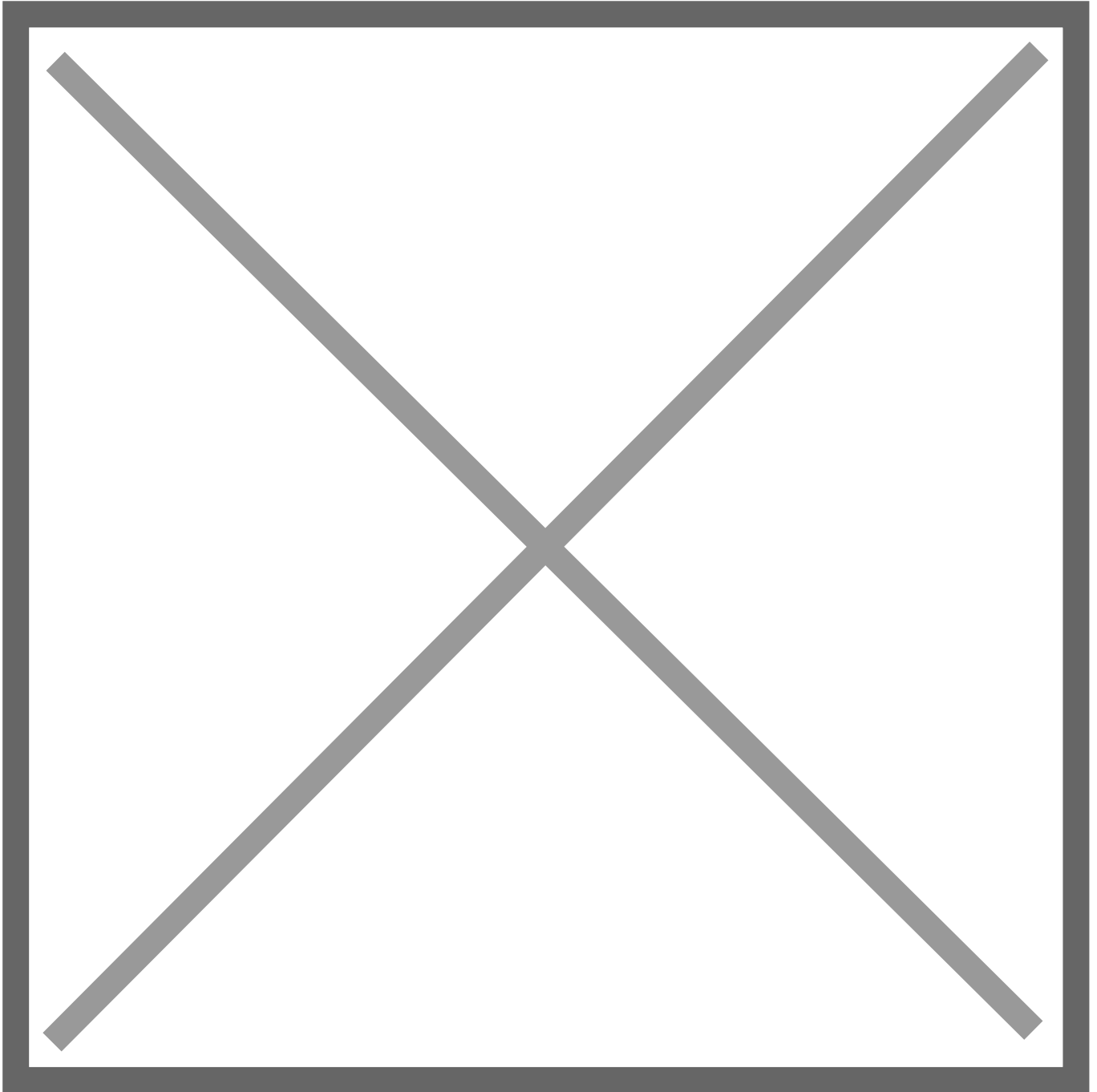
2 ARDUINOBLOCKS

Crear cuenta

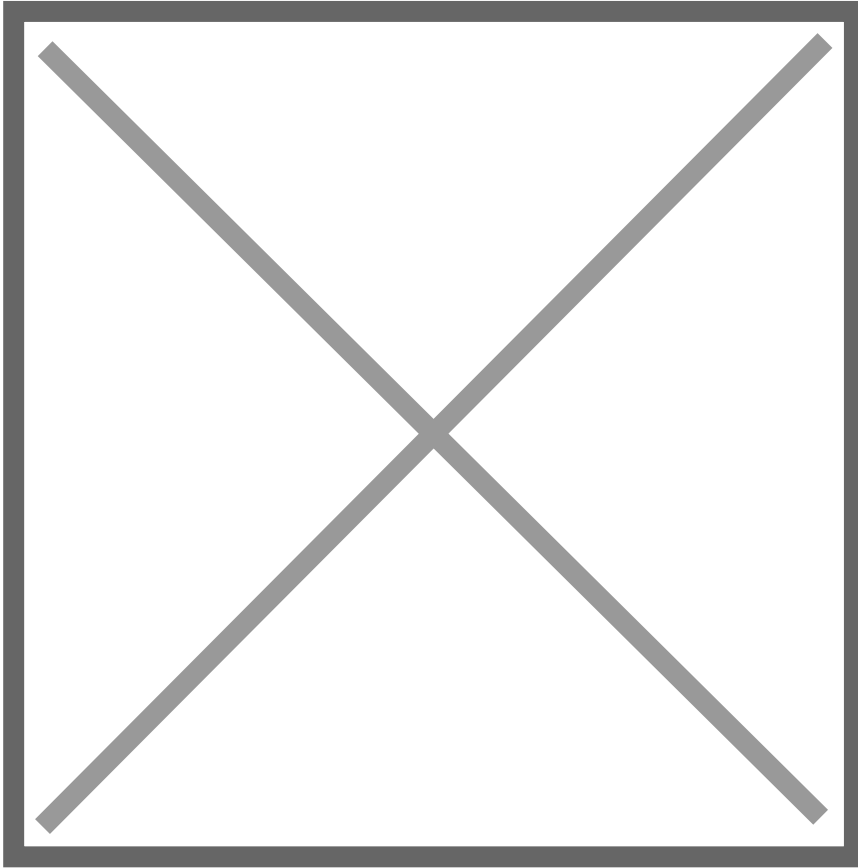
Registrándonos como usuarios de la plataforma **Steamakerblocks** podemos aprovechar todas estas posibilidades:

- Guardar tus proyectos en la nube de **Steamakerblocks** .
- Añadir información al proyecto: descripción, componentes utilizados, imágenes, etc.
- Añadir archivos adjuntos relacionados con el proyecto: esquemas, fotos, archivos para impresión 3D, aplicaciones, etc.
- Compartir proyectos con el resto del mundo.
- Importar proyectos compartidos por otros usuarios.
- Valorar y comentar proyectos.
- Programar directamente Arduino desde el propio navegador (con la aplicación: **Steamakerblocks -Connector**).
- Utilizar la consola serie desde el propio navegador.

Entramos en <https://www.steamakersblocks.com/> e iniciamos sesión



Y rellenamos el formulario



Autor José Andrés Echevarría @cantabRobots CC-BY-NC-SA

2 ARDUINOBLOCKS

Cuentas alumnado

Tal y como dice el tutorial de Juanjo López : *Permite a un usuario registrado con email, crear y administrar nuevas cuentas de usuario dentro de una organización, centro educativo o institución.*

- ♥Permite crear usuarios alumnos sin necesidad de ceder ningún tipo de datos
- ☐☐Puedes pasar proyectos a los alumnos, vacíos o empezados
- ☐☐Tú puedes ver los proyectos de los alumnos y ponerles comentarios

<https://www.youtube.com/embed/zdzKX0NX60Y>

- Si lo quieres en papel, te recomendamos el tutorial de Juanjo López son 12 diapositivas muy bien explicados
 - [Repositorio Github](#)
 - [Repositorio en steamakerbloks.com](#)
- Prueba con Catedu:
 - <https://www.steamakersblocks.com/>
 - usuario alumnox.catedu donde x de 1 a 20
 - contraseña [donde esta catedu en minúsculas]

2 ARDUINOBLOCKS

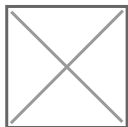
ArduinoBlocks connector

Espera !!! Aún no conectes tu placa (ESP32, Arduino, TDR STEAM...)

PRIMER PASO Descargar e instalar Connector

Para poder usar la herramienta **Steamakerblocks** tenemos que ejecutar antes **Connector**. Lo descargamos de la misma página de **Steamakerblocks** según el sistema operativo que usemos: Windows (**W7 E INFERIORES NO FUNCIONA**), Linux

Entra en <https://www.steamakersblocks.com/> y en Recursos, tienes la web para descargar este programa:



Lo descargamos y lo **instalamos**.

En el caso de tener equipos Vitalinux, es fácilmente accesible e instalable desde la aplicación **Vitalinux Play** o si se desea una instalación masiva en el centro a través de su página de [soporte](#):

[image-1667329017653.png](#)

SEGUNDO PASO: INSTALAR LOS DRIVERS

Si no hacemos estos pasos, cuando conectamos la placa, siempre sale en el COM1, le damos a subir y sale error

En <http://www.arduinoblocks.com/web/site/abconnector5> tenemos abajo ARDUINO SERIAL DRIVERS

RECOMENDAMOS EL PRIMER ENLACE Y EL TERCERO

2024-06-26 11_27_26-Coding with blocks ;).png

En el primero el instalador está en este enlace

https://cdn.sparkfun.com/assets/learn_tutorials/7/4/CDM21228_Setup.exe

2024-06-26 11_28_55-How to Install FTDI Drivers - SparkFun Learn.png

El segundo sólo si quieres utilizar Arduinos no oficiales, de fabricantes chinos, que tiene el CH340g y hay que leerse la página, paciencia

El tercero es necesario el 2102 si utilizas el ESP32 el instalador esta en este enlace, es una carpeta comprimida, la descomprimes y está el ejecutable instalador

https://www.silabs.com/documents/public/software/CP210x_Windows_Drivers.zip

2024-06-26 11_38_15-CP210x USB to UART Bridge VCP Drivers - Silicon Labs.png

concretamente hay que ejecutar este (al menos que el equipo sea muy viejo de 32bits)

2026-01-19 08_54_16-CP210x_Windows_Drivers - Explorador de archivos.png

COMPROBAR QUE DETECTA LA PLACA

Ahora **conectamos la placa** (ESP32, Arduino, NodeMCU, KeyStudio TDR STEAM...) a nuestro ordenador, y observamos si lo detecta, en Windows entramos en Administrador de dispositivos:

[administradordispositivos.png](#)

Y vemos que en los puertos COM se ha detectado correctamente la placa:

[puertos.png](#)

En el caso de que no aparezca, es que no se han instalado correctamente los *drivers* de Arduino. Entonces vamos a la página oficial de Arduino y descargamos el programa **ARDUINO IDE** : <https://www.arduino.cc/en/software> y lo instalamos. Al instalar este programa se instalan los drivers en nuestro ordenador. No hace falta ejecutarlo.

En el caso de equipos con sistema operativo Linux (como Vitalinux) el puerto serie tiene la forma **/dev/XXXX**

YA PUEDES EJECUTAR ARDUINOBLOCKS CONNECTOR

Ahora buscamos el programa ArduinoBlocks connector que hemos descargado e instalado en el primer paso y lo **ejecutamos**.

[2024-06-26 12_36_52-AB-Connector v5.3.png](#)

ATENCIÓN No podemos cerrar la ventana mientras utilizamos *Arduinoblocks*, la minimizamos simplemente.

En caso contrario, Arduinoblocks no se puede comunicar con nuestra placa Arduino, NodeMCU, KeyStudio, etc

YA PUEDES EJECUTAR ARDUINOBLOCKS

Entramos en la web ARDUINOBLOCKS <http://www.arduinoblocks.com/> nos logueamos e iniciamos un proyecto, Vemos que en el editor que aparece ya los puertos COM (si no te aparece, dale a la rueda actualizar)

Aparecen varios COM, **elegir el último que tiene que coincidir con el que has visto en el segundo paso**, no necesariamente es el COM más alto.

Si se queda una ruleta de espera demasiado tiempo, entonces, actualizar la página o darle a actualizar el botón 1 de la figura :

[arduinoblocks-com.jpg](#)

Una vez elegido el COM ya puedes darle al botón amarillo **SUBIR** cuando has realizado tu proyecto pero antes de subir, por si acaso dale a **guardar** el proyecto que has realizado.

En el caso de equipos con Linux veremos algo así:

[image-1667328938836.png](#)

¿Tengo que hacer los cuatro pasos cada vez?

No, sólo la primera vez para asegurar los drivers del Arduino, las siguientes veces que te conectes lo único que tienes que hacer es el tercer y cuarto paso

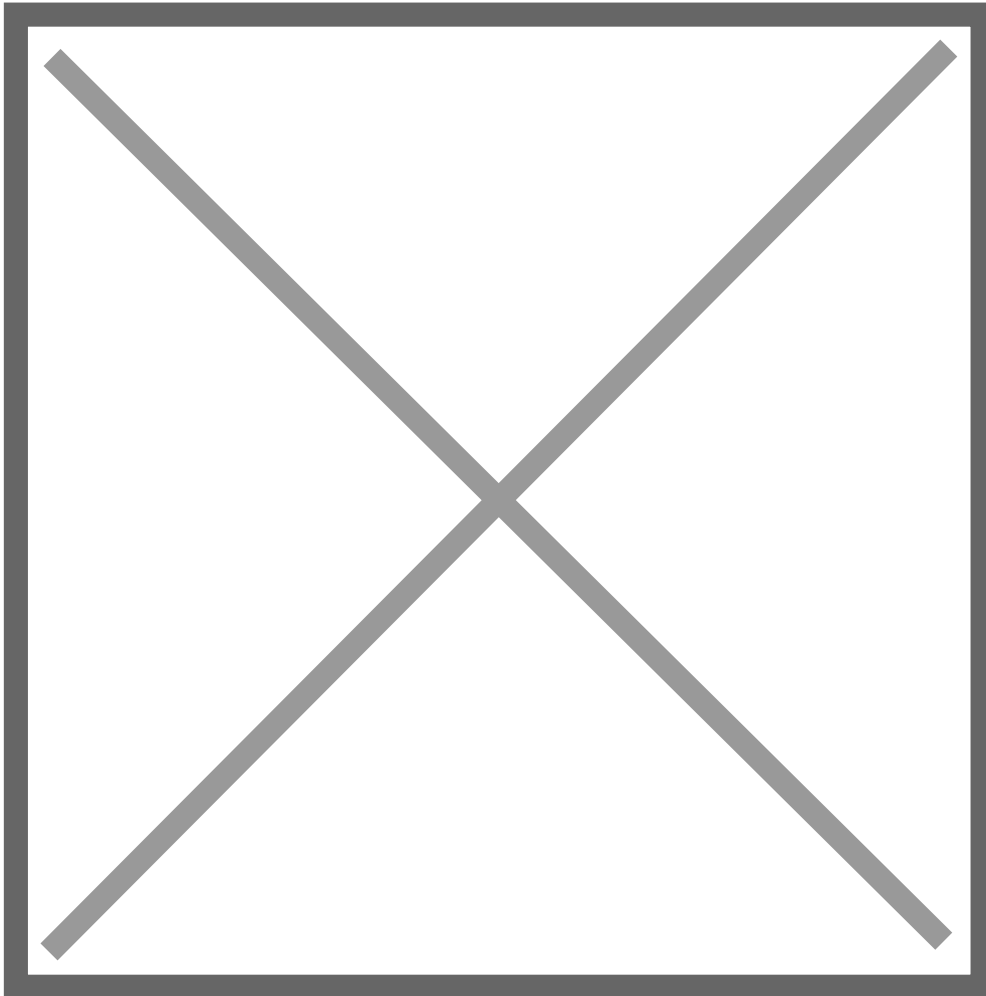


IMPORTANTE: TENER EL SOFTWARE ARDUINOBLOCKS ACTUALIZADO para que funcionen los nuevos bloques que se incorporan en Arudinoblocks

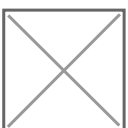
2 ARDUINOBLOCKS

Empezando

Entramos en Proyectos y podemos ver nuestros proyectos creados como también empezar uno.



Y nos aparece tres opciones :



En esta ventana podremos elegir que tipo de proyecto vamos a realizar:

- **Proyecto Personal:** Iniciar un nuevo proyecto que sólo será accesible para el usuario. Posteriormente se puede compartir al resto de la comunidad si se desea.
- **Proyecto Profesor:** Iniciar un proyecto como profesor. De esta forma no se inicia un proyecto como tal, sino que se especifican los datos del proyecto y se genera un código para que los alumnos se puedan suscribir al proyecto. El profesor podrá supervisar y valorar los proyectos de sus alumnos.
- **Alumno:** De esta forma nos unimos a un proyecto planteado por el profesor. Nosotros realizaremos el proyecto como si de un proyecto personal se tratara, pero el profesor podrá supervisar y valorar nuestro trabajo.

Adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

Lo primero que tenemos que elegir es para qué tipo de placa se hace el proyecto.

- En el caso de que estés con el kit de CATEDU **Rover marciano con Arduinoblocks** el tipo de proyecto es para **ESP8266 / NodeMCU**
- En el caso de que estés con el kit de CATEDU **Arduino con Arduinoblocks** el tipo de proyecto es para **Arduino UNO**
- En el caso de que estés con el kit de CATEDU **STEAMAKERSBLOCKS en el aula** tienes dos opciones totalmente válidas:
 - **ArduinoUno**
 - **ArduinoUno + Imagina TdR STEAM**
- En el caso de que estés con el kit de CATEDU **ESP32 en el aula** tienes dos opciones totalmente válidas:
 - **ESP32 STEAMakers**
 - **ESP32 STEAMakers + Imagina TdR STEAM**
- En el caso de que estés con el kit de CATEDU **SMARTHOME ESP32** tienes que elegir:
 - **ESP32 STEAMakers**
-

[placa-a-elegir-arduinoblocks.png](#)

ATENCIÓN luego **NO** se puede cambiar. Es decir, un proyecto realizado para un tipo de placa, no se puede cambiar a otro tipo de placa (la razón es simple: las instrucciones cambian)

Luego el nombre y el resto de campos es optativo pero **importante y buena costumbre** rellenarlos, sobre todo si el proyecto lo **compartimos**:

- Descripción

- Componentes
- Comentarios

Área de programación del proyecto

Este es el área sobre el que se trabaja en Arduinoblocks. En esta área arrastraremos y colocaremos los bloques que vamos a utilizar para crear nuestro programa.

En el área de trabajo hay un Zoom (2) para ampliar o reducir la imagen, un icono para centrar (1) y un icono donde podremos borrar los bloques que no utilicemos (3).

2022-04-13 19_02_05-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

Adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

Las principales secciones del área de programación son las siguientes :

2022-04-13 19_03_07-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

Adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

Ver el código

ArduinoBlocks genera el código de Arduino a partir de los bloques. El programa se puede compilar y subir directamente a la placa Arduino gracias a la aplicación **ArduinoBlocks-Connector**, sin embargo si deseamos ver o descargar el código podemos realizarlo desde el área de bloques.

2022-04-13 19_05_37-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

Adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

Siempre, desde un **lenguaje de programación en bloques** podemos obtener su equivalente a **Código de Arduino IDE** (de hecho es lo que hacen los programas), y luego con las funciones de **Código de Arduino IDE** el software lo pasa a **lenguaje máquina** que es la que se graba el Arduino, **pero no al revés** es decir, no existen programas que dado un código máquina o código Arduino IDE lo pasen a bloques gráficos, (igual que no hay programas que lean el código máquina que hay grabado en un Arduino y lo pasen a código



Arduino IDE). Esto no es del todo 100% verdadero pues la Ingeniería inversa en informática trata pues de eso: obtener la fuente aunque sea parcial, pues si obtienes el código legible, puedes alterar lo que quieras.

Cuando compras un programa comercial, te dan el lenguaje máquina ilegible. Mientras que los programas de software libre se publica el código fuente legible para que todo el mundo pueda mejorarlo.

Por ejemplo en la siguiente figura, el programa gráfico **mBlock** que se utiliza en Arduino, mBot, etc... pasa sus instrucciones de lenguaje de programación de bloques parecido a Scratch a lenguaje de **Código de Arduino IDE** y Arduino IDE graba instrucciones binarias de **lenguaje máquina** al Arduino.

[embedded-image-j3gKXHlj.png](#)

¡¡A disfrutar!!

Consejo: Te recomendamos visitar el canal de Youtube de Arduinoblocks

<https://www.youtube.com/c/ArduinoBlocks>

2 ARDUINOBLOCKS

Intermitente

Nuestro primer programa que vamos a hacer es un intermitente en D4

“ **tip** ¿Por qué con D4 y no con otro? Porque D4 tiene un led integrado en la placa, también D0.

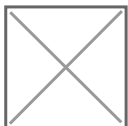


“ **danger**

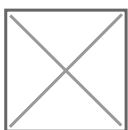
ATENCIÓN

Estos leds van al revés, (configuración [pop-down](#)) luego **para encender el led, D4 tiene que estar a nivel bajo**, y al revés para apagarlo, D4 en nivel alto.

El programa es muy sencillo y lo completamos que se pueda visualizar por la salida puerto serie, por eso al iniciar ponemos el puerto serie a 9600 baudios.



Le damos a **subir** (teniendo el programa Arduinoblocks conector minimizado, eso lo podemos ver enseguida pues detecta en que COM está conectado, en la figura sale COM5) y en **Cónsola** podemos ver el contenido del puerto serie.





2 ARDUINOBLOCKS

¿Qué es Arduino?

¿Qué es Arduino?

Arduino es una tarjeta electrónica que integra básicamente a un microcontrolador y un conjunto de pines de conexión de entradas y salidas que permiten, mediante un determinado programa, interaccionar con el medio físico mediante sensores y actuadores electrónicos. De esta forma podrás crear tus propios proyectos tecnológicos, dotarlos de sensores que detecten magnitudes físicas como luz, calor, fuerza, etc... y en base a esa información, escribiendo un programa, activar otros dispositivos (actuadores) como pequeñas bombillas, ledes, servomotores, pequeños motores DC, relés, etc... Los sensores se conectan a los pines de entrada y los actuadores a los de salida.

¿Sabías que.... ? Uno de los co-creadores de Arduino es Español, de Zaragoza: **David Cuartielles** [+info](#)

¿Qué es un microcontrolador?

Es un circuito integrado que se puede programar, o sea que puede ejecutar las órdenes que tenga almacenadas en su memoria. Tiene las tres funciones principales de un computador: la unidad central de proceso, memoria y entradas y salidas.

Arduino utiliza la marca ATMEL, y el modelo de microcontrolador depende del tipo de tarjeta, por ejemplo la tarjeta Arduino Uno utiliza el micro ATMEL MEGA 328P. Si quieres saber las entrañas de esta placa [aquí](#)

¿Qué se puede hacer con Arduino? ¿Algún ejemplo?

Realmente el límite lo marca tu imaginación pero por dar alguna pista, podrías diseñar un sistema para la apertura y cierre de la puerta de un garaje, hacer un robot móvil que detecte objetos o que siga una línea negra, crear un detector de luz y oscuridad, implementar un termómetro, controlar un cilindro neumático, etc...

En este manual tienes múltiples ejemplos de pequeños proyectos para el aula, aunque Arduino es una herramienta que también se utiliza en el ámbito profesional para monitorización de sensores y automatización a pequeña escala por su flexibilidad, fiabilidad y precio.

¿Qué son las entradas y salidas?



Mediante los conectores de Arduino correspondientes a las entradas y salidas podemos comunicar nuestros programas con el “mundo exterior”. Si queremos leer el valor de la magnitud física medida por un sensor, por ejemplo una LDR que detecta el nivel de luminosidad, lo tendremos que hacer conectando el sensor a uno de los pines de entrada (en este caso analógicas) de la tarjeta.

De esta forma con una simple instrucción de lectura en el programa, podremos obtener el valor de la magnitud física. Si nuestra intención es actuar o “hacer algo” una vez leído el valor del sensor, por ejemplo encender un led si el sensor de luminosidad detecta oscuridad, tendremos que conectar el actuador (en este caso el led) a un pin de salida que proporcionará la corriente necesaria para activarlo.

En Arduino las entradas pueden ser analógicas o digitales y las salidas sólo digitales. Cada pin digital tiene doble función entrada o salida. En la zona de configuración del programa hay que indicar explícitamente mediante una instrucción cuál es función desempeña un determinado pin.

¿Dónde se conectan los sensores? ¿A las entradas analógicas o digitales?

La mayoría de sensores miden señales analógicas y proporcionan una variación de voltaje dentro de un rango (normalmente de 0 a +5V) dependiendo de lo que varíe la magnitud física medida. Muchos sensores son resistivos a algo (luz, temperatura, humedad,...), es decir que varían su resistencia eléctrica con la magnitud física, pero mediante un sencillo montaje de divisor de tensión conseguimos una variación de voltaje apta para Arduino. Estos montajes los veremos en las prácticas.

Veamos este ejemplo:

El sensor LDR es una resistencia que cambia según la intensidad de la luz. La estrategia es colocar el LDR en [un divisor de tensión](#) con otra resistencia de valor parecido al promedio del que da el LDR (por ej 10k), y el valor del punto de unión proporciona una tensión entre 0 y 5V. Como es una señal analógica, la conectamos a una entrada analógica (en la figura al A6)

[embedded-image-YfsLGvOe.jpegldrana.png](#)

Una vez realizadas las conexiones, si midieramos la salida del sensor con un voltímetro nos daría un valor decimal, por ejemplo un nivel de luz “intermedio” (rango de 0 a 5V) de un sensor de luz podría dar 3,3 voltios. Este tipo de información el microcontrolador **no la entiende tal cual**, sólo es capaz de interpretar números binarios (“0” ó “1”) por lo que para traducir los valores analógicos dispone internamente de un conversor analógico - digital que hará la conversión entre los dos sistemas, de forma que podremos tener valores discretos de la medida de los sensores analógicos. En el Arduino **las entradas analógicas** leen valores analógicos entre 0V y la alimentación (normalmente 5V) y los convierten en números entre **0 y 1023** (porque lo codifica en 10 dígitos binarios proporcionan $2^{10} = 1024$ combinaciones).



Por ejemplo, si la entrada analógica lee un valor de 3,3V y la fuente de alimentación es 5V, la señal analógica que lee Arduino, haciendo una regla de 3, tiene un valor de $3,3 * 1023 / 5 = 675,18 = 675$

Mapeo

Para convertir estos valores 0 -1023 a valores más legibles, por ejemplo 0 - 100 para representarlo en % o 0-5 para que represente la medida en voltios ... veremos en programación la función **mapear**

La mayoría de los sensores nos lo venden ya preparados montados en una pequeña placa electrónica y con circuitos integrados auxiliares para no tener que estar haciendo divisores de tensión. Pueden tener salida analógica o **salida digital**, que en este caso lo tenemos que conectar a cualquier entrada digital D0 hasta D13.

Veamos el mismo ejemplo del LDR: Podemos comprar este módulo:

[ldrdig1.png](#)

Estos módulos proporcionan 3 pines: dos que son la alimentación, (0V, GND o -) y (+5V V+o Vcc) y el pin que proporciona la lectura (Vout o D0 o I/O). En el caso de que sea un sensor que mida una magnitud analógica como en este caso la luz, suelen proporcionar un potenciómetro para determinar qué luminosidad se considera un 0 o un 1.

¿Hay sensores digitales que no estén en una placa electrónica?

Las entradas digitales sin una placa electrónica son cuando las señales a leer son valores discretos. Por ejemplo queremos poner un pulsador o un interruptor que encienda un led. Hacemos un montaje que cuando se pulse, entren 5 voltios en el pin digital de entrada y cuando no se pulse que “entren” 0 voltios. De esta manera la lectura del pin digital de entrada será “HIGH” con 5 voltios o “LOW” con 0 voltios.

Veremos más adelante que un interruptor no es un simple cable que conectamos a +5V o a 0V pues ¿Qué valor lee Arduino mientras levantamos el cable de un sitio a otro?, para ello veremos [configuraciones Pull-up o Pull-down](#) que se repiten en muchos sensores.

¿Qué son las salidas digitales etiquetadas con PWM (~)?

Son salidas digitales que **simulan una salida analógica**. Las siglas significan *Modulación por Ancho de Pulso (Pulse Width Modulation)* es decir, proporcionan una onda cuadrada con un nivel alto (+5V) de “cierta” duración.

[pinesPWM.png](#)

Los valores PWM que podemos proporcionar pueden ir desde **0 a 255**.

- Un valor bajo es una señal cuadrada donde el pulso es pequeño comparado con el periodo (ver figura primera señal) por lo tanto el Valor medio es bajo.
- Un valor mitad, es decir $255/2 = 127$ o 128 , corresponde a una señal cuadrada perfecta (ver figura segunda señal) por lo que el Valor medio es la mitad, si 5V es la tensión de alimentación, sería 2,5V.
- Un valor alto es una señal cuadrada donde el pulso es grande comparado con el periodo (ver figura la tercera señal) por lo tanto el Valor medio es alto
- Un valor 0 corresponde a un 0V analógico y sería una señal sin pulso
- Un valor 255 corresponde a la máxima tensión (la de alimentación (normalmente 5V)).
- La frecuencia es de 490Hz para los pines 3, 9, 10, 11 y de 980Hz para los pines 5 y 6 en un Arduino UNO o Nano

[pwmsenal.png](#)

De esta manera podemos simular señales analógicas, esto es muy útil para activar servomotores y llevarlos a una posición determinada o variar la luminosidad de un led o en los motores de los robots que vayan más deprisa o más despacio

¿Puedo accionar motores DC con Arduino?

Si son motores muy, muy pequeños sí sería posible aunque no es recomendable, pueden dañar la placa. Los motores necesitan un consumo alto de corriente, sobre todo si tienen que mover cierta carga, por lo que se recomienda o bien utilizar una tarjeta Shield o extensión de Arduino que dispone de circuitería apta para proporcionar dicha corriente (transistores).

- En el [curso Arduino con código](#) utiliza una Shield llamada *Edubásica* que dispone de un [transistor](#) y un [circuito integrado LM293](#) para realizar esta función, además de otras ventajas para el aprendizaje de Arduino.
- En el curso [Rover con Arduino](#) utiliza Shield motor para NodeMCU

2 ARDUINOBLOCKS

Software del Arduino

Las placas electrónicas educativas (Echidna, Micro:bit, Arduino, ESP32 ...) se programan mediante **varias opciones** :

https://docs.google.com/presentation/d/e/2PACX-1vQb1Dv9wN9QK-F6V7yvwDoyzquqwWIGvlyVjr83Yk56kAoYD7bXlnYDm_tCQkeAgg/pubembed?start=false&loop=false&delayms=3000

OPCIÓN LENGUAJE GRÁFICO POR BLOQUES

Recomendado para primaria. Tenemos muchas posibilidades de lenguajes gráficos. Destacamos dos:

- **ECHIDNASCRATCH**
 - **Específico para Echidna e integra la IA [CURSO DE ECHIDNA](#)**
- **MBLOCK** Basado en Scratch. Aunque es un programa especializado en el robot comercial mBot, (basado en Arduino), el mismo programa está adaptado para programar Arduino.
 - **[CURSO ARDUINO CON MBLOCK](#)** se utiliza Arduino y placa Protoboard
 - **[CURSO DE ECHIDNA](#)** se utiliza la Shield Echidnam y EchidnaBlack
 - **[CURSO DE MBOT](#)** se utiliza el robot mBot
- **STEAMAKERBLOCKS (antes ARDUINOBLOCKS)** se trabaja online, muy visual y muy amigable. Está adaptado tanto para trabajar tanto Arduino como muchas placas controladoras y en el aula. Podemos verlo en los siguientes cursos:
 - **[CURSO ROVER CON ARDUINO](#)** aunque no se utiliza un Arduino, sino un NodeMCU pero la programación es exactamente igual
 - **[CURSO DE ARDUINO CON ARDUINOBLOCKS](#)** donde se utiliza el Arduino con una placa protoboard
 - **[CURSO ARDUINOBLOCKS EN EL AULA](#)** donde se utiliza la Shield TDR-STEAM



- **CURSO ESP32 EN EL AULA** donde también utiliza la Shiedl TDR Steam pero la placa no es un Arduino, sino ESP32, la programación es exactamente igual.
- **Microbloks** <https://microblocks.fun/> placas: Arduino, Microbit, ESP32, RaspberryPico,
[ver minitutorial](#)

Otros softwares para programar con bloques

- **Snap4Arduino** <https://snap4arduino.rocks/run/> Online, libre... ver [compartiva vs mBlock](#)
- **S4A** <https://s4a.cat/>

EN VIVO ¿Qué es eso?

Existe una posibilidad de utilizar la placa "en vivo" frente a "cargar" el programa en la placa. Es decir, interactuando con el ordenador. El programa está en el PC. En la placa hay un **firmware** que le dice que este a las órdenes del PC. De esta manera podemos por ejemplo:

- Enviar órdenes desde el ordenador a la placa.

Por ejemplo que al pulsar la tecla espacio que se encienda el led D13

- Enviar información desde la placa al ordenador

Por ejemplo que muestre por pantalla nos muestre la cantidad de luz, que registra el sensor LDR, etc...

Que nosotros sepamos, estos programas permiten la programación en vivo :

- **mBlock** placas: Arduino, Microbit, Raspberry Pi, ... robots de Makeblock: mBot, Cyberpi...

- **EchidnaScratch** [CURSO DE ECHIDNA](#)

- **Microblocks**

VENTAJAS LA PROGRAMACIÓN EN VIVO PERMITE MUCHO JUEGO Y POSIBILIDADES A LA HORA DE ELABORAR PROYECTOS

INCONVENIENTES: Necesitas el ordenador encendido y conectado al robot.

TAMBIÉN ES EN VIVO PERO

Hay otros softwares que técnicamente trabajan en vivo, es decir, que el programa se ejecuta desde el ordenador, no se ejecuta en la placa, son :

- **Snap4Arduino** para placas Arduino
- **Picobriks** blocks para Picobrick board

Pero no permiten trabajar utilizando los elementos del ordenador (teclado, webcam, pantalla, sprite o objetos,,)



Es **importante** que entiendas que para trabajar en vivo, la placa tiene que tener cargado un "**firmware**" para que interactúe con el ordenador.

P: ¿Qué es eso de "firmware"?

R: No es más que un software que se graba en los chips de la placa.

P ¿Y por qué se llama así, y no se llama software o programa y en paz?

R: Digamos que como se graba en los chips, es un medio camino entre software y hardware, para diferenciarlo del software habitual.

EN CARGA ¿Qué es eso?

Simplemente el programa que estas haciendo **se carga** en la placa

VENTAJAS: Eres independiente del ordenador, tu robot funciona independiente

DESVENTAJAS Pierdes todas las posibilidades de utilizar los recursos de un ordenador, teclado, pantalla, webcam, altavoces...

Es **importante** que si **cargas** tu programa en la placa, **pierdes** lo que había antes
Es decir, si quieres volver a trabajar EN VIVO tienes que cargar el firmware correspondiente.

OPCIÓN LENGUAJE POR CÓDIGO

Recomendable a partir de secundaria.

LENGUAJE POR CÓDIGO ARDUINO IDE

Es un lenguaje basado en Wiring y permite la programación del Arduino en un entorno de desarrollo (basado en Processing). El programa se llama **ARDUINO IDE** y se puede descargar desde la página oficial de Arduino: <https://www.arduino.cc/en/software>.

Hay otra posibilidad que es utilizarlo online, con la ventaja de tener tus proyectos "en la nube" y no depender del equipo. OJO, TIENES QUE TENER INSTALADO EL SOFTWARE

CREATE AGENT

<https://create.arduino.cc/getting-started/plugin/welcome>

arduinoide.png

En los cursos de CATEDU se ha utilizado el lenguaje por código empezando desde cero en:



- **CURSO ARDUINO CON CÓDIGO** donde se trabaja con el Arduino y con diferentes sensores y actuadores, con o sin placa Shield Edubásica.
- **CURSO DE DOMOTICA CON ARDUINO** donde se realiza una maqueta de una casa controlada con domótica. También el curso ofrece la versión de hacer la misma maqueta utilizando lenguaje gráfico por bloques.

Recomendamos estas hojas resumen si vas a trabajar con código:

- En Español: [enlaceDrive](#), [enlaceGithub](#)
- En Inglés: [enlaceDrive](#), [enlaceGithub](#), [enlaceSpakrfun](#)

LENGUAJE POR CÓDIGO PYTHON

Es un lenguaje más amigable que ArduinoIDE y tiene muchos campos de aplicación aparte de la robótica

- Arduino tiene su propia versión para trabajar con sus placas compatibles: **Arduino Lab for Micropython**
 - [Ver curso Arduino ALVIK](#)
- Un compilador universal **Thonny**
 - [Ver curso Pico-briks](#)

VENTAJAS E INCONVENIENTES LENGUAJE GRÁFICO POR BLOQUES vs CÓDIGO

El lenguaje gráfico por bloques es un lenguaje **sencillo de utilizar**, nos evita tener en cuenta muchas **librerías** y **cálculos**.

Otra ventaja, es que el lenguaje por bloques es el único que permite programación "en vivo"

Por ejemplo, la instrucción leer valor distancia el sensor ultrasonidos, mediante programación por bloques es

[leer-us.jpg](#)

mientras que en código es

```
double distancia;
```

```

double fnc_ultrasonic_distance(int _t, int _e){
  unsigned long dur=0;
  digitalWrite(_t, LOW);
  delayMicroseconds(5);
  digitalWrite(_t, HIGH);
  delayMicroseconds(10);
  digitalWrite(_t, LOW);
  dur = pulseIn(_e, HIGH, 18000);
  // devuelve cuanto tarda el pulso alto en microseg; 18000 es el tiempo a esperar limite
  if(dur==0)return 999.0;
  return (dur/57);
  // la velocidad del sonido es 344m/s = 34400 cm/seg = 0,0344 cm/microseg
  // como v=e/t luego e = v*t y como cuenta la ida y la vuelta distancia = v*t/2
  // luego distancia = 0,0344/2 * dur = dur/57
}

void setup()
{
  pinMode(6, OUTPUT);
  pinMode(5, INPUT);
}

void loop()
{
  distancia = fnc_ultrasonic_distance(6,5);
}

```

Como se puede ver en **código**, tiene que calcular la distancia haciendo cálculos del tiempo de rebote del eco, mientras que el **gráfico** es sumamente sencillo y se centra en el objetivo del algoritmo a crear, no en lo accesorio. Esto hace que un lenguaje gráfico por bloques se puede aplicar **desde los 8 años**.

Por otra parte, el lenguaje **código tiene todo el potencial**, es decir, no todo está en los lenguajes gráficos. Si se quiere cosas más avanzadas, hay que recurrir al código.

Un lenguaje **gráfico** se convierte en lenguaje **código**, pero **al revés no se puede**, debido a que el código es más depurado y no tiene la información necesaria para volver a su origen en bloques, ya lo has visto con el anterior ejemplo, el código tiene más información.

¿No te lo crees? Haz la prueba, métete en <https://www.tinkercad.com/> crea un programa con bloques, dale a la pestaña de código y te aparecerá una advertencia que perderás el programa con bloques ! **no puedes volver atrás!**

Curiosamente, tiene una opción *bloques+código* que traduce cada bloque con un código, es decir, traduce cada bloque sin perder información, sólo de esa manera se puede pasar de bloques a código y viceversa.

2024-09-23 10_51_49-Circuit design PIN0-CORE-CREW - Tinkercad.png

Hay [herramientas para pasar de bloques a código](#) pero no al revés

bloques-codigo.png

Imagen Federico Coca [Notas sobre ESP32 STEAMakers](#) CC-BY-SA

El lenguaje **código** se traduce en lenguaje **máquina** (ceros y unos) entendible para el microprocesador, **pero al revés no se puede**.

En [este vídeo](#), en mi opinión se olvida de mBlock, Snap4Arduino, S4A pero puedes ver un vistazo de los diferentes editores

OPCIÓN SIMULACIÓN

Incluimos dentro del apartado de Software los diferentes programas que hay para simular placas electrónicas como Arduino, ESP32, Micro:bit etc...


Tinkercad

Esta herramienta <https://www.tinkercad.com> aparece en el [Curso Arduino con código](#) en la práctica [Comunicación entre dos Arduinos](#), pero también es una plataforma que sirve para hacer

los diseños de elementos 3D, ver curso [Impresión 3D con Tinkercad](#)

Tiene la ventaja que es aplicación **online**, muy **visual** y buscan un reflejo de la práctica **real**, además de estar la herramienta **adaptada al aula** (gestión de alumnos y proyectos). Como desventajas podemos decir que no tiene mucha variedad de componentes electrónicos y la simulación es algo lenta.

<https://www.youtube.com/embed/pXEv0wxW9Jo?rel=0>

Además permite embeber, **pruébalo !!!**  Pulsa **Iniciar simulación** y luego pulsa el **botón A** de cualquiera de los dos micro:bits

<https://www.tinkercad.com/embed/8vBL2E8wWDx?editbtn=1>

Wokwi

Si Tinkercad se queda corto, puedes probar esta plataforma <https://wokwi.com/> con muchas posibilidades. Es **online** y puede trabajar con **multitud de placas**: ArduinoUno, ESP32, Raspberry,,,,

Como única desventaja que encontramos, es que echamos de menos la realidad de Tinkercad, por ejemplo no puedes poner una placa protoboard para realizar las conexiones, pero a cambio se gana simplicidad de cableado.

[2022-12-05 10_14_42-Knob.ino - Wokwi Arduino and ESP32 Simulator.png](#)

UnoArduSim

Es una **aplicación local**. [UnoArduSim](#) además es una aplicación portable fácil de instalar y con los elementos de leds, motores servos ya preparados, ideal para ejemplos sencillos y para examinar señales y no depender de Internet, pero no es tan versátil.

https://www.youtube.com/embed/WLJ_l4uGjXg?rel=0

SIMULIDE

En <https://simulide.com/> podemos encontrar un programa local de software libre genérico para electrónica, incluido Arduino. En esta captura se puede ver uno de los ejemplos que incorpora el programa:

[2025-02-07 16_09_42-DesktopWindowXamlSource.png](#)

OPCIÓN SÓLO DIBUJAR

- **TinkerCad** es un buen programa para dibujar los planos
 - permite también la simulación
 - permite embeber y compartir
 - no tiene muchos componentes
- **SimulIDE** es software libre. Es un programa portable.
 - Tiene muchos componentes
 - permite también la simulación
 - le faltan algunos sensores, pero van incorporando
- **Fritzing** es un clásico. Es un programa portable.
 - Tiene muchos componentes
 - no es gratis, hay que pagar 8€
- **Circuit canvas**
 - puede compartir [por ejemplo](#)
 - tiene buenos tutoriales sobre electrónica
 - todo en inglés

2 ARDUINOBLOCKS

Sensores

Esta sección es una visión rápida de las posibles **entradas** de los robots.

NO LEAS TODOS SINO LOS QUE TIENE TU ROBOT

Un poco de teoría...

Cualquier sistema de control podríamos decir que funciona de una manera similar a un ser humano, salvando las distancias. Nosotros recibimos la información del mundo exterior gracias a nuestros sentidos (oído, olfato, gusto, vista y tacto), nuestro cerebro procesa esa información y a través de nuestros músculos o de nuestra voz realizamos diferentes acciones. Pues lo mismo sucede con los sistemas de control, reciben información del exterior gracias a los diferentes **SENSORES**, procesan esa información en sus PLACAS CONTROLADORAS (sus cerebros) tales como Arduino y dan una respuesta utilizando sus diferentes **ACTUADORES**.

[2022-04-13 19_23_12-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Un sensor es un objeto capaz de detectar magnitudes físicas o químicas y transformarlas en variables eléctricas. Los sensores o periféricos de entrada nos permiten obtener información del mundo real para utilizarla desde el programa de Arduino.

En la actualidad la cantidad de sensores disponibles es tan extensa como las variables que queramos medir, desde sensores de temperatura, humedad, luminosidad,... hasta acelerómetros, giroscopios, GPS,... pasando por detectores de gases, de pulsos cardiacos, sensores de efecto HALL,...

Tipos de sensores

- **DIGITAL**: un sensor digital sólo tiene dos estados: activado/desactivado, ON/OFF, 1/0, Alto/Bajo, ... En este caso conectaremos el sensor a una de las entradas digitales de Arduino para leer el estado.

Ejemplo: un pulsador es un tipo de sensor sencillo que sólo nos da dos estados, "pulsado o no pulsado". Conectado a la placa Arduino debe generar 0v en reposo y 5v al pulsarlo. De



esta forma desde el programa de Arduino podremos leer el estado del botón.

[2022-04-13 19_25_30-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

- **ANALÓGICO:** el sensor nos puede dar un rango de valores, normalmente se traduce en un valor de tensión o de corriente variable en función de la señal captada al sensor. En este caso conectaremos el sensor a una de las entradas analógicas de Arduino (A0,..., A5). El rango de entrada será una tensión entre 0v (GND) y 5v.

Ejemplo: Una fotorresistencia es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. Su valor varía entre 0 y 5 v. la cantidad de valores que pueden leer las entradas analógicas de Arduino son de 10 bits es decir 1024 valores. De tal modo que 0 = 0 v. y 1023 = 5V.

[2022-04-13 19_26_06-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

- **DATOS:** el sensor ofrece su información a través de una interfaz de comunicación. La forma de comunicación puede ser por sistemas estándar como **I2C** o SPI o algunos sensores usan su propio protocolo para codificar la información y debemos realizar desde el software la decodificación correcta para interpretar los datos del sensor (normalmente los desarrolladores de este tipo de sensores ofrecen una librería software para Arduino que hace todo el trabajo).

Ejemplo: el sensor DHT11. Por un solo pin envía los datos de temperatura y humedad.

[2022-04-13 19_27_43-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Sensores modulares.

En la actualidad existen infinidad de sensores que los fabricantes presentan en forma modular. Esto hace que su conexión y utilización sea mucho más sencilla que la tradicional, olvidándonos de resistencias, polaridades, cableados,... para su correcto funcionamiento.

Sensor pulsador

Es un sensor digital, que presenta dos estados; cuando se presiona el botón, emite una señal de bajo (0V), cuando suelta el botón, emite una señal de bajo alto (5V). [Datasheet](#)

Un ejemplo de uso

- [en el robot mClon](#)

[2022-04-13 21_58_38-Modulo pulsador con tapa, 6uds.pngimage-1649922433118.png](#)

Otra manera más "barata" de sustituir este módulo pulsador es poner un pulsador normal y una resistencia ($\pm 10k$), al pulsar se produce una entrada en el Arduino, hay dos configuraciones, que al pulsar se emita un 0 lógico (configuración **Pull up**) o que al pulsar emita un 1 lógico (configuración **Pull down**) [¿Por qué hay que poner una resistencia?](#)

[image-1650097141548.pngimage-1650097157889.png](#)

Lo "normal" es que al pulsar se emita un '1' configuración **Pull down**, pero hay pulsadores que funcionan **Pull up** y los llaman **lógica invertida**, por eso en la programación por bloques podemos encontrar esto:

[logicainvertida.png](#)

Sensor Táctil Capacitivo.

Este pequeño sensor puede "sentir" a las personas y el tacto y la retroalimentación de metales a un nivel de voltaje alto / bajo. Incluso aislado por alguna tela y papel, todavía puede sentir el tacto. Su sensibilidad disminuye a medida que la capa de aislamiento se hace más gruesa. En nuestra opinión lo preferimos frente al *Sensor pulsador* pues es muy económico, duradero y fiable.

Un ejemplo de uso en

- [Disparo láser en Arduino con ArduinoBlocks](#)
- [Apertura de puerta en Domotica con Arduino](#)

[2022-04-13 21_59_29-Interruptor táctil TTP223B.png](#)

Potenciómetro y joystick

Un potenciómetro es una resistencia variable, es decir, cambia de valor mecánicamente, lo tenemos en multitud de dispositivos. El **joystick** es internamente dos potenciómetros con un pulsador integrado en un solo mando.

[potenciometro-joystick.png2023-11-24 18_45_14-Quiero una - EchidnaShield.png](#)



Este sensor es **analógico**, su salida puede ser cualquier valor entre Vcc y GND (si está en divisor de tensión como en la placa Edubásica no llega a esos valores extremos), por lo tanto hay que conectarlo a una entrada analógica de Arduino y como cualquier entrada analógica, proporcionará valores entre 0 y 1023.

Ejemplos de uso:

- Arduino con código: [Mapeo del potenciómetro](#)
- Arduino con código: [Regular la luz con potenciómetro](#)
- [Arduinoblocks en el aula](#)
- En [Arduino con Echidna](#), con joystick
- [Domótica con Arduino con joystick](#)

Sensor Fococélula LDR.

El uso de fotocélulas es muy común en nuestras vidas, las encontramos en el encendido automático de farolas, apertura de puertas,... La fotocélula es un semiconductor. Es ampliamente utilizado en campos de interruptores de control automático como cámaras, luces solares de jardín, lámparas de césped, detectores de dinero, relojes de cuarzo, tazas de música, cajas de regalo, mini luces nocturnas, interruptores de control de luz y sonido, etc. Es un sensor analógico dando valores entre 0 y 5V y como entrada analógica de un Arduino se traduce en un rango de 0 a 1023 valores.

Un ejemplos de uso :

- [el interruptor crepuscular del curso Arduino con ArduinoBlocks](#)
- [Medir la luz en Rover con Arduino](#)
- [Medir la oscuridad en Arduino con mBlock](#)
- [Hinchar un balón en Arduino con mBlock](#)

Una manera más económica de montar este sensor es utilizar una resistencia y un LDR:

- El LDR cuando más oscuridad, más resistencia
- En una configuración **PULL DOWN**, cuanto **más** luz, la resistencia del LDR baja, por lo tanto **más** tensión en A6

[embedded-image-YfsLGvOe.jpeg](#)

[2022-04-14 08_04_14-3.5 Cableado sensores _ Librería CATEDU.png](#)

Los módulos LDR que se venden suelen esta configuración Pull down, es decir, cuanto **más** luz, **más** tensión:

[image-1650370178627.png](#)

La instrucción con Arduinoblocks ya cuenta con esta configuración Pull downUp de que cuando **más** luz, **más** valor tiene la entrada analógica.

[2022-04-19 14_05_05-ArduinoBlocks.png](#)

Hay módulos LDR ya montados, que tienen componentes **activos** es decir, llevan circuitos electrónicos, transistores que amplifican etc... y dan la salida **digital** con un potenciómetro para definir el rango de luz que cambia de estado lógico. Puedes ver en la figura que tiene una salida digital **D0**.

[sensorluzarduinoigital.jpg](#)

O hay [algunos que tienen 4 pines](#) como en la figura que ofrecen las dos cosas: salida analógica **A0** y digital **D0**.

[sensorluzarduino.jpg](#)

Nosotros aconsejamos el divisor de tensión por tres razones: más barato, no implica gran circuitería y es visible su funcionamiento frente a estos encapsulados.

Sensor de Ultrasonidos.

Es un sensor digital de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 350 cm. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno.

No es un sensor preciso, con una ligera inclinación de la superficie ya da lecturas erróneas pero es muy barato

El más común es el **HC-SR04** que tiene 4 pines de conexión: **VCC Trig** (Disparo del ultrasonido)

Echo (Recepción del ultrasonido) y **GND** aunque en algunos modelos como el de [ElecFreaks](#) tiene 3 pines. Integra Trig y Echo en uno sólo.

La distancia se calcula con esta fórmula:

Distancia en cm = {(Tiempo en segundos entre Trig y el Echo) * (V.Sonido 34000 en cm/s)} / 2

Si programas en código, tienes que utilizar la fórmula anterior, previamente tienes que programar el cálculo del tiempo entre una emisión de un pulso en Trg y la respuesta en Echo.

Si utilizas la programación en bloques, no es necesario, seguro que hay un bloque que lo hace todo por ti ☐

[2022-04-14 08_10_43-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

[2022-04-14 08_11_21-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Ejemplos de uso:

- [Alarma en Domótica con Arduino](#)
- [Piano invisible en Arduino con ArduinoBlocks,](#)
- [Sensor parking en Arduino con ArduinoBlocks](#)
- [Piano invisible en Arduino con mBlock](#)
- [Sensor parking en Arduino con mBlock](#)
- [Sensor de distancia de ultrasonidos con Picobricks](#)

Sensor DHT11 (Temperatura y Humedad).

Este sensor de temperatura y humedad **DHT11** nos permite [determinar las zonas de confort](#) para un rango de temperaturas entre 0°C y 50°C con un error de $\pm 2^\circ\text{C}$ y un rango de humedad entre 20 y 90 % $\pm 5\%$. Una salida digital para dos variables cómo lo hace? Tiene dentro un pequeño microprocesador que lanza por el bit de datos 40 bits en serie, los 16 primeros son la humedad (en BCD) y los 16 restantes es la temperatura (en BCD) los 8 restantes son de comprobación

Checksum (en binario) como la letra del DNI. Por ejemplo **0100 0111 0000 0011 0001 1001 0000 0000 0001 1000** es **0100 0111 0000 0011** = 47.03% de humedad y **0001 1001 0000 0000** = 19.00°C y la comprobación es la suma de 4+7+0+3+1+9+0+0=24=**11000**

Ejemplos de uso:

- [Medir H y T con Blink en Rover con Arduino](#)
- [Estación meteorológica Arduino con Arduinoblocks](#)
- [Arduinoblocks en el aula](#)
- [SMART HOME con Micro:bit](#)

2022-04-13 19_27_43-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

No es un sensor con gran sensibilidad, pero para propósitos educativos cumple sus funciones. Por dentro tiene una resistencia NTC que decremента su resistencia si aumenta la temperatura. Hay otros que van al revés, los PTC. Tanto los NTC como los PTC se llaman **thermistores**. Para la humedad, mide la capacidad de un condensador que es sensible a la humedad, o sea, un **sensor capacitivo**.

Tenemos dos opciones comerciales: **Encapsulado** que lo tienes preparado para conectar la alimentación y leer por el pin de datos, o **sin encapsular**, que hay que colocar una resistencia de aproximadamente 10k entre Vcc y Data

[dht11-pines.png](#)

Ejemplo de uso de un DHT11 sin encapsular:

[dht11sinencapsular.png](#)

Fuente Luis LLamas CC-BY-NC-SA <https://www.luisllamas.es/arduino-dht11-dht22/>

Si queremos mejorar la sensibilidad, podemos utilizar el **DHT22** que es igual pero de color blanco y más caro. Si lo que queremos es sólo la temperatura es mejor utilizar el **LM35D** que tiene un rango de temperaturas desde 0°C a 100°C con una sensibilidad de 2mV/°C

[lm35.jpg](#)

Es un sensor bastante mediocre, si necesitas una precisión el doble, te recomendamos el DHT22 que funciona exactamente igual pero es de color blanco y más caro ~3€. Ver <https://www.luisllamas.es/arduino-dht11-dht22/>

Sensor IR

Es un sensor para distancias cortas hasta 2cm y no da la distancia, simplemente si hay o no hay obstáculo, pero son muy baratos, unos 0.30€. [Aquí tienes un ejemplo de evita obstáculos en un rover marciano con Raspberry](#) Para saber más te recomendamos [esta página de Luis Llamas](#)

<https://sketchfab.com/models/6ad4f3afb83940fea95cd3846aa68a18/embed>

[IR Sensor Module for Arduino Projects | 3D Model](#) by [Veer AI](#) on [Sketchfab](#)

Sensor llama

Este sensor de llama se puede utilizar para detectar fuego u otras luces cuya longitud de onda se encuentra entre 760 nm ~ 1100nm.

Un ejemplo de su uso:

- [Alarma por fuego en Domótica con Arduino](#)

2022-04-14 08_06_56-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

2022-04-14 08_07_38-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

Sensor de Gas (MQ2).



Detecta gases inflamables : GLP, I-butano, propano, metano, alcohol, hidrógeno, humo... con más sensibilidad en algunos que en otros. Siempre detecta el conjunto. Son usados en electrónica de consumo y mercados industriales.

- **Sensibilidad** Tiene alta sensibilidad y se puede ajustar girando el potenciómetro.
- **Tiempo de respuesta:** Internamente posee un calentador para aumentar su temperatura y que estos gases reaccionen con la resistencia interna que tiene, por lo tanto tardan algo en responder la primera vez que se conectan, incluso horas en algunos modelos. Una vez calentados son rápidos en la respuesta.
- **Tipo de salida:** Analógico pero si tiene 4 pines como el de la figura, incorpora un pin digital.
- Ejemplos de uso:
 - [Smart Home Microbit](#)
 - [Smart Home ESP32](#)

[2022-04-14 11_05_36-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Sensor de humedad de suelo.

“ Un higrómetro de suelo FC-28 es un sensor que mide la humedad del suelo. Son ampliamente empleados en sistemas automáticos de riego para detectar cuando es necesario activar el sistema de bombeo. El FC-28 es un sensor sencillo que mide la humedad del suelo por la variación de su conductividad. No tiene la precisión suficiente para realizar una medición absoluta de la humedad del suelo, pero tampoco es necesario para controlar un sistema de riego. Los valores obtenidos van desde 0 sumergido en agua, a 1023 en el aire (o en un suelo muy seco). Un suelo ligeramente húmedo daría valores típicos de 600-700. Un suelo seco tendrá valores de 800-1023.

Luis Llamas CC-NC-BY-SA <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>

[Se puede utilizar este sensor para hacer un dispositivo de riego automático, puede detectar si las plantas “tienen sed” y evitar que se marchiten.](#)

La corriente de trabajo del sensor es menor de 20mA. El voltaje de salida es de 0V (en el aire) a 2,3V (totalmente sumergido en agua).

- [Smart Agriculture Kit micro:bit](#)



2022-04-14 08_15_34-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

Sensor de humedad.

Este sensor **analógico** está diseñado para identificar y detectar la presencia de agua y su cantidad. Puede servir para detectar el nivel de agua, para disparar una alarma en caso de una fuga de agua, también para hacer un limpiapalabrisas automático.... puedes ver un ejemplo de uso en :

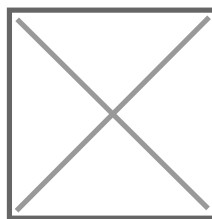
- Curso de [Domótica con Arduino](#)
- [Smart Agriculture Kit micro:bit](#)

Mide el volumen de agua caída a través de una serie de rastros de cables paralelos expuestos.

2022-04-14 08_19_22-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png
2025-11-08 18_13_20-.png

Sensor de efecto Hall.

Este es un sensor de inducción magnética. Detecta los materiales magnéticos dentro de un rango de detección de hasta 3 cm. El rango de detección y la fuerza del campo magnético son proporcionales. La salida es digital.

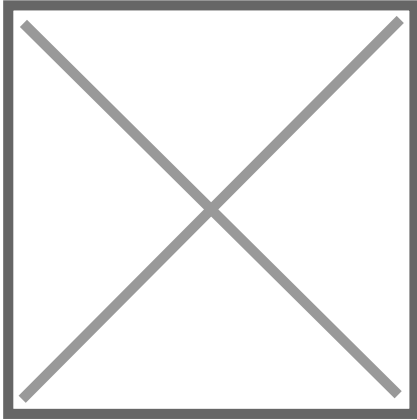


2022-04-14 10_07_05-1.3 Conexiones _ Librería CATEDU.png

Sensor Hall.

Edwin Helber Hall De Desconocido - [Popular Science Monthly Volume 64, Dominio público](#)

[Edwin Helbert Hall](#) descubrió en 1879 que en presencia de un campo magnético, un conductor que conduzca una corriente se le producía un campo eléctrico porque las cargas eléctricas se desviaban de su trayectoria principal, nuestro sensor simplemente mide ese campo eléctrico:



De [Luis Llamas](#) CC-BY-NC

El sensor tiene un led de color rojo que indica que hay una lectura de campo magnético. Un ejemplo de uso lo puedes ver aquí: [medir rocas magnéticas con el Rover con Arduino](#)

Sensor inclinación

Este sensor funciona al hacerle vibrar, emitiendo una señal digital de todo o nada. El módulo del sensor viene provisto de un potenciómetro para poder regularlo.

[2022-04-14 11_01_13-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Sensor de golpe

Es un sensor digital que al ser golpeado este sensor envía una señal momentánea.

[2022-04-14 11_02_09-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Sensor de pulso cardíaco.

Este módulo utiliza un LED infrarrojo (IR) ultrabrillante y un fototransistor para detectar el pulso en el dedo. Principio de funcionamiento: Se debe colocar el dedo entre el LED infrarrojo ultrabrillante (parte superior) mientras que el fototransistor, que queda en el otro lado, recoge la cantidad de luz transmitida. La resistencia del fototransistor variará levemente a medida que la sangre pase a través de su dedo.

[2022-04-14 11_03_47-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Sensor de Alcohol.

Este sensor de gas analógico MQ-3 es adecuado para detectar alcohol. Se puede usar en un analizador de aliento. También tiene una alta sensibilidad al alcohol y baja sensibilidad a la bencina

(éter de petróleo). La sensibilidad se puede ajustar con el potenciómetro.

[2022-04-14 11_04_42-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Sensor de CO2

Hay sensores que utilizan **el protocolo I2C**, este protocolo permite conexiones serie y pueden compartir el mismo cable pues cada elemento tiene una dirección diferente. Esto lo veremos en el Display LCD. Se identifican por los pines SDA y SCL

CCS811-KEYSTUDUUDIO.png

Resistencia Flex

Es una resistencia que cuanto más se dobla más resistencia ofrece, desde 25k hasta 125k

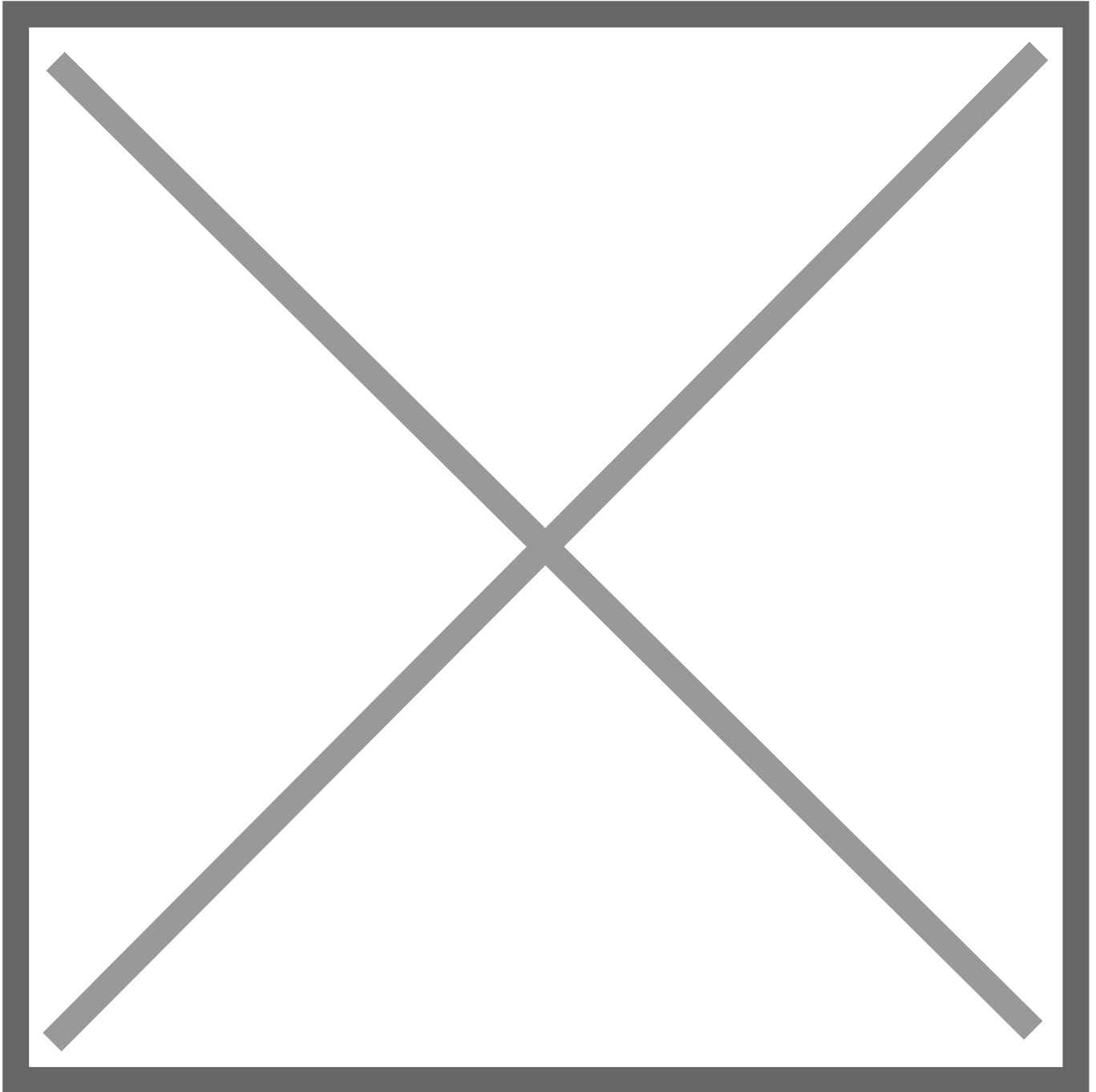
[Datasheet Sparkfun](#)

Para utilizar esta resistencia haremos un **DIVISOR DE TENSION** que consistirá en poner dos resistencias en serie y repartirá la tensión total entre 0V y 5V en las dos resistencias, **el punto medio** será un punto que tendrá una tensión variable en función de las dos resistencias, como la es variable, esa tensión es variable y ya tenemos la entrada **analógica**:



Es decir:

- La resistencia entre masa GND del ARDUINO (cable negro) y un punto en la placa protoboard
- ese punto medio conectarlo a una entrada analógica, por ejemplo A0 (cable amarillo)
- Una resistencia de valor parecida a la Flex de decenas de K entre ese punto y +5V (cable rojo en la foto)



Este sensor tiene posibilidades para usarlo en "ropa inteligente".

Sensor de movimiento

“ Los sensores infrarrojos pasivos (PIR) son dispositivos para la detección de movimiento. Son baratos, pequeños, de baja potencia, y fáciles de usar. Por esta



razón son frecuentemente usados en juguetes, aplicaciones domóticas o sistemas de seguridad.

Los sensores PIR **se basan en la medición de la radiación infrarroja**. Todos los cuerpos (vivos o no) emiten una cierta cantidad de energía infrarroja, mayor cuanto mayor es su temperatura. Los dispositivos PIR disponen de un sensor piro eléctrico capaz de captar esta radiación y convertirla en una señal eléctrica. En realidad **cada sensor está dividido en dos campos** y se dispone de un circuito eléctrico que compensa ambas mediciones. Si ambos campos reciben la misma cantidad de infrarrojos la señal eléctrica resultante es nula. Por el contrario, si los dos campos realizan una medición diferente, se genera una señal eléctrica.

De esta forma, si un objeto atraviesa uno de los campos se genera una señal eléctrica diferencial, que es captada por el sensor, y se emite una señal digital.

[2026-02-09 12_25_59-Greenshot.png](#)

El otro elemento restante para que todo funcione es **la óptica del sensor**. Básicamente es una cúpula de plástico formada por lentes de fresnel, que divide el espacio en zonas, y enfoca la radiación infrarroja a cada uno de los campos del PIR.

De esta manera, cada uno de los sensores capta un promedio de la radiación infrarroja del entorno. Cuando un objeto entra en el rango del sensor, alguna de las zonas marcadas por la óptica recibirá una cantidad distinta de radiación, que será captado por uno de los campos del sensor PIR, disparando la alarma.

[2026-02-09 12_27_59-Inbox - jjquintana@educa.aragon.es - Mozilla Thunderbird.png](#)

Luis Llamas CC-BY-NC-SA <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>

Puedes ver ejemplos de uso en robótica en :

- [Smart Home para Microbit](#)
- [Smart Agriculture Kit para Microbit](#)

[2022-04-16 10_24_22-sensor pir arduino at DuckDuckGo.png](#)[2025-11-08 18_11_48-.png](#)

Más sensibles son los **sensores de microondas**. Son un radar que por efecto Doppler pueden captar cualquier objeto en movimiento dentro de un alcance de 5-7 metros en cualquier dirección e independiente de su temperatura. Es un buen sensor para alarmas, activación de luz por presencia..... Para saber más [ver la página de Luis Llamas](#)

[image-1650097316026.png](#)

Su conexión es muy sencilla, es un detector digital que hay que alimentarlo como el resto de sensores.

[image-1650097889205.png](#)

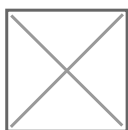
Curiosamente, la gran ventaja técnica de los de microondas **es un gran inconveniente para usarlo en el aula**, con cualquier movimiento se dispara, luego **para clase es mejor el sensor PIR**

Esta página esta adaptada de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA.

2 ARDUINOBLOCKS

Sensores del rover Arduino

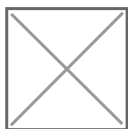
La ventaja de ARDUINOBLOCKS es que tiene ya incorporado una cantidad de bloques con sus librerías y funciones integradas por lo que el alumno no tiene que pelear con códigos y situaciones especiales, simplemente arrastrar el bloque y funcionar:



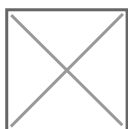
Probando el sensor LDR.

Vamos a probar este sensor que está conectado en una entrada analógica, para ello simplemente que escriba su valor en el puerto, y veremos cómo va cambiando.

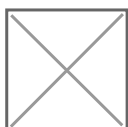
Es recomendable para la lectura de los sensores, crear variables que almacenen los valores leídos, para luego utilizar estos valores:



El siguiente código lee el sensor cada 2 seg. y lo vemos por el puerto serie. Podemos comprobar que **cuanto más luz, la lectura es menor.**



Guardamos el proyecto, lo subimos al Arduino y pulsamos en consola para ver cómo cambia los valores.



Probando el sensor DHT22.

Vamos a probar este sensor, que mide a la vez temperatura y humedad. Con el bloque correspondiente, [nos ahorramos bastante código](#).



Probando el sensor Hall.

Este sensor tiene también su bloque :



Podemos construir un programa análogo al anterior y veremos que su salida es simplemente 0 o 1 depende si acercamos o no un imán.

2 ARDUINOBLOCKS

Actuadores y otras salidas

2 ARDUINOBLOCKS

Motores en el rover Arduino

Los motores **derechos** se gobiernan con la siguiente instrucción :



- Activando el pin D1 encendemos los motores
- El pin D3 en ON van hacia delante, si queremos que vayan hacia atrás, ponemos D3 en OFF
- El valor PWM del pin D1 es la potencia que transferimos al motor, puede ser desde 0 hasta 1023

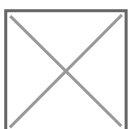
Recomendamos leer [esta página](#) sobre el significado de las salidas PWM.

Para los motores **izquierdos** se gobiernan con la siguiente instrucción :



- Activando el pin D2 encendemos los motores
- El pin D4 en ON van hacia delante, si queremos que vayan hacia atrás, ponemos D4 en OFF
- El valor PWM del pin D2 es la potencia que transferimos al motor, puede ser desde 0 hasta 1023.

Recomendamos ejecutar estas instrucciones y jugar con estos valores para ver si están bien conectados los motores :





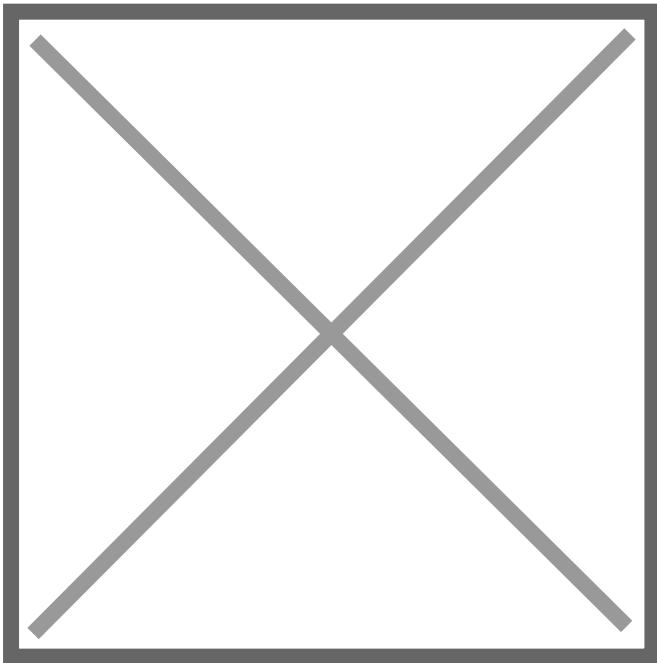
3 BLYNK

3 BLYNK

Internet de las cosas IoT

El **Internet de las cosas** (Internet of Thing IoT) describe objetos físicos —o grupos de estos— con sensores, capacidad de procesamiento, software y otras tecnologías que se conectan e intercambian datos con otros dispositivos y sistemas a través de internet u otras redes de comunicación. El Internet de las cosas se ha considerado un término erróneo porque los dispositivos no necesitan estar conectados a la Internet pública. Sólo necesitan estar conectadas a una red y ser direccionables individualmente

[Fuente Wikipedia IoT Internet de las cosas CC-BY-SA](#)



[De Drawed by Wilgenbroed on FlickrTranslated by Prades97 CC BY-SA 3.0](#)

Estamos hablando de dispositivos que se conectan a internet de forma desatendida, por vía hardware (o mejor dicho firmware) a diferencia de un ordenador, tablet o móvil, donde tienes que configurar por software el dispositivo y hay un diálogo entre usuario y dispositivo sobre el uso de Internet (el software solicita tal página web, tales datos etc por voluntad del usuario o por diálogo con el usuario) Aquí los dispositivos están ya configurados de los datos que se comunican. Es decir "conectar y olvidar".

Piensa en la diferencia entre un enchufe inteligente y un ordenador, el primero es lo que se considera dentro de IoT

Desventajas: El acceso a Internet de dispositivos caseros puede generar problemas a nivel mundial:

- [el caso Mirai](#)
- [aspiradores que nos espían](#)

IoT en los cursos de Aularagón

- **Blynk:** lo que nos gusta de esta herramienta es que es casi "instantánea" o "síncrona". Esto es imprescindible con ciertos robots como el **Rover Marciano con Arduino**. Necesitamos que "gire" para evitar un obstáculo, no podemos esperar !!! Veremos con **BLYNK** un protocolo que entre el dispositivo electrónico (nuestro robot) y nosotros (en ordenador, en una APP en el móvil) la comunicación es instantánea, gracias a un servidor que hará de intermedio, que puede ser local (BLYNK LEGACY) o en Internet (BLYNK IoT).
 - **Blynk legacy** es la que se va a trabajar en
 - [Rover Marciano con Arduino](#)
 - [Arduinoblocks en el aula](#)
 - [ESP32 en el aula](#)
 - **Blynk IoT** es la que se va a trabajar con
 - [En ESP32 en el aula](#)
 - [En Smart Home ESP32](#)
- **ThinkSpeak y SmartioSpace**
 - [Smart Agriculture Kit para Micro:bit](#)
- **MQTT** El emisor envía datos, se almacenan en un servidor, y cuando puede, lo vuelca al cliente. Cliente y emisor pueden ser el dispositivo electrónico y nosotros o viceversa. Veremos que esto es lo que hace el protocolo **MQTT** y está tremendamente extendido por lo barato y fácil que es. Hace que los servidores no estén tan ocupados, por lo tanto hay varios proveedores que ofrecen este servicio gratuitamente. Hay robots como los que tienen la placa **TDR STEAM IMAGINA** que envía datos de temperatura, humedad, .. y pueden recibir datos pero no precisan de esta exigencia instantánea como un rover.
 - [ESP32 EN EL AULA](#)
 - [En Smart Home ESP32](#)
- **TELEGRAM**
 - [ESP32 EN EL AULA](#)
 - [En Smart Home ESP32](#)



- **Arduino cloud IoT**
 - [Arduino Alvik](#)
- **Cyberpi y mBot2**
 - [IoT con Cyberpi](#)

3 BLYNK

Qué es Blynk

Blynk pertenece al mundo del Internet de las cosas IoT que has visto anteriormente. Hay dos opciones en IoT.

- MQTT ocupa pocos datos pero es lento
- Instantaneo: Blynk Aquí utilizaremos Blynk para comunicarnos casi instantáneamente. Con el **Rover con Arduino** esto es imprescindible ¡¡no queremos que se estrelle!! con **TdR STEAM** no es tan imprescindible, pero si queremos interactuar con él (encender sus leds, hacer un pitido...) con MQTT puede ser desesperante.

Blynk nos posibilita la conexión de un hardware conectado a Internet: Arduino, Raspberry, NodeMCU, ESP32 ... con la APP Blynk (hay dos APPs que veremos IoT y Legacy) que puedes poner diferentes elementos de control.



O sea, se necesitan tres cosas:

1. Un dispositivo para poner la **APP** por ejemplo un móvil Smartphone o una tableta, la APP hay dos tipos:
 1. **Blynk IoT**
 2. **Blynk Legacy**
2. El dispositivo o **robot** que esté preparado para conectarse por Internet IoT Hardware
 1. En el caso del Rover con Arduino es el NodeMCU
 2. En el caso de Arduinoblocks en el aula es un Arduino con el ESP01 o ESP8266
 3. En el caso de ESP32 en el aula es un ESP32 que tiene la wifi integrada.
3. Una conexión entre los dos
 1. En el caso de **Blynk IoT** lo proporciona la empresa, con un servidor remoto en Internet
 2. En el caso de **Blynk Legacy** lo proporcionamos nosotros con una Raspberry que hará de servidor local en nuestra red local. (en los cursos de Catedu para que no te entretengas con esto, te proporcionamos un servidor nuestro).



3 BLYNK

Blynk legacy y Blynk IoT


Antes de junio 2022 existía **Blynk legacy** en los servidores de [Blynk Inc](#) pero a partir de esa fecha, ha dejado de funcionar pasando a **Blynk IoT** ([ver noticia](#)) esto crea varios inconvenientes:

VENTAJAS E INCONVENIENTES DE USAR BLYNK LEGACY

- Compatible con todas las placas en Arduinoblocks
- Muchos centros y proyectos utilizan en sus servidores locales **BLYNK LEGACY** por lo que ARDUINOBLOCKS mantendrá en el futuro su compatibilidad con Blynk legacy.
- La APP se instala en en Google Play (pero con riesgo que ya no dejen de ofertarlo) o fuera de Google Play
- El servidor hay dos opciones
 - En tu centro hacer un **SERVIDOR LOCAL** lo más fácil es con una Raspberry, en el curso se enseña cómo hacerlo.
 - En un servidor de CATEDU SÓLO PARA LOS CURSOS DE ROBOTICA
- Al utilizar **BLYNK LEGACY** con tu propio servidor tiene la ventaja que no tiene ninguna limitación de widgets y tienes el control de los alumnos, pero si el servidor es local, no se puede utilizar por ejemplo desde casa.

VENTAJAS E INCONVENIENTES DE USAR BLYNK IoT

- **BLYNK IoT** tiene limitaciones, la mayoría de widgets son de pago pero los esenciales: switch, slider, led, label, gauge son gratuitos.
- Es una buena opción para que los alumnos puedan trabajar en casa con su robot, estamos a la espera que ARDUINOBLOCKS sea compatible con **BLYNK IoT ACTUALMENTE SÓLO ES POSIBLE CON LA PLACA ESP32 STEAMAKER** (OJO hay que tener [Arduinoconnector actualizado](#))

“ Ya puedes probar los bloques para el nuevo Blynk IoT [@blynk_app](#) con la placa [#esp32steamakers](#) de [@Innovadidactic](#) y [@ArduinoBlocks](#)  [#iot](#) [#blynk](#) [#esp32](#) [#steamakers](#) [#steam](#) [#dashboard](#) pic.twitter.com/kzPNkSpbAV

— ArduinoBlocks (@ArduinoBlocks) [November 19, 2022](#)



4 BLYNK LEGACY

4 BLYNK LEGACY

Conseguir la App BLYNK LEGACY

La APP de **BLYNK LEGACY** no la encontramos en el buscador de Google Play de forma tradicional, pues está sin mantenimiento, tenemos varias opciones para descargar la APP.

OJO no confundir Blynk (legacy) que es el que vamos a trabajar
BLYNKEGACY.PNG
con Bynk IoT que **sí** que lo encuentras en el buscador de Google Play
BlynkIoTApp.png

OPCIÓN A Desde Google Play

En este enlace aún tiene la APP antigua, pero es de propiedad de Blynk Inc luego **no es de extrañar que desaparezca**.

<https://play.google.com/store/apps/details?id=cc.blynk>

appblynklegacy.jpeg

OPCION B Fuera de Google Play

OPCIÓN B: ANTES DE LA INSTALACIÓN

No nos tenemos que fiar de descargar APK de repositorios fuera de Google Play, luego una vez descargado, pasamos el fichero por una página de Antivirus, por ejemplo

<https://www.virustotal.com/gui/home/upload>

VIRUSTOTAL.png

No nos tiene que dar ningún virus

antivirustotal-2.png

Para instalarlo en tu móvil:



Primero, tenemos que llevar el fichero .apk a tu dispositivo, recomendamos usar email, por ejemplo enviándote a ti mismo un correo electrónico y lo consultas desde el móvil.

Segundo, tenemos que instalarlo y para ello hay que decirle que confiamos con apps instaladas fuera del repositorio oficial (Google Play), como depende las pantallas del modelo de tu móvil, puedes encontrar cualquier tutorial en internet cómo se hace pero prácticamente es llegar a la misma pantalla en seguridad/privacidad- permitir apps de orígenes desconocidos.

[origenesdesconocidos.png](#)

Tercero instalarlo

Cuarto, deshacer el segundo paso

OPCIÓN B DESCARGA

Puedes buscar en un navegador esta APP, encontraras varios sitios para descargarla, por ejemplo aquí:

- <https://blynk.uptodown.com/android/descargar>
- <https://blynk.en.uptodown.com/android/download/3921223>
- Github: <https://github.com/BlynkMobile>
- [Desde este repositorio nuestro](#)

[BlynkApp.png](#)

4 BLYNK LEGACY

Crear cuentas

Antes se creaban cuentas en Blynk usando el correo electrónico: Cada usuario en la APP de Blynk podía crearse una cuenta, enviando un email. El servidor local Blynk puede enviar correos electrónicos si tiene configurado un fichero que se llama mail.properties y se envía el correo con las credenciales. No obstante esta opción de crear cuenta **ha desaparecido** de la APP (En CATEDU ya *barruntábamos* esto, que Blynk Inc al quitar crear nuevos usuarios es que pensaba quitar este servicio Blynk legacy).

No obstante, la creación de usuarios mediante emails era un problema pues menores de 16 años no pueden tener emails ni redes sociales, ni whatsapps (¿oigo risas?).

En el curso vas a utilizar los servidores de CATEDU luego tendrás asignado 12 cuentas de alumnos + un profesor

Recibirás un email con los nombres de usuario y las contraseñas

¿Y fuera del curso?

En este caso necesitarás tener un servidor Blynk legacy local que tal y como lo explicamos en los siguiente capítulos, montado en una Raspberry.

Entramos en la **carpeta de Blynk en nuestra Raspberry**, ya sea por VNC, o con una pantalla, teclado y ratón o por SSH y lo que hay que hacer es crear unos ficheros texto que se llamen **emailusuario.Blynk.user** los emails de los usuarios pueden ser inventados, por ejemplo en la figura puedes ver que en la raspberry hay dos usuarios user1@yo.es user2@yo.es luego los ficheros textos son **user1@yo.es.Blynk.user** y el otro es **user2@yo.es.Blynk.user**.

Entrando por VNC o pantalla, teclado y ratón:

[userblynkvnc.jpg](#)

Entrando por SSH

[userblynkssh.jpg](#)

El contenido por ejemplo del fichero **user2@yo.es.Blynk.user** es:

|



```
{ "name": "user2@yo.es", "email": "user2@yo.es",
  "appName": "Blynk", "region": "local", "ip": "ip_local_raspberry", "pass": "uffzlHwvej
  aeZZz2PzqmKuB22OACUDRLWvKEI7RCDQI=", "lastModifiedTs": 1663010486621,
  "lastLoggedIP": "192.168.1.73", "lastLoggedAt": 1635057921978, "profile": {}, "isFacebookUser": false, "isSuperAdmin": false, "energy": 99600, "id": "user2@yo.es-Blynk" }
```

Luego lo único que tienes que hacer para crear nuevos usuarios es **crear ficheros texto que se llamen [userx@yo.es.Blynk.user](#) y con contenido igual que el anterior pero cambiando el número 2 por x** (si quieres respetar [userx@yo.es](#) pero puedes inventarte cualquier email ficticio)

¿Por qué puede ser que el contenido de un fichero pueda tener más información o menos que otros?

Porque en esos ficheros txt tiene la información de los proyectos creados por el usuario

¿Entonces si el fichero [user1@yo.es.Blynk.user](#) tiene toda la información del usuario y todas sus proyectos... si ese usuario quiere llevarse sus proyectos a otro servidor Blynk sólo tiene que copiar y pegar es fichero texto?

Respuesta : SI , así de sencillo, ese fichero texto (por lo tanto libre de virus) tiene toda la información.

¿En la APP no se almacena ningún proyecto?

Ninguno, todos los proyectos están almacenados en esos ficheros txt si no hay conexión con el servidor, en la APP no aparece NADA

Vale, ya sé crear los nombres de usuarios ¿Y las contraseñas?

En teoría están encriptadas en el fichero texto, como puedes ver, sale `uffzlHwvej` `aeZZz2PzqmKuB22OACUDRLWvKEI7RCDQI=` eso significa *alcorisa* Pero tranquilo, luego veremos en el panel de control cómo se pueden poner las contraseñas sin encriptar

+información [aquí](#)

4 BLYNK LEGACY

Entrando en el Blynk Legacy: El panel de control

Como entro en el panel de control

- **En el curso de CATEDU**
 - Habrás recibido un email con la URL para entrar en la administración web del Blynk Legacy, con usuario y contraseña
- **Fuera del curso de CATEDU usando tu servidor local**
 - Desde cualquier ordenador conectado en la misma red local que la raspberry, ejecutas en un navegador la siguiente dirección:
 - Si la IP de la Raspberry es 192.168.1.112 entonces entramos en:
 - <https://192.168.1.112:9443/admin>

Vale, ya estoy ¿y ahora qué?

Seguramente te saldrá la siguiente advertencia por el certificado SSL, dale a **Configuración avanzada** y luego a **Acceder a (la IP del servidor Blynk Legacy) sitio no seguro**



Luego entramos con el usuario y contraseña que nos han proporcionado

Cambiando las contraseñas de los alumnos

Y podemos ver los usuarios creados con los ficheros textos mencionados anteriormente

[userblynk.jpg](#)

Entrando en un usuario PODEMOS GENERARLE UNA CONTRASEÑA, por supuesto la pones normal, sin encriptar

También puedes borrar el usuario si no nos interesa este usuario.

[userblynk2.jpg](#)

4 BLYNK LEGACY

En la APP

Conectar con el nuevo usuario

Entramos en la APP de Blynk que hemos visto anteriormente como descargarla e instalara y entramos en **Log In**

[appblynk1.jpg](#)

Y ponemos el nombre de usuario que hemos creado anteriormente, y la contraseña, la que hayamos definido en el panel de control

IMPORTANTE: Entra en **CUSTOM** y pon **LA DIRECCIÓN IP DEL SERVIDOR BLYNK LEGACY**

- Si es dentro del curso de CATEDU habrás recibido un email de las IPs y de los puertos que hay que poner en esta APP de Blynk Legacy
- Si es fuera del curso, la IP DE LA RASPBERRY, O SEA DEL SERVIDOR LOCAL BLYNK y puerto 9443

[appblynk2.jpg](#)

Crear un nuevo proyecto

Podemos ver los proyectos que hay ya almacenados, vamos a crear un nuevo proyecto:

[pantallaNewProjectblynk.jpg](#)

Nos preguntará por el **DEVICE:**

[deviceBlynk.jpg](#)

- En **Arduinoblocks en el Aula**, la placa es el TDR STEAM colocado en un Arduino con el ESP01 conectado, luego **DEVICE Arduino Uno CONNECTION TYPE Wifi**
- En **Rover con Arduino**, la placa es un NodeMCU v2, luego **DEVICE NodeMCU CONNECTION TYPE Wifi**
- En **ESP32 en el Aula**, la placa es un ESP32 con el Imagina TDR STEAM, luego **DEVICE ESP32 DevBoard CONNECTION TYPE Wifi**

Empieza un nuevo proyecto, tienes un montón de energía y widgets para hacerlo:



Crea un nuevo proyecto, y así se crea en el servidor Blynk local un **TOKEN** que ahora veremos qué es

Aquí tienes la captura de un proyecto o dashboard creado para el kit Imagina TDR Steam

[appblynk3.jpeg](#)

4 BLYNK LEGACY

Poner el TOKEN en ARDUINOBLOCKS

¿Cómo conseguir el TOKEN?

OPCION A: QUE LO HAGA EL ALUMNO En la APP

Es la opción más sencilla, entrar en Devices **pulsar en el Token y se copiará en el portapapeles** del móvil, luego ir a cualquier aplicación para poder enviarla al PC (email, whatsapp...) y pegarla en Arduinoblocks en la instrucción que luego veremos.

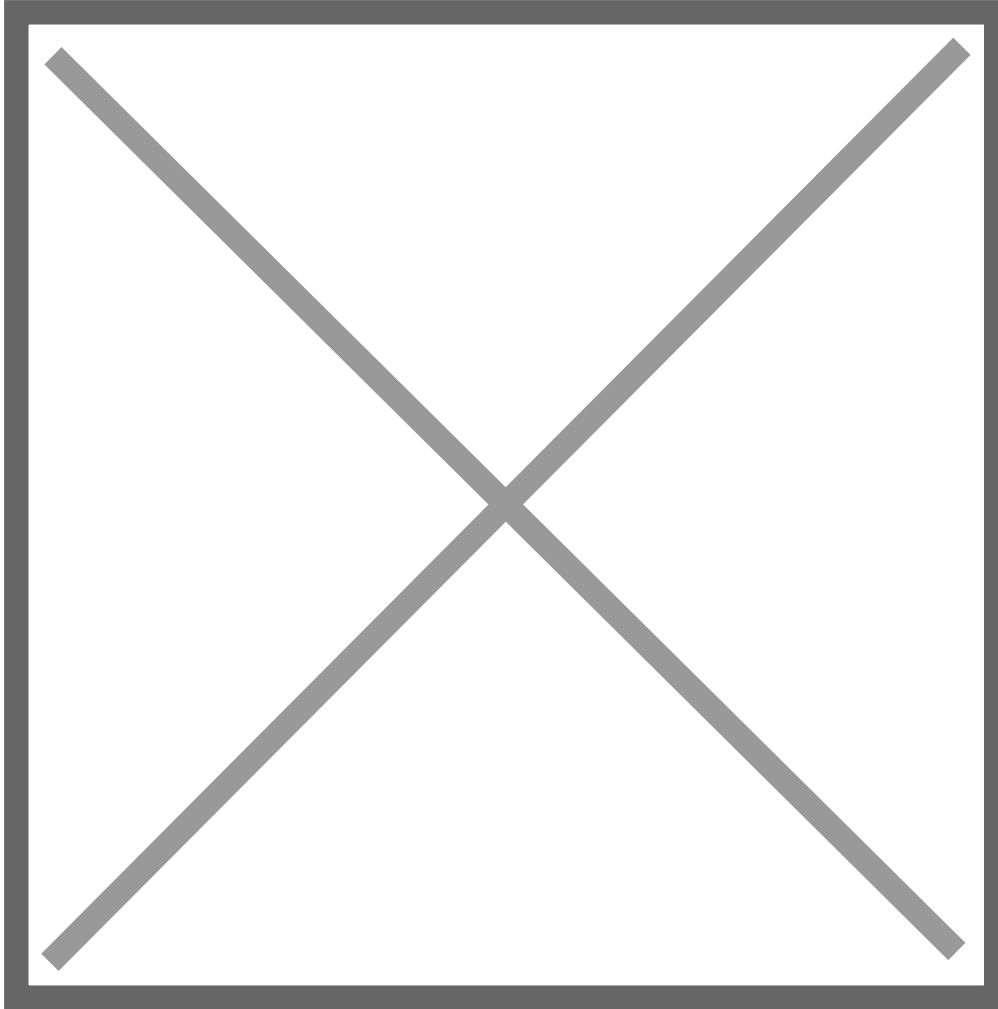
[devices-app-blynk.png](#)

OPCION B: QUE LO HAGA EL PROFESOR En el panel de control

Entramos en el panel de control del servidor Blynk y podemos ver en los usuarios creados los tokens

[userblynk.jpg](#)

Entrando en un usuario PODEMOS VER SUS PROYECTOS Y LOS TOKENS



RECUERDA: UN PROYECTO UN TOKEN

¿Qué se hace con ese TOKEN? Se pone en ARDUINOBLOCKS:

Entramos en ArduinoBlocks y vamos al bloque de comunicaciones IoT - Blynk legacy

ATENCIÓN: ARDUINOBLOCKS CONNECTOR tiene que estar actualizado, al menos v5 (octubre 2022)

Tenemos que poner los datos de la **IP servidor BLYNK LOCAL**, y el **TOKEN** del proyecto

- **DENTRO DEL CURSO CATEDU** Habrás recibido un email con las IPs y los puertos a poner en Arduinoblocks
- **SI ES EN TU SERVIDOR BLYNK LOCAL** Hay que poner la IP de la Raspberry o tu servidor local y **EL PUERTO 8080**



En el caso de que es un proyecto con **ARDUINO** o **TDR-STEAM- ARDUINO** con **ESP01** sale :

[arduinoblocksblink2.jpg](#)

En el caso de sea un proyecto con **NodeMCU** o **ESP32** como en el curso del Rover con Arduino o IoT en el aula sale :

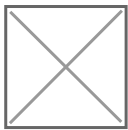
[nuevoblinkconexion.jpg](#)

4 BLYNK LEGACY

Endender LED

Vamos a empezar a hacer algo....

Si antes [hicimos un intermitente por D4](#), ahora vamos a encender el LED en D0:

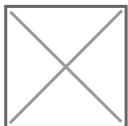


Y el servidor BLYNK hará de puente entre nuestra App y el NodeMCU :

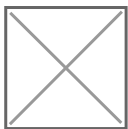


Primer programa con Blynk: Encender D0 : 1 Conseguir el Token

Empezamos en la APP Blynk creamos un nuevo proyecto:



En la siguiente pantalla pregunta por el dispositivo, le decimos ESP8266 y le ponemos un nombre:



Y nos dice que el **Token** nos lo ha enviado por correo electrónico **MENTIRA**, el token lo vemos tal y como hemos visto anteriormente.



Recuerda : **Un proyecto en BLYNK = Un Token**

Primer programa con Blynk: Encender D0 : 2 Poner los controles

Una vez dentro del proyecto de la App aparece todo un panel si nada, añadimos controles apretando al + que hay en la parte superior



Seleccionamos el botón



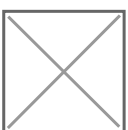
Nos aparece el botón



Pulsamos sobre él para entrar en sus propiedades y cambiamos que vaya de 1 a 0 pues recuerda que este pin va al revés RECUERDA QUE EL NODEMCU TRABAJA CON **LOGICA INVERSA**



Pulsamos en PIN y le decimos que sea el **GP16**





¿Pero no era el D0? ¿Qué es eso del GP16?

La nomenclatura de los pines cambia, el D0 es el GP16, viene de GPIO General Port Input Output. El D4 es el GP2

¿Por qué los fabricantes nos lían? ☐☐



Primer programa con Blynk: Encender D0 : 3 ARDUINOBLOCKS

Ahora le tenemos que decir a nuestro NodeMCU la configuración de la Wifi, y el Token que hemos recibido, ¿cómo? muy fácil, con ARDUINOBLOCKS

Abrimos un proyecto y ponemos esta instrucción :

EL TOKEN ES EL QUE HEMOS CONSEGUIDO

Novedad: las capturas a veces cambian, ahora el nombre de la wifi y la clave está en el mismo bloque de Blynk Iniciar

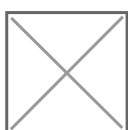
[BLINKCONFIG.png](#)

Donde :

- **WIFI SSID** el nombre de la Wifi donde queremos que se conecte nuestro rover
- **WIFI CLAVE** la contraseña de la WIFI
- **IP Servidor** PONEMOS LA IP del servidor Blynk Legacy.
 - En el curso CATEDU habrás recibido un email con las IPs y el puerto
 - Fuera del curso, tendrás que poner la IP local de tu servidor y el puerto 8080
- **Codigo Auth** es el TOKEN que has conseguido

¿Nada más?

Nada más !! así de sencillo. Le damos a **subir** (teniendo el programa Arduinoblocks conector minimizado, eso lo podemos ver enseguida pues detecta en que COM está conectado, en la figura sale COM5)





Vamos a la APP de BLYNK, Pulsamos al botón de **play ▶** de arriba arriba y vemos el estado NO TIENE QUE APARECER UN PUNTO ROJO si aparece es que nuestra placa no está conectada

[Blink-chivato.png](#)

Luego al botón:



Y podemos ver cómo se enciende y apaga el led D0



Recuerda : Un proyecto en BLYNK = Un Token

RETO: AHORA CON D4

4 BLYNK LEGACY

Encender gradualmente LED

Para ello vamos a incorporar un Slider



Y de propiedades que vaya al GP2 y que vaya al revés de 1023 a 0



El programa en ARDUINOBLOCKS es el mismo !!! no hace falta tocar nada de lo visto anteriormente !!! O sea el nombre de la wifi, su contraseña, la IP del servidor Blynk Legaci, y el token, que es el mismo pues estamos en el mismo proyecto o dashboard.



4 BLYNK LEGACY

Medir la luz del LDR

Ahora añadimos otro control **Gauge**



Elegimos de PIN el analógico (sólo hay uno). [Recuerda](#) que el LDR está conectado al pin analógico.



Y como va al revés, vamos a poner el medidor al revés: de 1023 a 0



https://www.youtube.com/embed/n6m_8mGBwXY

¿Y en Arduinoblocks?

Es el mismo !!!! no hay que tocar nada !!!

4 BLYNK LEGACY

Medir Temperatura y Humedad con el DHT22

Este caso es distinto, pues

- En el **Rover marciano con Arduino** el DHT22 está conectado al pin D5 digital del NodeMCU
- En el **Arduinobloks en el aula**, el DHT11 de la placa Imagina TDR Steam está conectado al pin D4 digital del Arduino
- En el **IoT en el Aula** el DHT11 de la placa Imagina TDR Steam está conectado al pin D4 digital del ESP32

Pero los dos miden dos variables **Temperatura Y Humedad**

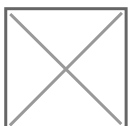
□□□ □¿Cómo hacemos para medir dos variables en un sensor que está conectado a un sólo PIN?□□□

Solución PINES VIRTUALES: si, has leído bien, vamos a utilizar **pinos virtuales**.

En ARDUINOBLOCKS

Vamos a ARDUINOBLOCKS y establecemos dos pines virtuales, uno para la temperatura y otro para la humedad, arbitrariamente he puesto V2 y V3 pero puede ser cualquiera.

- Para el kit de **ROVER MARCIANO CON ARDUINO** poner pin **D5**
 - Si el sensor es AZUL entonces DHT11 si el sensor es BLANCO entonces DHT22



- Para el kit que tenga la placa **TDR STEAM IMAGINA** es decir los cursos **ARDUINO BLOCKS EN EL AULA y ESP32 EN EL AULA** poner pin **D4**
 - Si el sensor es AZUL entonces DHT11 si el sensor es BLANCO entonces DHT22



dht11arduinoblockstdrsteam2.jpg

Fíjate que hay dos maneras de programar estos eventos:

- en el primero, utilizamos un bucle propio de Blynk que simplemente cada segundo lee el sensor y los almacenan los pines virtuales V2 y V3
- en el segundo no se ha utilizado ese bucle propio de Blynk sino simplemente dentro de Bucle hemos puesto otro bucle que se ejecuta cada 2 segundos, y dentro una función que se llama LECTURAS. dentro de lecturas esta la lectura de los pines virtuales V2 y V3

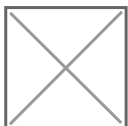
Da igual. Nosotros tenemos preferencia por el segundo método, nos parece más elegante y más controlable.

En la APP de Blynk

Y en Blynk incorporamos un Gauge que sea al PIN VIRTUAL V2.



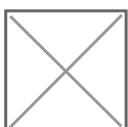
Modificamos también los límites, pues por defecto sale 0 a 1023 y se vería muy bajo la temperatura, ponemos -10 a 30. Si es para el Rover marciano de Arduino, allí se puede llegar a -50°C y si es para el TDR Steam será en Aragón, el récord lo tiene [Fuentes Claras con -30°C](#).



Para la humedad hacemos lo mismo:

- Pin virtual V3
- Límites 0% a 100%

Resultado :



4 BLYNK LEGACY

Medir el campo magnético

El programa en ARDUINOBLOCKS es: (no cambiamos el bloque iniciar con el nombre de la wifi, su contraseña, la IP del servidor Blynk Legacy y el Token pues estamos añadiendo Gadgets al mismo proyecto o dashboard :



Y el programa en Blynk



Ten en cuenta que va al revés, cuando hay un imán, se pone a 0 **NODE MCU TRABAJA CON LÓGICA INVERSA**

El resultado es (acercando un imán):



Reto:

- Escribir un programa que encienda el pin D4 cuando se acerque un imán

4 BLYNK LEGACY

A por los motores

Cambiamos nuestro programa de ARDUINOBLOCKS utilizando unos pines virtuales para controlar los motores, arbitrariamente le hemos puesto V1 y V2 de tal manera que cuando se reciba un valor de estos pines virtuales, fijan la potencia de los motores:



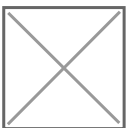
Ponemos dos sliders verticales



Y configuramos uno para V5 y otro para V6:



No te olvides de pulsar el botón de la Shield para que le de energía a los Motores



<https://www.youtube.com/embed/fhz5tYRrmAk>

4 BLYNK LEGACY

A divertirnos con un Joystick

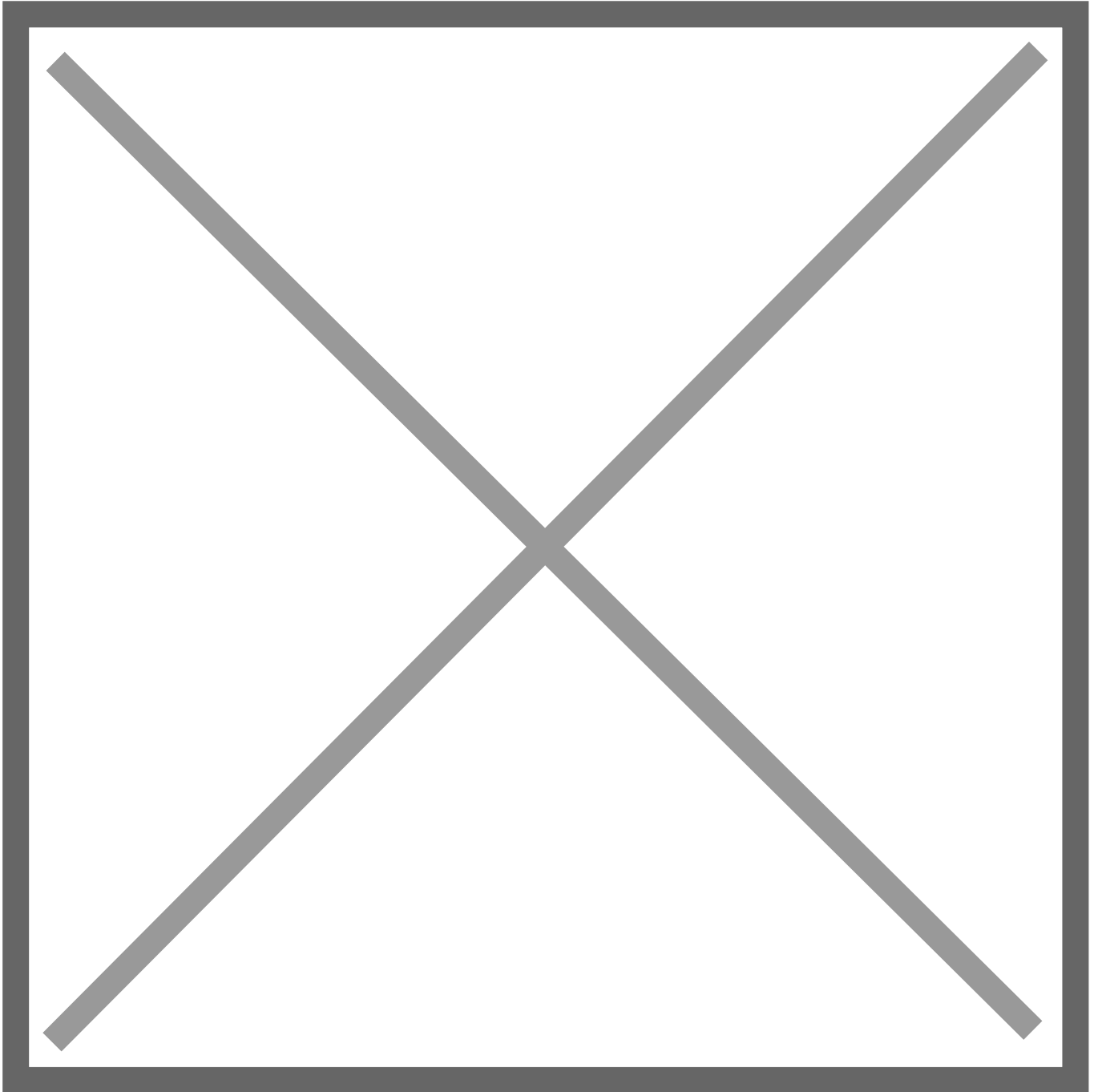
Este elemento es muy gamificador y nos posibilitará ir hacia atrás



Hay que tener en cuenta que los valores del Joystick son los siguientes :



Luego la programación en Arduinoblocks es la siguiente :



4 BLYNK LEGACY

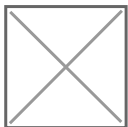
Proyecto final

Vamos a juntarlo todo :

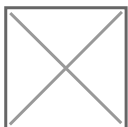
<https://www.youtube.com/embed/g1jwdec86aA>

En Arduinoblocks

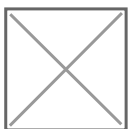
Creamos una función **SENSORES** que lee los tres sensores digitales en tres pines virtuales



Unas funciones **adelante**, **atrás derecha** e **izquierda** (he multiplicado el valor de la potencia por dos para que sea más sensible el Joystick)



Configuramos Blynk Legacy para que nos lea nuestra Wifi, nuestro Token la IP de nuestro servidor Blynk Legacy y si se recibe los valores V0 y V1 del Joystick que vayan a las variables X e Y



Y finalmente el programa principal





El proyecto es público y lo tienes en este enlace (hay que estar logueado en Arduinoblocks)

<http://www.arduinoblocks.com/web/project/486534>

La APP Blynk

Ponemos los siguiente controles

1. El Joystick con los pines virtuales V0 y V1
2. El Gauge de Temperatura a V2
3. El Gauge de Humedad a V3
4. El Gauge de efecto Hall a V4
5. El Gauge de Luz al analógico
6. Un botón al D4

El panel está completo :



5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

Servidor local Blynk

Ya que Blynk Inc ha dejado sin soporte a **Blynk legacy**, tenemos que conseguir un **SERVIDOR BLYNK LOCAL** que haga de puente entre nuestro robot y nuestra App.

Se puede hacer en:

- Una **Raspberry** fácil, barato y poco consumo.
- En un ordenador con Windows
- En un ordenador con Linux

EL CURSO DE CATEDU FACILITA UN SERVIDOR BLYNK en Internet. ES DECIR NO ES LOCAL

SI TU CENTRO ES UN **CENTRO VITALINUX** seguramente tendrá un servidor local. Ponte en contacto con nosotros soportecatedu@educa.aragon.es y te lo instalamos remotamente.

Desventajas de usar un servidor local

Nuestro equipo robot y nuestro móvil **tiene que estar en la misma red local** esto imposibilita que el alumno pueda trabajar en casa, necesita el servidor local.



Energía y control

Los alumnos tienen TODA la energía para hacer sus proyectos por lo tanto pueden usar tantos gadgets como quieran



Además de tener control vía web de todos usuarios, pues tú eres el administrador.

5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

OJO ESTO NO TIENES QUE HACERLO

Este apartado es para tu conocimiento NO LO TIENES QUE HACER, sólo para que sepas como CATEDU ha montado el servidor en la RASPBERRY por si te interesa seguir los mismos pasos

Como CATEDU ya te proporciona la Raspberry con el Blynk local, sólo tienes que conectarlo a tu red y ponerle una IP fija

Opción A La Raspberry con pantalla, teclado y ratón

Esta es más fácil, pero tienes que encender la Raspberry con teclado, pantalla y ratón:

https://docs.google.com/presentation/d/e/2PACX-1vQShQenHWgn0_axlCmlKpIX6kqQOHHzalHhy7SxTtXjLOXZip40J05B-UB7KRx6g1Z8M1LsUg2N8fi8/embed?start=false&loop=false&delayms=3000

Opción B La Raspberry sin pantalla, teclado y ratón

Con esta opción es **más difícil**, pero no tienes que encender la Raspberry para la configuración, sólo manipular los ficheros de la tarjeta micro SD.

<https://docs.google.com/presentation/d/e/2PACX-1vSuE389RRI67Vi1LGxySj5rWjhfnMjV9t27FbO80BOqMqDjyKcH7hFZWdOXubwsuO8NtnwXtaTQfg4p/embed?start=false&loop=false&delayms=3000>

Texto que hay que poner en interfaces

```
auto wlan0
iface wlan0 inet static
address 192.168.1.xxx
gateway 192.168.1.1
netmask 255.255.255.0
wpa-ssid el_nombre_tu_red_wifi
wpa-psk la_contraseña_de_tu_wifi
dns-nameservers 8.8.8.8 188.244.82.1
```

Texto que hay que poner en wpa_supplicant.conf

```
# /etc/wpa_supplicant/wpa_supplicant.conf
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="nombre de tu router o SSID"
    psk="tu contraseña del wi-fi"
    key_mgmt=WPA-PSK
}
```

5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

Raspberry: Qué hay que hacer para empezar

Lo primero que tenemos que conseguir es:

- Tener una Raspberry con el sistema operativo propio de la raspberry instalado, antes se llamaba Raspbian, ahora [Raspberry Pi OS](#)
- Comunicarnos con la Raspberry por comandos SSH
- Una vez hecho esto que es lo básico instalaremos el servidor Blynk



Ahora vamos a ver unas páginas para

- Instalar un **sistema operativo** en una Raspberry
- Una vez instalado hay que **conectarlo a la red wifi** y **ponerle una IP FIJA**
- Para **comunicarse vía comandos SSH**, aconsejamos hacerlo de forma remota, pues lo contrario implica tener siempre una pantalla, teclado y ratón conectado en la Raspberry

¿Qué Raspberry habría que comprar?

Aconsejamos una que tenga la Wi-fi incorporada, nosotros hemos hecho pruebas con la Raspberry Pi 3B+ y con la Raspberry Pi4 y no encontramos diferencias de velocidad para estos propósitos de robótica, luego por la mitad de precio aconsejamos la [Raspberry Pi 3B](#)

5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

Raspberry: Poner un Sistema Operativo

¿Es el único sistema operativo que se puede instalar?

No, podemos instalar otros sistemas operativos en nuestra Raspberry (evidentemente uno sólo cada vez, poner dos a la vez en la *rasp* sería forzarlo demasiado). Por ejemplo:

- Podemos instalar un **Linux-Ubuntu** ligero pero que se puede usar en una aula de informática ([ver libro del curso de Aularagon Monta tu aula de informática con Raspberry](#)).
- Si no podemos vivir sin Windows, hay un **Windows 10 especial** gratis (*sí has leído bien, los de Microsoft van entendiendo eso de si no puedes con ellos únete a ellos, pero no te hagas ilusiones, sólo vale para la Raspberry y no esperes una máquina que sustituya a tu PC va lentísimo*).
- Sistemas operativos para aplicaciones concretas:
 - Video juegos: [Recalbox](#).
 - Dómótica y control [Home Assitant](#)
- Nuestro sistema operativo de este curso (y el mejor que "rula en la Rasp") pues está adaptado a la arquitectura ARM de la Raspberry es **RASPBIAN** o actualmente lo llaman **RASPBERRY PI OS**

Hay dos opciones, elige la que quieras (nuestra preferida es la A):

Opción A Instalar el sistema operativo en la tarjeta sin NOOBS

Con esta opción no tienes que encender la Raspberry.

https://docs.google.com/presentation/d/e/2PACX-1vQoxDLLHMvB-mCQwm2en9cBgb1faamFG0YJliFDFuNrGH8TuH8U-4zCDg_K9CkM4gFI-Wy6TxBNLg9j/embed?start=false&loop=false&delayms=3000

Opción B Instalar la Raspbian en la tarjeta con NOOBS

Con esta opción SI que tienes que encender la Raspberry con pantalla, teclado y ratón

https://docs.google.com/presentation/d/e/2PACX-1vR4wwk8BwdSEgk-aTQ_xgvrQYJ-cr2QhN_35Q-mYJxYedhT5P-vF6UkDbSsHJ5I_zLS7IqBirdNfl4t/embed?start=false&loop=false&delayms=3000

Esto es válido para cualquier ordenador: Descargar la ISO de un Sistema Operativo, grabarlo en un pincho de forma autoarrancable, pero por curiosidad, no es tan fácil como en la RASPBERRY
Se añade esta presentación como curiosidad:

https://docs.google.com/presentation/d/e/2PACX-1vSvr4JxqRNkTwl0mcFJdfble5BtDpSGyLO4ucyAyk65f3zXsFa3zxlyFZiPcqRuv_2YEfrGY39SFVi0/embed?start=false&loop=false&delayms=3000

5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

Raspberry: Conectarlo a la red

Opción A La Raspberry con pantalla, teclado y ratón

Esta es más fácil, pero tienes que encender la Raspberry con teclado, pantalla y ratón:

https://docs.google.com/presentation/d/e/2PACX-1vQShQenHWgn0_axlCmlKplX6kqQOHHzalHhy7SxTtXjLOXZip40JO5B-UB7KRx6g1Z8M1LsUg2N8fi8/embed?start=false&loop=false&delayms=3000

Opción B La Raspberry sin pantalla, teclado y ratón

Con esta opción es **más difícil**, pero no tienes que encender la Raspberry para la configuración, sólo manipular los ficheros de la tarjeta micro SD.

<https://docs.google.com/presentation/d/e/2PACX-1vSuE389RRI67Vi1LGxySj5rWjhfnMjV9t27FbO80BOqMqDjyKcH7hFZWdOXubwsuO8NtnwXtaTQfg4p/embed?start=false&loop=false&delayms=3000>

Texto que hay que poner en interfaces

```
auto wlan0
iface wlan0 inet static
```

```
address 192.168.1.xxx
gateway 192.168.1.1
netmask 255.255.255.0
wpa-ssid el_nombre_tu_red_wifi
wpa-psk la_contraseña_de_tu_wifi
dns-nameservers 8.8.8.8 188.244.82.1
```

Texto que hay que poner en wpa_supplicant.conf

```
# /etc/wpa_supplicant/wpa_supplicant.conf
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="nombre de tu router o SSID"
    psk="tu contraseña del wi-fi"
    key_mgmt=WPA-PSK
}
```

5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

Raspberry: Conexión SSH

Controlar Raspberry pi de forma remota y de forma textual es muy rápido y eficaz, sobre todo para la robótica con AlphaBot.

SSH (Secure Shell) según [Wikipedia](#) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder servidores privados a través de una puerta trasera. Te recomiendo visitar [esta página](#) de Luis Llamas.

Recuerda que lo que hemos hecho es:

- Hemos conectado la Raspberry a la Wifi
- Hemos activado el protocolo SSH y VNC en la raspberry
- Hemos fijado una IP y la sabemos para poder acceder a él.

Ya podemos trabajar por SSH con la Raspberry, no nos hace falta que conectemos la raspberry a una pantalla, teclado, etc....

YA PODEMOS ENCENDER LA RASPBERRY si has elegido las opciones A de [3.-Raspbian](#) y [4.- Conectando](#) aún no habrás encendido la Raspberry.

Si trabajas con Windows

No lo tiene nativo, tenemos que instalar [Putty](#)

Y rellenamos los campos de la IP (la que hayas determinado, en la imagen es 192.168.1.111), puerto y el protocolo SSH, es recomendable grabar la sesión para tenerlo en futuras entradas.



Se abre una ventana pidiendo el usuario **pi** y contraseña por defecto **raspberry**



Si trabajas con Linux

Linux tiene de forma nativa el protocolo SSH, lo activamos en el icono en negro que es el **Terminal** :



Supongamos que la IP fija que le hemos asignado a la Raspberry la IP 192.168.1.131 Teclamos

```
ssh pi@192.168.1.131
```

Te pide usuario y contraseña, por defecto es usuario pi y la contraseña *raspberry*, la tecleas **no la puedes ver**, te tiene que salir algo así en la pantalla:

Cuando te sale pi@raspberrypi: es que ya está preparado para recibir órdenes

```
cat@cat-Aspire-E5-571 ~ $ ssh pi@192.168.1.131
pi@192.168.1.131's password:
Linux raspberrypi 4.9.80-v7+ #1098 SMP Fri Mar 9 19:11:42 GMT 2018 armv7l

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 Mar 23 18:03:09 2018 from 192.168.1.131
pi@raspberrypi:~ $
```

Nota por si te pasa

Si después de un intento fallido, o realizar la conexión con otra computadora.... te sale este error al intentar conectarte por ssh:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @ WARNING: REMOTE
HOST IDENTIFICATION HAS CHANGED! @
```



```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ IT IS POSSIBLE THAT
SOMEONE IS DOING SOMETHING NASTY! Someone could be eavesdropping on you right now
(man-in-the-middle attack)! It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
Please contact your system administrator. Add correct host key in
/home/user/.ssh/known_hosts to get rid of this message. Offending key in
/home/user/.ssh/known_hosts:1 RSA host key for ras.mydomain.com has changed and you
have >requested strict checking. Host key verification failed.
```

Entonces antes de intentar conectarte por ssh, hay que obligar a la Raspberry que reinicie las claves de conexión de ssh, ejecuta esta orden :

Supongamos que la IP fija que le hemos asignado a la Raspberry en el [punto 4](#) es 192.168.1.131
Tecleamos

```
ssh-keygen -R 192.168.1.131
```

5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

Raspberry: Cambiar usuario y contraseña

El usuario por defecto es **pi** y su contraseña es **raspberry** pero lo podemos cambiar perfectamente

Opción A remota con SSH

Entramos por SSH [como hemos aprendido](#), y tecleamos

```
sudo passwd pi
```

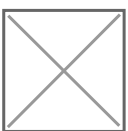
y nos pide enseguida la contraseña, la tenemos que insertar dos veces *no se puede ver*.

Opción B local por pantalla

Entramos por VNC o conectamos una pantalla, teclado y ratón a la raspberry y en **Preferencias-RaspberryPi configuration**



Y entramos en **System**



5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

Raspberry: Apagar

Apagar de forma caliente (es decir simplemente quitando de la fuente de alimentación la corriente) daña al sistema operativo Raspbian dejando ficheros corruptos.

7.1 Opción de forma remota con SSH

Muy fácil, teclea

```
sudo poweroff
```

7.2 Opción local de forma gráfica

Igual que en otros sistemas operativos Windows o Linux tenemos que apagar desde el menú **Shutdown**



5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

Raspberry VNC

VNC es un programa que nos permite la conexión remota con los servidores pero de forma gráfica. Es el sistema elegido por RASPBIAN pues ya lo tiene incorporado, pero no activado.

Hay que tener claro que son **dos programas**: * **VNC SERVER** que es el programa que se ejecuta en la RASPBERRY y que aunque viene instalado, hay que activarlo. Es el programa que lanza el escritorio para que lo lea: * **VNC VIEWER** es el programa que tenemos que tener en nuestro ordenador para ver lo que lanza VNC SERVER.

En la imagen podemos ver un pantallazo de mi ordenador, el escritorio negro es mi escritorio y el de la careterra es el de la Raspberry:



5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

Raspberry: Instalar Java 8

Una vez que tenemos la Raspberry funcionando y accesible por SSH o por VNC es el momento de instalar el servidor Blynk, primero hay que, para instalar Blynk, se necesita la **versión 8 de Java**

Seguimos las instrucciones de <https://github.com/Peterkn2001/blynk-server#blynk-server>

Entramos en la ventana de comandos SSH y ejecutamos las siguientes órdenes, si en algún momento nos pide confirmación [Y/n] es porque faltan descargar paquetes o confirmación de instalación, por lo tanto aceptamos.

Con esta orden busca las actualizaciones

sudo update

y ya ejecutamos que instala la versión 8 de java (curiosamente no funciona con la versión 64 bits de Raspberry OS sino con la de 32 bits)

sudo apt install openjdk-8-jdk

Por último una vez finalizado, comprobamos la versión que coge por defecto

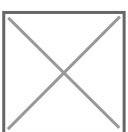
java -version

Tiene que salir Openjdk version 1.8.0 etc...

Si no sale esa versión...

Suele pasar que tenga ya instalado la versión 11, tenemos que obligar a que sea la 8 por defecto, para ello ejecutamos

sudo update-alternatives --config java





Si seleccionamos el 2 nos lo pondrá como por defecto (sale con un (*). Podemos comprobar que ya nos sale la versión con la orden :

java -version

Pero esto no nos sirve, pues al reiniciar la Raspberry volverá a la versión 11.

Para ponerlo por defecto, edita el fichero *environment*

sudo nano /etc/environment

Y si en la orden **sudo update-alternatives --config java** salía que la carpeta donde está la versión 8 es */usr/lib/jvm/java-8-openjdk-armhf/jre/bin/java* luego tenemos que poner en el fichero *environment* :

JAVA_HOME="/usr/lib/jvm/java-8-openjdk-armhf/jre/bin/java"

Guardar, reiniciar

sudo reboot

Y comprueba

java -version

Y tiene que salir :



Para más cosas

Por ejemplo, desinstalarlo, volver a la 11, etc.. visitar esta página :

<https://phoenixnap.com/kb/install-java-raspberry-pi>

5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

Raspberry: Instalar y ejecutar el servidor Blynk local

1 Descarga Blynk

Crea una carpeta en tu directorio home, por ejemplo Blynk

mkdir Blynk

Y descarga Blynk :

```
wget "https://github.com/Peterkn2001/blynk-server/releases/download/v0.41.16/server-0.41.16-java8.jar"
```

El enlace <https://github.com/Peterkn2001/blynk-server/releases/download/v0.41.16/server-0.41.16-java8.jar> conviene actualizarlo, recomendamos visitar la página <https://github.com/Peterkn2001/blynk-server#blynk-server> y coger la última versión de aquí [blynkserver.jpg](#)

2 Configurar server.properties

Necesitamos crear un fichero de configuración para las diferentes opciones que queremos en nuestro servidor Blynk.

Entra en la carpeta creada Blynk y crea el fichero *server.properties*.

cd Blynk

sudo nano server.properties

A continuación se muestra un posible contenido de *server.properties*.

```
initial.energy=1000000
allow.reading.widget.without.active.app=false
user.message.quota.limit=100
```

```
logs.folder=./logs
user.dashboard.max.limit=100
lcd.strings.pool.size=6
server.ssl.key=./server_embedded.key
webhooks.response.size.limit=96
hardware.mqtt.port=8440
table.rows.pool.size=100
terminal.strings.pool.size=25
admin.email=admin@blynk.cc
admin.rootPath=/admin
user.widget.max.size.limit=20
listen.address=
blocking.processor.thread.pool.limit=6
stats.print.worker.period=60000
enable.db=false
force.port.80.for.csv=false
enable.raw.db.data.store=true
restore.host=blynk-cloud.com
csv.export.data.points.max=43200
restore=false
user.profile.max.size=256
allow.store.ip=true
allowed.administrator.ips=0.0.0.0/0,::/0
net.interface=eth
webhooks.frequency.user.quota.limit=1000
http.port=8080
web.request.max.size=524288
user.devices.limit=50
async.logger.ring.buffer.size=2048
user.tags.limit=100
server.ssl.key.pass=
admin.pass=admin
hard.socket.idle.timeout=10
product.name=Blynk
data.folder=/Path
map.strings.pool.size=25
```

```

profile.save.worker.period=60000
https.port=9443
log.level=info
server.ssl.cert=./server_embedded.crt
force.port.80.for.redirect=true
notifications.queue.limit=2000
notifications.frequency.user.quota.limit=5
server.host=192.168.137.1
app.ssl.port=8443
hardware.default.port=8442
hardware.ssl.port=8441
hardware.mqtt.port=8440

```

Para la explicación de cada línea, aconsejo consultar esta [página](#) en el apartado *Configuración avanzada del servidor local*

3 Ejecutar el servidor Blynk local

En la Raspberry por comandos SSH, entramos en la carpeta donde hemos creado el servidor Blynk

cd Blynk

Y ejecutamos el servidor Blynk instalado, pero que cargue la configuración de server.properties que en nuestro caso como el la Raspberry el usuario se llama catedu la carpeta es catedu:

```
java -jar server-0.41.16-java8.jar -dataFolder /home/catedu/Blynk -serverConfig /home/catedu/Blynk/server.properties
```

Curiosamente la primera vez que ejecutas esta instrucción te sale los datos del usuario

admin@blynk.cc **y su contraseña sin encriptar** que por defecto es admin. Si no has tomado nota, ejecutar la instrucción anterior no sirve de nada pues ya ha creado el fichero texto de este usuario.

[usuarioadminblynk.jpg](#)

La única forma de que te vuelva a mostrar la contraseña es borrar el fichero texto y ejecutar otra vez la orden java -jar server....

4 Que la orden de ejecución se haga automáticamente cada vez que se reinicie la Raspberry



Para no repetir estos dos comandos `cd Blynk` y `java -jar server-0.41.16-java8.jar -dataFolder /home/catedu/Blynk -serverConfig /home/catedu/Blynk/server.properties` cada vez que reiniciamos la Raspberry puedes generar un script para que lo ejecute automáticamente, puedes ver buenos tutoriales en Internet.

En internet puedes ver varios métodos:

Método 1 es entrar en `/etc` y editar el fichero `rc.local` y añadir esta línea

```
java -jar server-0.41.16-java8.jar -dataFolder /home/catedu/Blynk -serverConfig /home/catedu/Blynk/server.properties &
```

Método 2 usar el comando `crontab -e` y poner al final la siguiente línea `@reboot java java -jar server-0.41.16-java8.jar -dataFolder /home/catedu/Blynk -serverConfig /home/catedu/Blynk/server.properties`

5 Probarlo

Si la IP de la Raspberry es 192.168.1.112 entonces entramos en:

<https://192.168.1.112:9443/admin>

Vale, ya estoy ¿y ahora qué?

Seguramente te saldrá la siguiente advertencia por el certificado SSL, dale a **Configuración avanzada** y luego a **Acceder a (la IP del servidor Blynk Legacy) sitio no seguro**



Si quieres generar certificados SSL propios para que no salga la anterior pantalla consulta [aquí](#)

Tienes que entrar con el usuario y contraseña fijada en **server.properties** :

```
# Default admin name and password. Will be created on initial server start
admin.email=admin@blynk.cc
admin.pass=admin
```



Si quieres cambiar la contraseña, tienes que hacerlo como un usuario normal en la página de administración tal y como hemos visto en el capítulo [Entrando en el Blynk local: El panel de control](#)

6 Para saber más :

- <https://github.com/Peterkn2001/blynk-server#blynk-server>
- [Intalación de Blynk : How to Install a Blynk Local Server on Raspberry Pi](#)
- [Configuración de server.properties.](#)

7 Para saber más : Configurar mail.properties

Este aparatado ya comentamos que no lo aconsejamos, pues los alumnos no suelen tener email y la APP ya no permite crear usuarios con email, pero si queremos que envíe los tokens por email, hay que crear este fichero para que el servidor envíe por email los tokens de los proyectos

Entra en la carpeta creada Blynk y crea el fichero *mail.properties*.

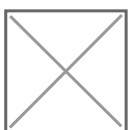
cd Blynk

sudo nano mail.properties

A continuación se muestra una muestra del posible contenido de mail.properties :

```
mail.smtp.auth=true
mail.smtp.starttls.enable=true
mail.smtp.host=smtp.gmail.com
mail.smtp.port=587
mail.smtp.username=Your EMAIL ID
mail.smtp.password=Password
```

Utilizando **Your EMAIL ID** y **Password** los datos de una cuenta de gmail tuya. En esa cuenta tienes que permitir accesos no seguros. Aquí se muestra dónde está en la pantalla de configuración de Gmail :





Para saber más

- [Configuración mail](#)

5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

En Windows Instalar Java 11

Entramos en la web de Oracle - Java - Technologies - JavaSE y descargamos e instalamos el servidor Java 11, si nuestro ordenador no es viejo, seguro que soportará la versión 64

<https://www.oracle.com/es/java/technologies/javase/jdk11-archive-downloads.html>

java11-1.png

Seguramente para la descarga Oracle nos pide que nos registremos. No entendemos el por qué pero efectivamente es necesario proporcionar un email y a ese email envía un enlace para el registro, Una vez registrado ya permite la descarga

Instalamos el servidor Java descargado.

Para que Windows encuentre la ruta del Java instalado cuando ejecutemos comandos, le tenemos que decir en qué camino está la carpeta del java instalado está. Para ello vamos a **C -> Archivos de programa -> Java -> Jdk11...-> carpeta bin** y copiamos la ruta :

java11-2.png

Nos vamos al sistema

java11-3.png

Y configuración avanzada del sistema - variables de entorno- path - editar - nueva y pegamos la ruta que hemos copiado anteriormente

java11-4.png

5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

En Windows Instalar Blynk local

Descargamos el servidor Blynk server desde

<https://github.com/blynk/blynk-server/releases>

o desde <https://drive.google.com/file/d/1nPZxn-9Q9q0ceUhCiZGd4WRC1c8pufEQ/view?usp=sharing>

y lo pegamos en una carpeta, por ejemplo en c:\Blynk_server

Creamos **mail.properties** y **server.properties** como ficheros nuevos, los editamos con un procesador de texto y pegamos el código que hemos explicado [aquí](#)

[blynk-server-windows.png](#)

Ojo: Ten en cuenta que Windows le gusta colocar sus propias extensiones. Si editas un fichero con un procesador de textos, por ejemplo el bloc de notas, le añadirá .txt, si lo editas con Word le añadirá la extensión .docx, tienes que asegurarte que no se añada ninguna extensión, que los nombres de los ficheros sean **mail.properties** y **server.properties** y no otros. Ten activo la visualización de las extensiones para asegurarte que windows no añada las suyas

[blynk-server-windows-2.png](#)

5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

En Windows Ejecutar Blynkserver

Entramos en comandos de windows como administrador

[2022-09-01 20_07_49-.png](#)

Y ejecutamos esta instrucción

```
java -jar server-0.41.17.jar -dataFolder /Path
```

de esta manera se crean las carpetas logs y static que nos dicen los errores que podemos encontrar

La primera pantalla nos dirá el nombre la url, el nombre de usuario y la contraseña para entrar en el servidor blynk

[blynk-user.png](#)

5 CÓMO MONTAR UN SERVIDOR BLYNK LEGACY LOCAL

En Linux

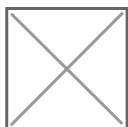
Seguimos las instrucciones de <https://github.com/Peterkn2001/blynk-server#blynk-server>

Pero si tu centro tiene Vitalinux, ponte en contacto con nosotros soportecatedu@educa.aragon.es para instalar en un ordenador Vitalinux el servidor Blynk de forma local

Créditos

Agradecimientos a :

CONSEJERÍA DE EDUCACIÓN Y FORMACIÓN PROFESIONAL DEL GOBIERNO DE CANTABRIA



Autoría

Javier Quintana (2021)

Cualquier observación o detección de error en soporte.catedu.es

Los contenidos se distribuyen bajo licencia **Creative Commons** tipo **BY-NC-SA** excepto en los párrafos que se indique lo contrario.

[image-1648462225402.gif](#)

[image-1648462299882.png](#)

[image-1648462361893.png](#)