



Tutorial Physical Etoys con Arduino

Entorno de programación visual que conecta el mundo virtual de las computadoras con el mundo físico.

COLECCIÓN DE APLICACIONES GRATUITAS PARA CONTEXTOS EDUCATIVOS

Plan Integral de Educación Digital

Gerencia Operativa Incorporación de Tecnologías (InTec)

buenosaires.gob.ar/educacion   educaciónBA



Buenos Aires Ciudad

Ministerio de Educación del Gobierno de la Ciudad de Buenos Aires

01-02-2026



EN TODO ESTÁS VOS

Prólogo

Este tutorial se enmarca dentro de los lineamientos del Plan Integral de Educación Digital (PIED) del Ministerio de Educación del Gobierno de la Ciudad Autónoma de Buenos Aires que busca integrar los procesos de enseñanza y de aprendizaje de las instituciones educativas a la cultura digital.

Uno de los objetivos del PIED es “fomentar el conocimiento y la apropiación crítica de las Tecnologías de la Información y de la Comunicación (TIC) en la comunidad educativa y en la sociedad en general”.

Cada una de las aplicaciones que forman parte de este banco de recursos son herramientas que, utilizándolas de forma creativa, permiten aprender y jugar en entornos digitales. El juego es una poderosa fuente de motivación para los alumnos y favorece la construcción del saber. Todas las aplicaciones son de uso libre y pueden descargarse gratuitamente de Internet e instalarse en cualquier computadora. De esta manera, se promueve la igualdad de oportunidades y posibilidades para que todos puedan acceder a herramientas que desarrolleen la creatividad.

En cada uno de los tutoriales se presentan “consideraciones pedagógicas” que funcionan como disparadores pero que no deben limitar a los usuarios a explorar y desarrollar sus propios usos educativos.

La aplicación de este tutorial no constituye por sí misma una propuesta pedagógica. Su funcionalidad cobra sentido cuando se integra a una actividad. Cada docente o persona que quiera utilizar estos recursos podrá construir su propio recorrido.

Índice

¿Qué es?	4
Requerimientos técnicos.....	4
Consideraciones pedagógicas.....	4
Nociones Básicas	5
▪ Descarga y descompresión del programa.....	5
▪ Pantalla de inicio.....	7
▪ Selección de idioma	7
▪ Entorno de trabajo.....	8
▪ Agregar un objeto específico de robótica.....	9
▪ Halos del objeto Arduino.....	10
▪ Visor del objeto Arduino	12
▪ Categorías del objeto Arduino	13
Paso a paso	13
▪ Conectar la placa Arduino con Physical Etoys	13
▪ Hacer titilar un led.....	16
▪ Modificar la intensidad de un led	21
▪ Simular un semáforo con 3 leds.....	24
▪ Mover un servomotor con joystick.....	26
▪ Ayuda en Physical Etoys.....	29
Enlaces de interés.....	31

¿Qué es?

Physical Etoys es un entorno de programación visual de dispositivos que conecta el mundo virtual de las computadoras con el mundo físico, facilitando la programación de diversos kits de robótica.

Tiene una interfaz sencilla e intuitiva, orientada a la programación de objetos. Es una extensión del programa Etoys, ambos basados en el proyecto Smalltalk.

Es de código abierto y multiplataforma. Se descarga e instala en los equipos para ser utilizado offline.

Physical Etoys fue creado con el fin de ampliar la oferta de programación en robótica educativa específicamente, ante esto puede trabajar con distintos kits destinados a este fin.

Este tutorial se centra en el uso de Physical Etoys con la placa Arduino.

Requerimientos técnicos

- Puede utilizarse bajo los sistemas operativos Windows (32 bits) y GNU/Linux.
- No requiere acceso a internet una vez instalado el programa.
- Para optimizar el uso del programa, es necesaria una placa Arduino donde se puedan apreciar los proyectos programados en Physical Etoys
- URL para descargar el programa: <http://tecnodacta.com.ar/gira/projects/physical-etoys/>

Consideraciones pedagógicas

Nivel: todos los niveles.

Áreas: todas

- **Physical Etoys** se puede utilizar para crear proyectos relacionados con robótica educativa.
- Permite a los niños explorar y experimentar con los conceptos de progra-

mación en una interfaz gráfica sencilla con el fin de generar un movimiento autónomo en una construcción robótica.

- Se recomienda leer en primer lugar el tutorial de Etoys dentro del cual se encuentran especificadas varias herramientas de Physical Etoys.

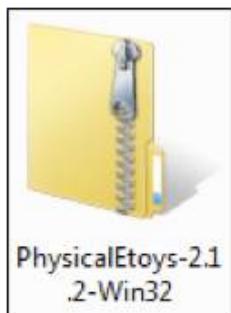
Nociones Básicas

► Descarga y descompresión del programa

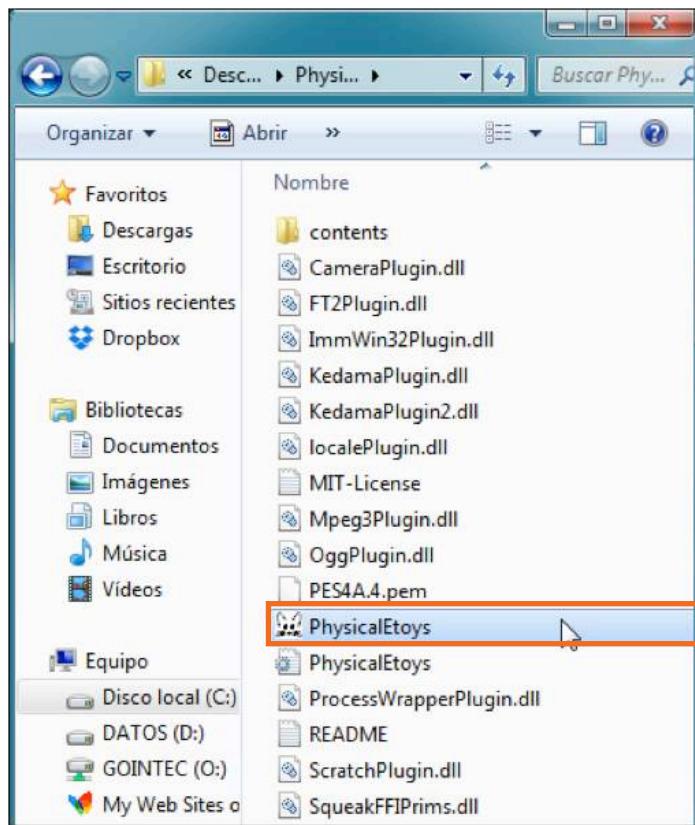
La descarga del programa se realiza utilizando cualquier navegador ingresando a la página <http://tecnodacta.com.ar/gira/projects/physical-etoys/#descarga>



El archivo que se descarga, se encuentra comprimido.

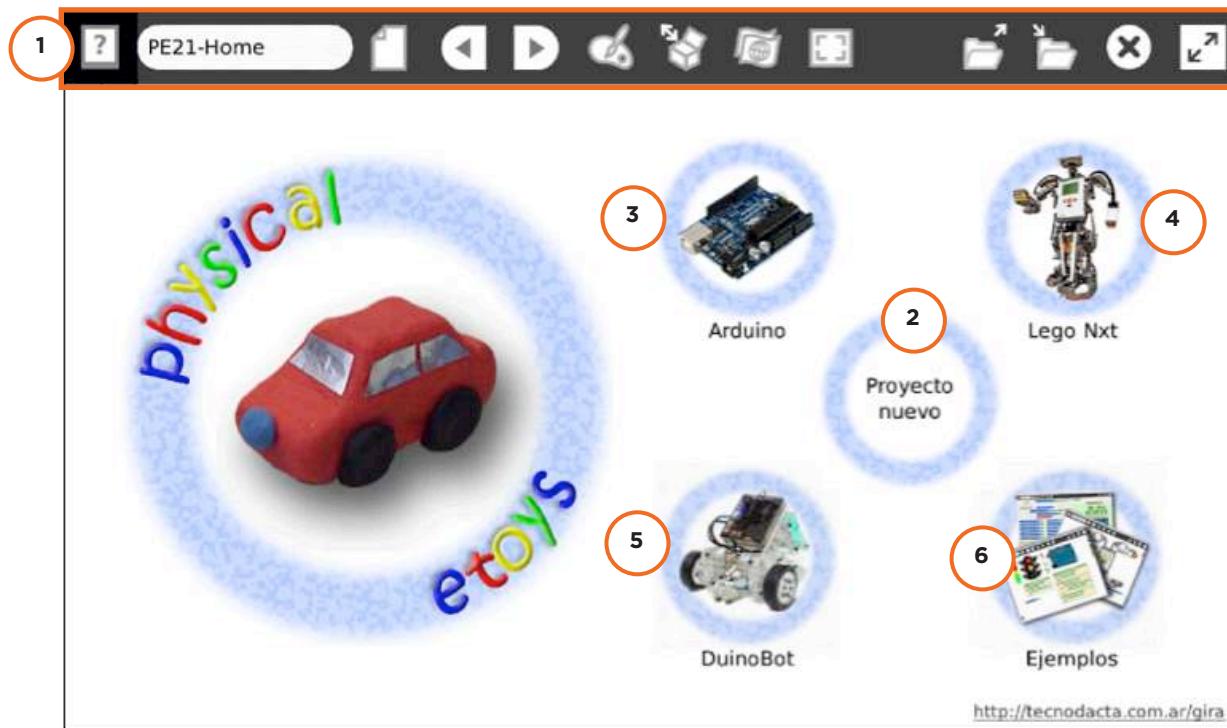


Se descomprime la carpeta y se ejecuta el archivo **PhysicalEtoys.exe**.



► Pantalla de inicio

Cuando se accede al programa puede observarse la siguiente ventana con las opciones de trabajo:



1. **Barra de herramientas.**
2. **Proyecto nuevo:** crea un proyecto nuevo.
3. **Arduino:** forma de conectar y trabajar con placa Arduino.
4. **Lego Nxt:** forma de conectar y trabajar con kit de Lego Mindstorms NXT.
5. **DuinoBot:** forma de conectar y trabajar con kit de Robotgroup con placa Duinobot.
6. **Ejemplos:** trabajos pre-armados con ejemplos de aplicación.

► Selección de idioma

Physical Etoys ofrece variadas opciones de idiomas. Pulsa el botón **Seleccionar idioma**.

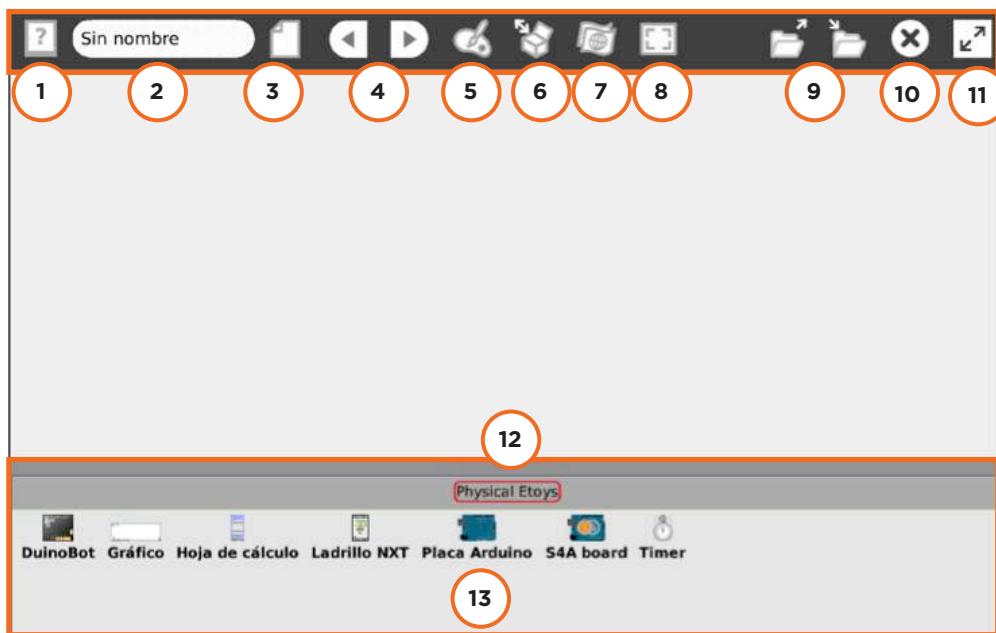


Dentro de las opciones elegir **Español**.



► Entorno de trabajo

Cuando se selecciona la opción **Proyecto nuevo** se abre la ventana principal en blanco, similar a la del programa Etoys, con la diferencia de un menú desplegable en la parte inferior de la misma que se abre al acercar el *mouse*.



1. Ayuda de Physical Etoys.
2. Nombre del proyecto nuevo.
3. Empezar un nuevo proyecto.
4. Navegación entre proyectos: anterior y siguiente.
5. Editor gráfico de objetos (ver Tutorial de Etoys en el Campus Virtual de Educación Digital - Tutoriales - Programar - Animaciones y videojuegos - Etoys)
6. Caja de Provisiones (ver Tutorial de Etoys en el Campus Virtual de Educación Digital - Tutoriales - Programar - Animaciones y videojuegos - Etoys)
7. Selector de lenguaje

8. Activa y desactiva la pantalla completa
9. Cargar y guardar un proyecto
10. Cierra el proyecto
11. Visualiza y esconde la barra de herramientas
12. Barra desplegable de objetos específicos para Physical Etoys
13. Objeto placa Arduino

► Agregar un objeto específico de robótica

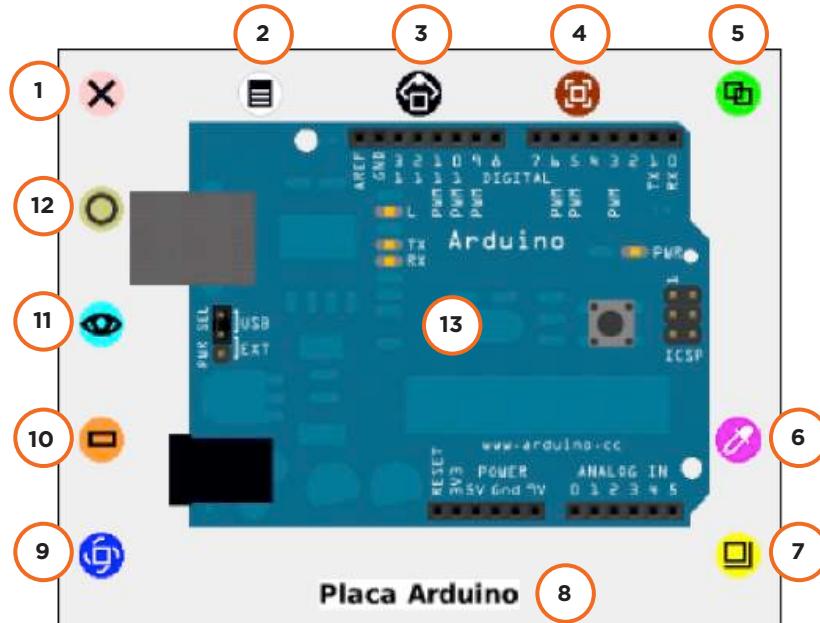
Los objetos que se encuentran en la **Caja de Provisiones** de la barra superior de herramientas son similares a los que se encuentran en el programa **Etoys** (ver [Tutorial de Etoys](#)).

Para agregar un objeto al proyecto, arrastrarlo hacia el entorno de trabajo (conocido como **mundo**).

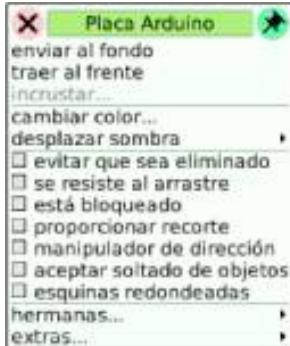


► Halos del objeto Arduino

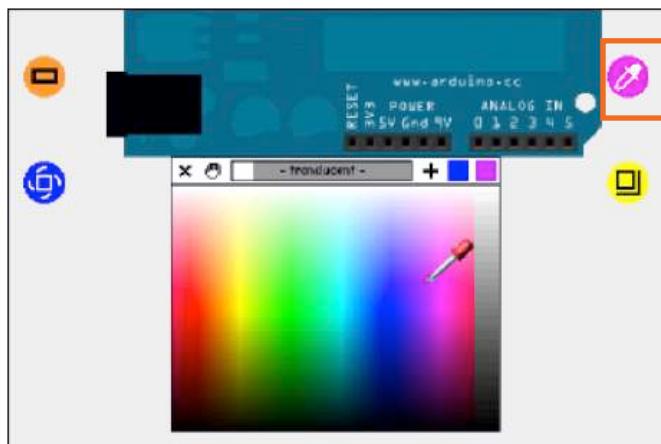
Al pulsar con el botón derecho del *mouse* sobre el objeto se abren los **halos** del mismo. Son pequeños íconos que rodean el objeto. Cada halo cumple una función diferente.



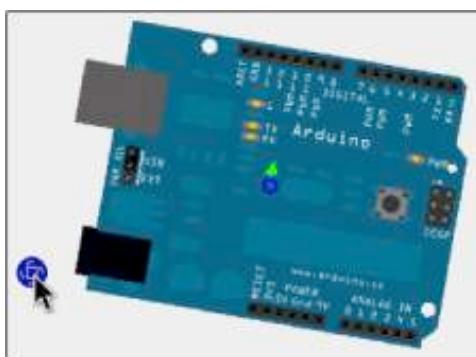
- 1. Mover al cubo de basura.** Elimina el objeto.
- 2. Menú.** Abre un menú desplegable que modifica diversas opciones del objeto.



- 3. Levantar.** Levanta el objeto para poder trasladarlo por el entorno de trabajo sin necesidad de mantener presionado el botón del *mouse*.
- 4. Mover.** Mueve el objeto para trasladarlo por el entorno de trabajo. A diferencia del anterior requiere mantener presionado el botón del *mouse* hasta su destino.
- 5. Duplicar.** Duplica el objeto.
- 6. Cambiar color.** Modifica el color del objeto.



7. **Cambiar tamaño.** Modifica el tamaño del objeto.
8. **Nombre del objeto.** Se puede editar el nombre del objeto al pulsar sobre él.
9. **Rotar.** Permite rotar el objeto.



10. **Hacer un mosaico.** Genera una tesela (mosaico) que representa al objeto.
11. **Abrir un visor.**
12. **Colapsar.** Permite minimizar el objeto. Produce un efecto similar a si se cerrara, pero sigue abierto y activo, simplemente desaparece de la ventana.



13. **Objeto**

► Visor del objeto Arduino

Cuando se pulsa sobre el halo **Abrir un Visor** se despliega una lista de órdenes en un menú, agrupadas por categorías.

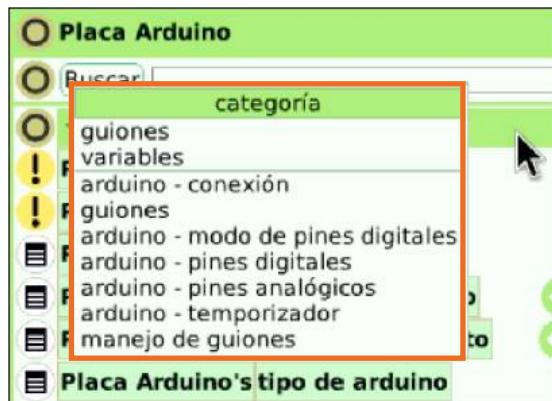
Cabe destacar que cada objeto, según sus potencialidades, tiene órdenes en común con otros objetos y órdenes propias.



1. Conecta y desconecta la placa Arduino al programa.
2. Indica si la placa Arduino está conectada y si el programa la reconoce.
3. Nombre del puerto al cual está conectada la placa Arduino.
4. Tipo de placa Arduino conectada.
5. Flecha que despliega las categorías del objeto.
6. Nombre de una categoría.

► Categorías del objeto Arduino

Cada orden del cuadro que aparece al presionar el halo **Visor** se encuentra dentro de una categoría. Como en el caso de las órdenes existen categorías comunes a algunos objetos y otras propias. Para poder ver todas las categorías de un objeto, se pulsa sobre la flecha desplegable de cualquier categoría.



Paso a paso

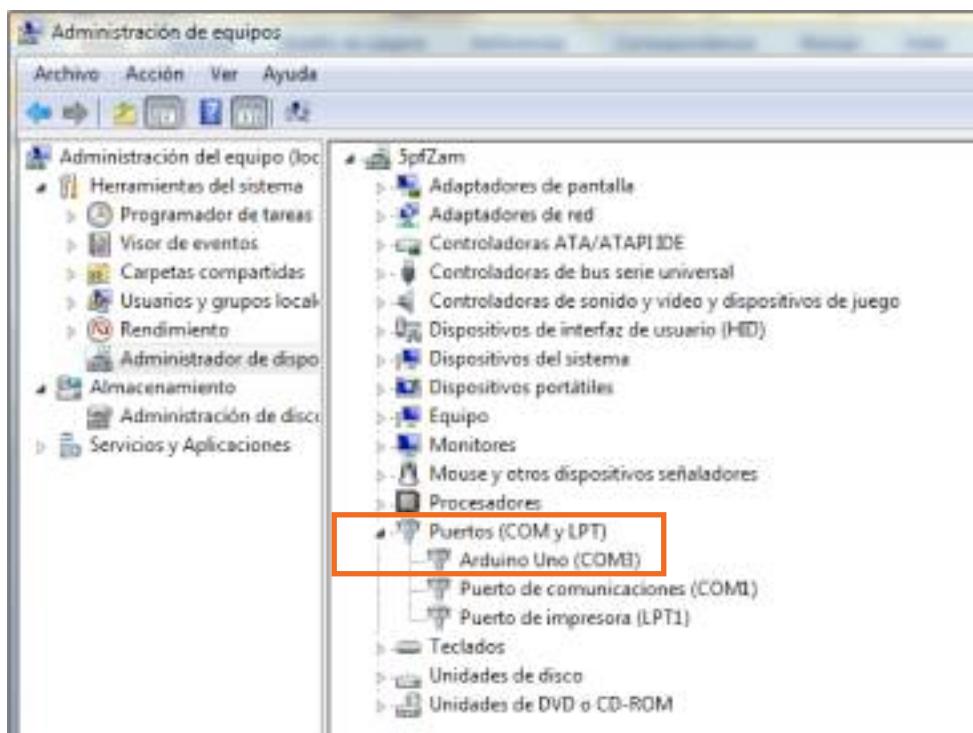
► Conectar la placa Arduino con Physical Etoys

Conectar a través de un cable USB la placa Arduino a la computadora que tiene instalado el programa Physical Etoys.

Para que la computadora reconozca la placa, es necesario instalar el programa Arduino que se descarga desde la página <https://www.arduino.cc/en/Main/Software>

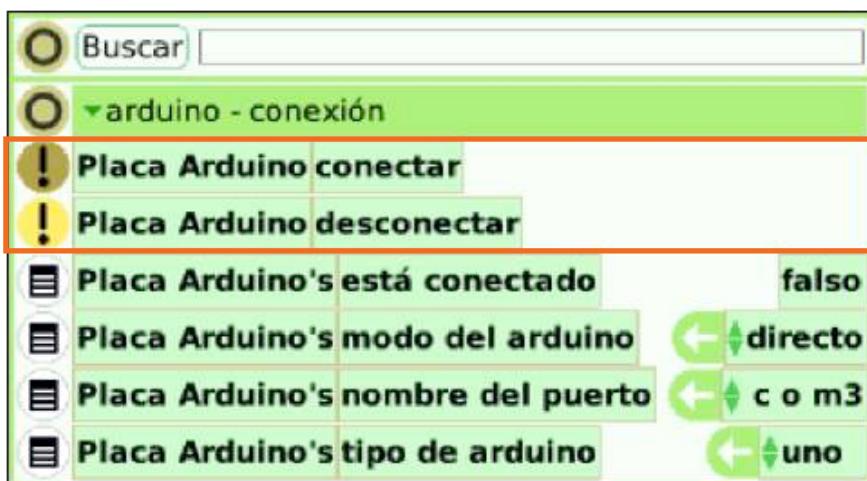


Es importante saber el **puerto COM** en la cual está conectada la placa. Acceder desde el **Panel de Control** al **Administrador de dispositivos**.



Al abrir el programa Physical Etoys es necesario conectar la placa con el programa. Para esto se arrastra al mundo el objeto **Placa Arduino**. Se abren los halos y se selecciona el **Visor** para acceder a las categorías de las órdenes.

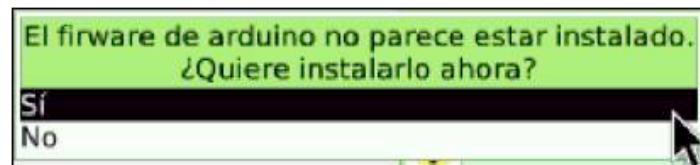
En la categoría **Conexión** está la orden que conecta la placa y otra que la desconecta. Presionar sobre el ícono delante de la orden.



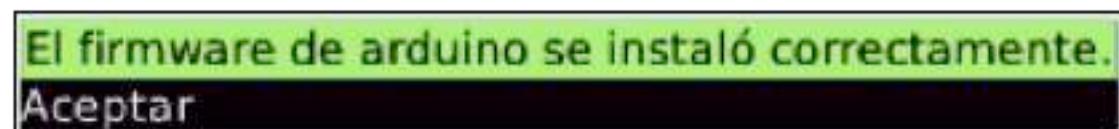
Estas órdenes se ejecutan una sola vez con el fin de conectar o desconectar la placa del programa.

Es importante que el tipo de Arduino y el nombre del puerto estén configurados correctamente.

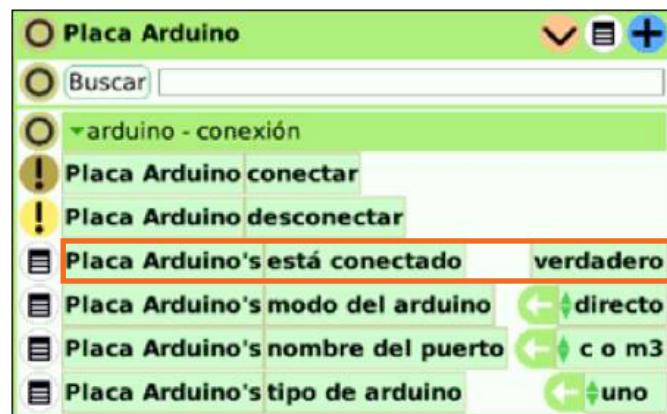
Es probable que al conectar por primera vez la placa, pida actualizar el Firmata.



Se muestra una ventana de información una vez instalado correctamente. En caso de existir algún mensaje de error, repetir la operación.



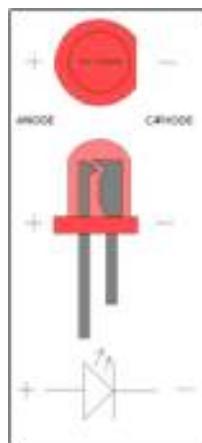
Si la placa está conectada, la orden **Placa Arduino's está conectado** se mostrará en **Verdadero**.



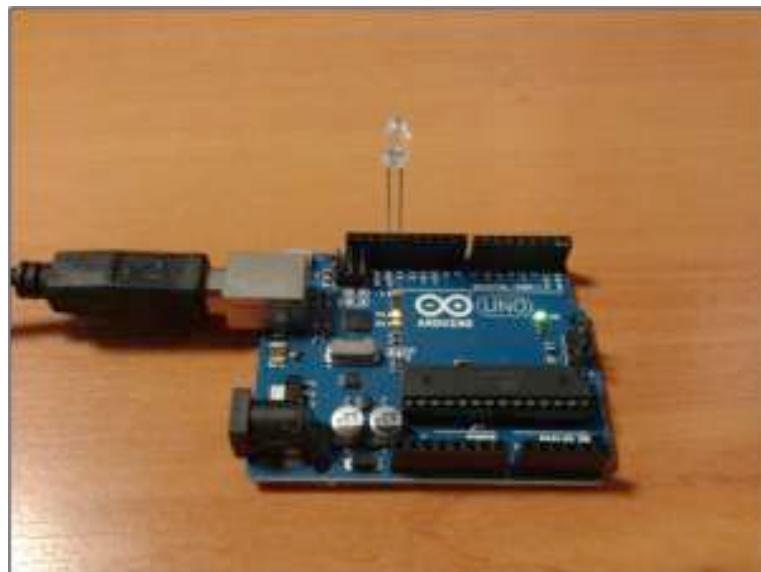
► Hacer titilar un led

Un **led** (*light emitting diode* - diodo emisor de luz) es una pequeña lámpara que emite luz. Se puede controlar su encendido, apagado e intensidad conectándolo a un pin de la placa Arduino y programándolo con Physical Etoys.

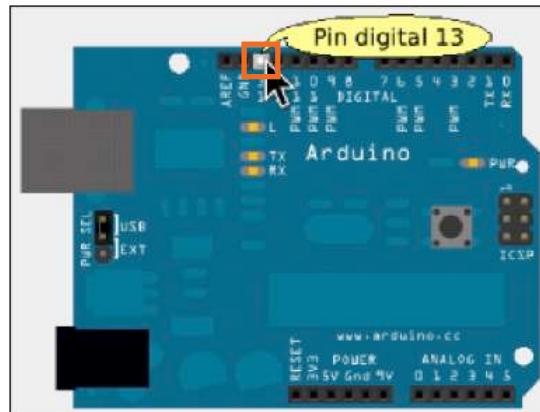
Un led tiene polaridad, es decir, una de sus patas tiene que estar conectada al positivo y la otra al negativo.



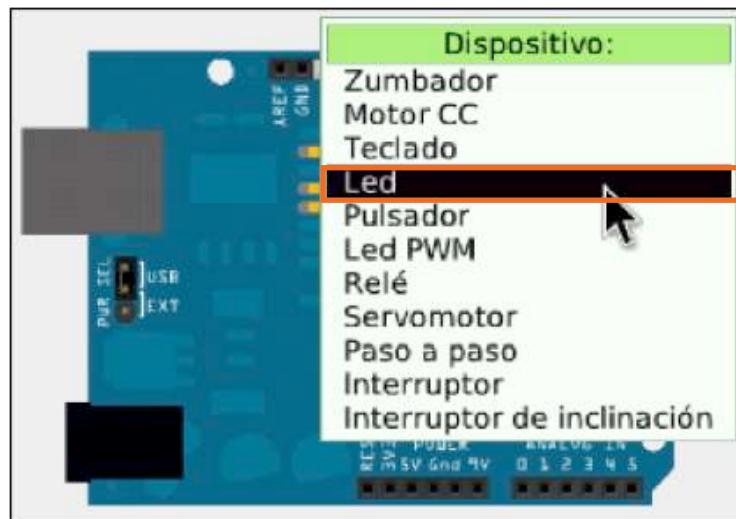
Conectar la pata positiva al pin 13 de la placa Arduino y la pata negativa al GND (ground = tierra = negativo).



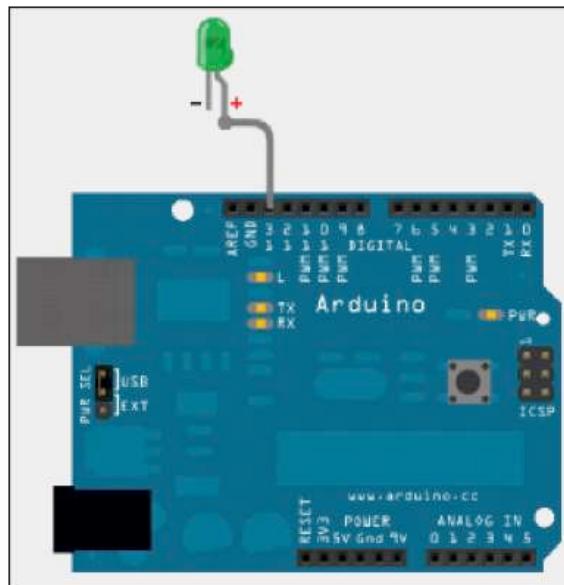
En Physical Etoys colocar el puntero del *mouse* sobre el **pin 13** y presionar el botón derecho del *mouse*.



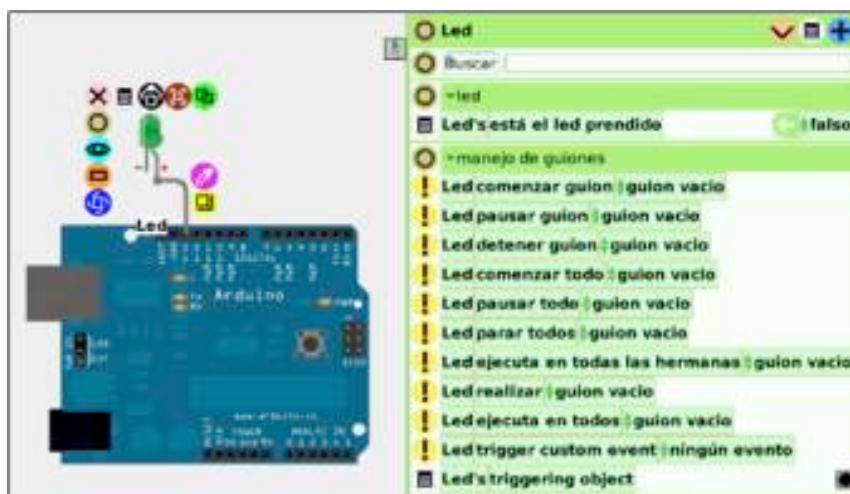
Se abre un menú contextual con una lista de dispositivos posible de conectar en dicho pin. Elegir el led.



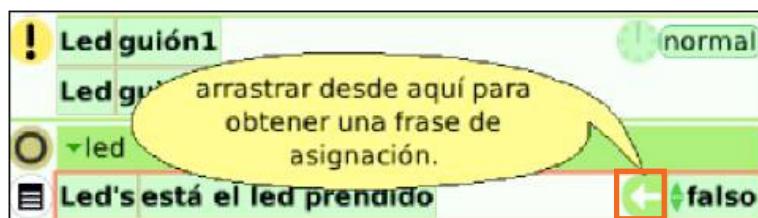
Se conectará virtualmente el led en la placa.



Para poder controlar el led es necesario seleccionarlo y abrir el halo **Visor** para acceder a las órdenes que programen su comportamiento.



Arrastrar la orden **Led's está el led prendido** al entorno de trabajo desde la flecha de asignación para poder brindarle a la orden un valor.

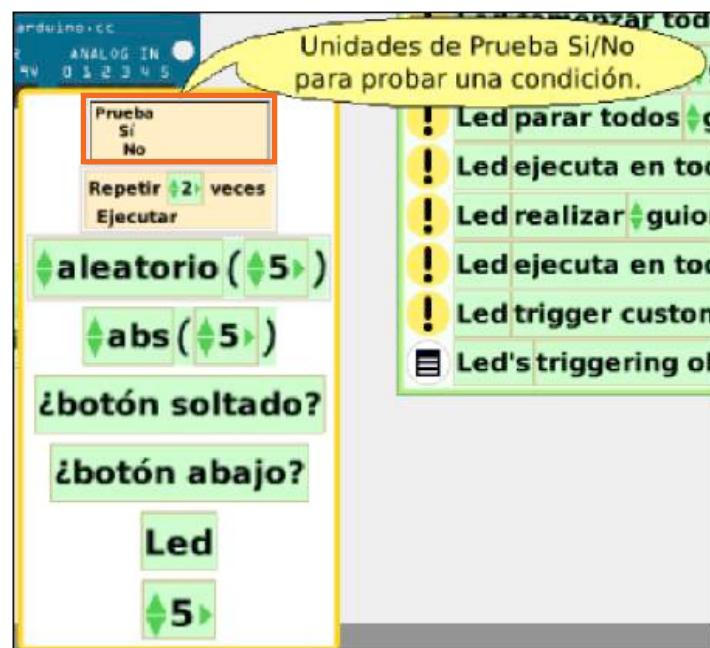


Al arrastrar esta orden al **mundo** (entorno de trabajo) se abre un **guión** (programa) nuevo.

Dentro del cuadro de guión se encuentra un ícono con forma de caja que contiene herramientas útiles para la programación.

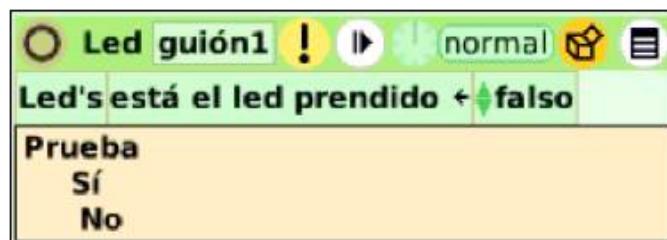


Para hacer titilar el led se necesita utilizar un condicional.



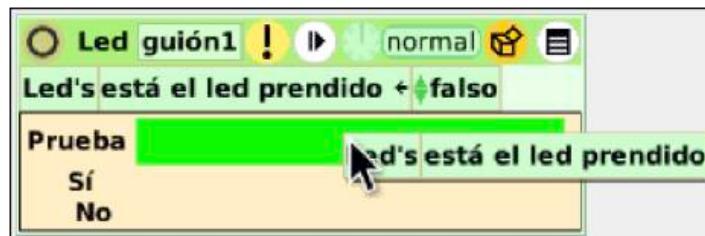
En un condicional se presenta una afirmación (**Prueba**). En el caso que sea verdadera (Sí) se ejecutan ciertas acciones y en el caso que sea falsa (**No**) se ejecutan otras acciones.

Se arrastra el cuadro condicional dentro del guión (programa).

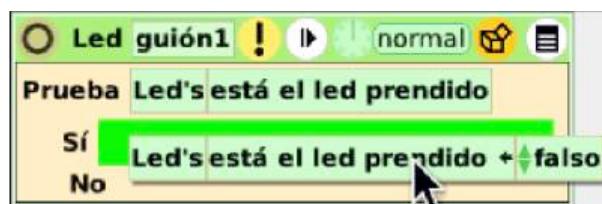


En este caso la afirmación es **Led's está el led prendido**. Si esta condición es verdadera se apaga el led. Si es falsa quiere decir que está apagado, entonces se prende el led.

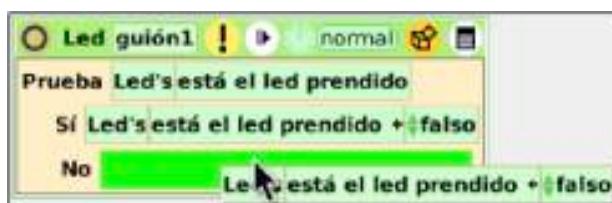
Arrastrar la orden **Led's está el led prendido** desde el cuadro del visor **pero esta vez no tomarlo por la flecha de asignación**. Colocarla al lado de la palabra **Prueba** en el cuadro condicional.



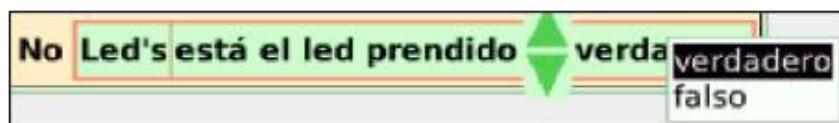
Mover la asignación arrastrada anteriormente desde la flecha de asignación al lado de la palabra **Sí** en el cuadro condicional y dejarle asignado el valor **falso**.



Arrastrar una nueva orden **Led's está el led prendido** desde el cuadro del visor, pero esta vez desde la flecha de asignación nuevamente, ya que se le asignará un valor.



Modificar el valor de asignación de falso a verdadero.



Quedará programado el led para que encienda si lo encuentra apagado y lo apague si lo encuentra encendido.



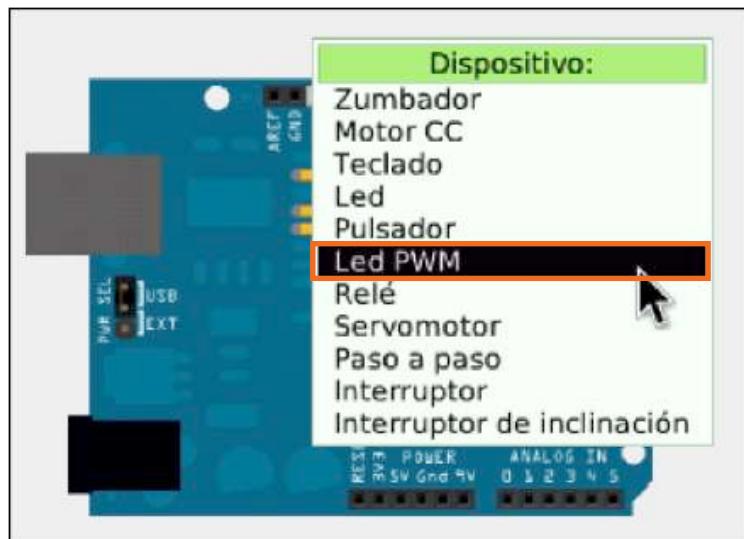
Para ejecutar este programa pulsar el ícono con forma de reloj. La palabra **normal** se cambiará por **latiendo**, lo que indica que el programa se está ejecutando.

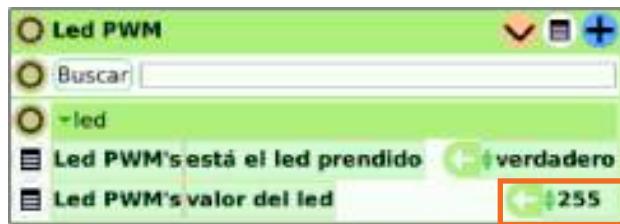


► Modificar la intensidad de un led

Con la construcción y programación anterior se puede prender o apagar un led pero no regular la intensidad de luz.

Para esta acción es necesario agregar en el pin otro led diferente al anterior: un **led PWM**. La imagen gráfica del led que aparece en el entorno de trabajo es similar. La diferencia principal es que el led común sólo puede tomar dos valores (verdadero o falso) y el led PWM puede tomar valores entre 0 y 255.





Para completar la acción será necesario abrir la **Caja de Provisiones** en la parte superior de la pantalla y arrastrar al mundo la **Barra de desplazamiento**.



Abrir el halo **Visor** de la **Barra de desplazamiento** de la misma forma que los otros objetos. Se despliegan órdenes específicas para este objeto.

Una de las órdenes dentro de la categoría **básico**, es el valor numérico que toma la barra a medida que se va moviendo el dispositivo deslizable. Estos valores varían entre 0 y 1, tomando en su trayecto, diferentes valores decimales con el fin de obtener más precisión.



Para comenzar a crear el guión (programa) arrastrar desde las órdenes del **Visor** del **led PWM** la orden **Valor del led** desde la **flecha de asignación**, ya que se le asignará un valor al mismo.



Desde las órdenes del **Visor** de la **Barra de desplazamiento**, arrastrar el valor numérico de la misma y reemplazar el valor 255 en el valor del led.



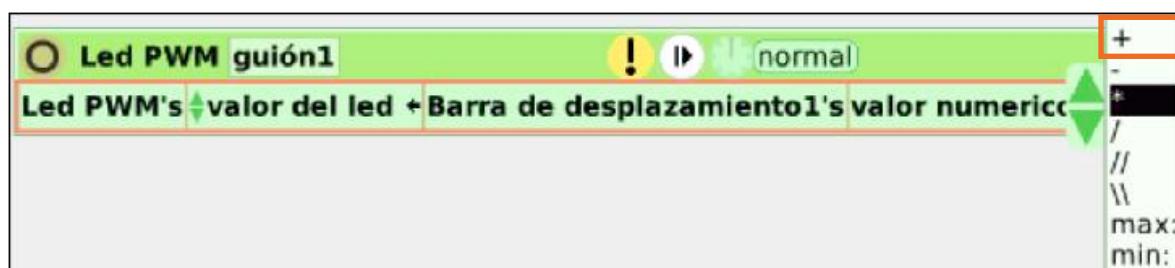
De esta forma el led va a tomar el valor que determine el movimiento de la **Barra de desplazamiento**.

Como el valor que puede tomar un led varía entre 0 y 255, y el valor al que puede llegar la **Barra de desplazamiento** es hasta 1 se debe multiplicar el valor brindado en la barra por **255** (valor máximo al cual se desea llegar) para equiparar los valores de ambos objetos.

Para esto pulsar en la flecha verde al lado de **Valor del led**. Puede observarse un **+ 1**.



Modificar el signo **+** por el de multiplicación ***** y el valor **1** por el de **255**.



Ejecutar el programa presionando el ícono con forma de reloj, cambiando su estado **Normal** por **Latiendo**.

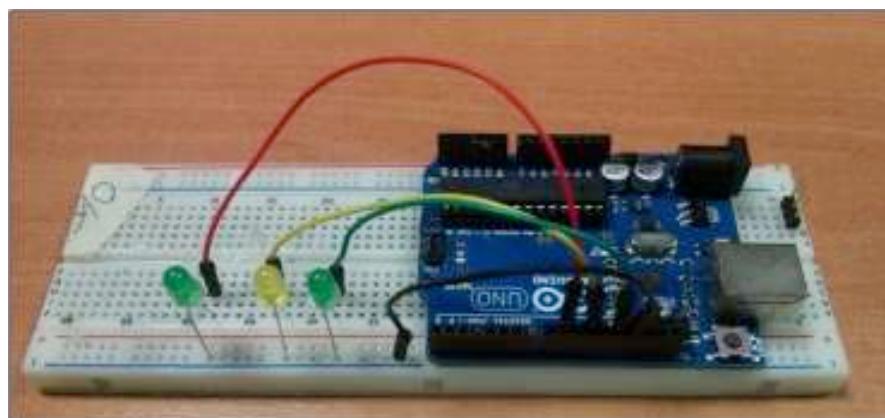
► Simular un semáforo con 3 leds

Conectar el pin GND (ground = negativo) de la placa mediante cable con punteras a la línea negativa de la Protoboard.

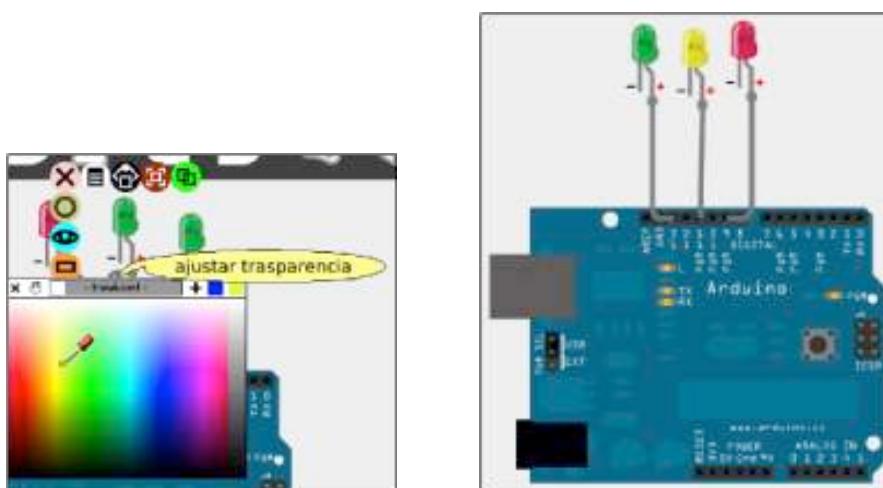
Luego conectar a dicha línea las patas negativas de los tres leds, mientras que las patas positivas se conectan a una línea del centro de la protoboard.

Conectar mediante un cable con punteras el positivo del led rojo al pin 9, el del led amarillo al pin 11 y el del verde al pin 13.

La conexión debe quedar similar a la siguiente imagen:



Simular la conexión en el programa Physical Etoys utilizando leds comunes, ya que sólo se van a prender y apagar. Modificar los colores de los leds con el halo **Color**



Para que la programación resulte más sencilla es conveniente modificar el nombre de cada objeto led desde el halo **Nombre**.



Programar la lógica de los leds para que se encienda en primer lugar el rojo, luego el amarillo y por último el verde.



En el ejemplo se utilizaron bloques de repetición para retrasar el tiempo de prendido y apagado.

La luz amarilla tiene menos bucles de repetición para que se mantenga prendida menos tiempo que la luz roja y verde.

Para que la intermitencia sea más lenta, se cambian las pulsaciones (latidos) por segundo del guión. Por defecto, el guión se ejecuta a 8 latidos por segundo. Este valor se modifica manteniendo presionado el reloj de ejecución para que se despliegue un cuadro de latidos por segundo.



► Mover un servomotor con la herramienta joystick

Physical Etoys ofrece herramientas virtuales que permiten controlar los objetos físicos conectados en la placa Arduino.

Diferencia entre un motor de corriente continua (CC o DC) y un servomotor:

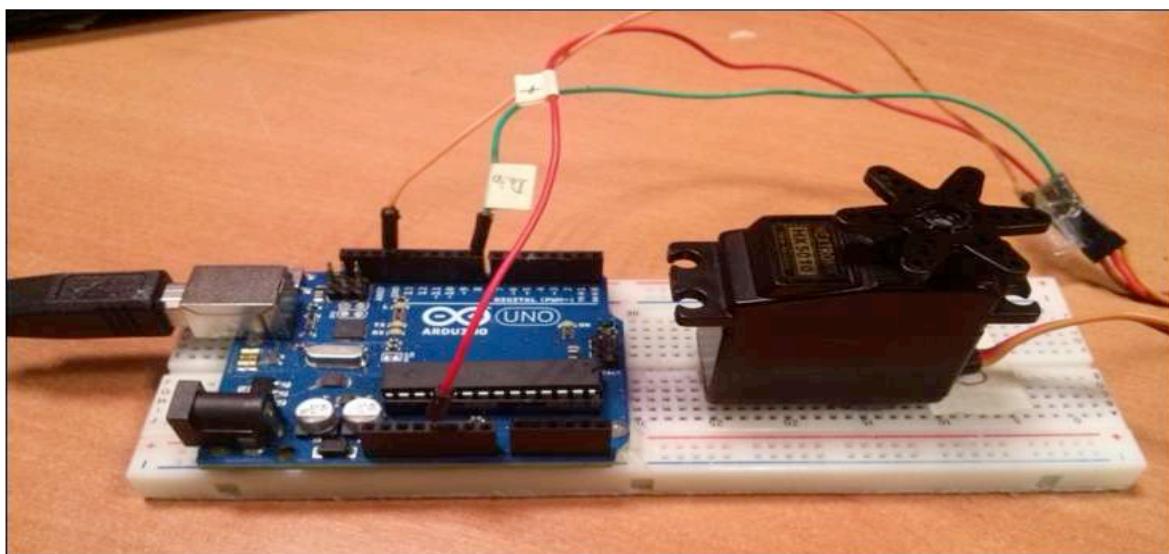
Los motores de corriente continua (CC) tienen dos cables: positivo y negativo. Al conectarlos a la alimentación giran de forma rápida y continua, hasta que se les quite esa alimentación.

Con los motores servos se tiene un mejor control de la graduación de giro del mismo. Generalmente tienen tres cables: positivo, negativo y un tercer cable de control. Están destinados a tareas más específicas de precisión.

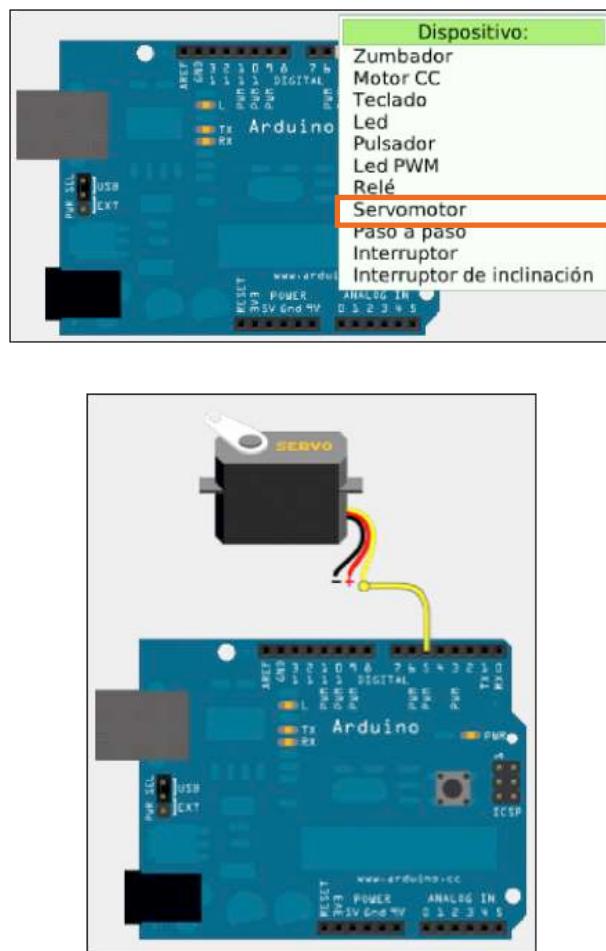
Conectar un servomotor en la placa Arduino de la siguiente forma:

- Conectar el cable negativo del motor al pin GND (ground = negativo) de la placa.
- Luego conectar el cable positivo al pin 5V (5 volts = alimentación) de la placa.
- Conectar el cable de datos al pin 5 de la placa.

La conexión deberá quedar similar a la siguiente imagen:



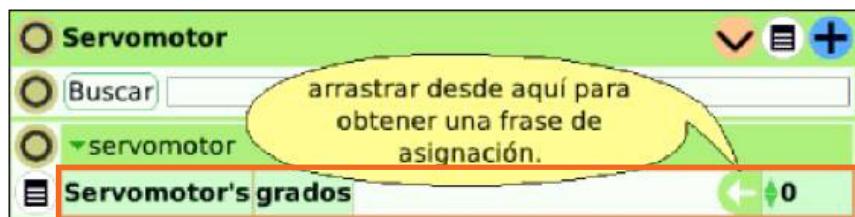
Simular la conexión en el programa Physical Etoys.



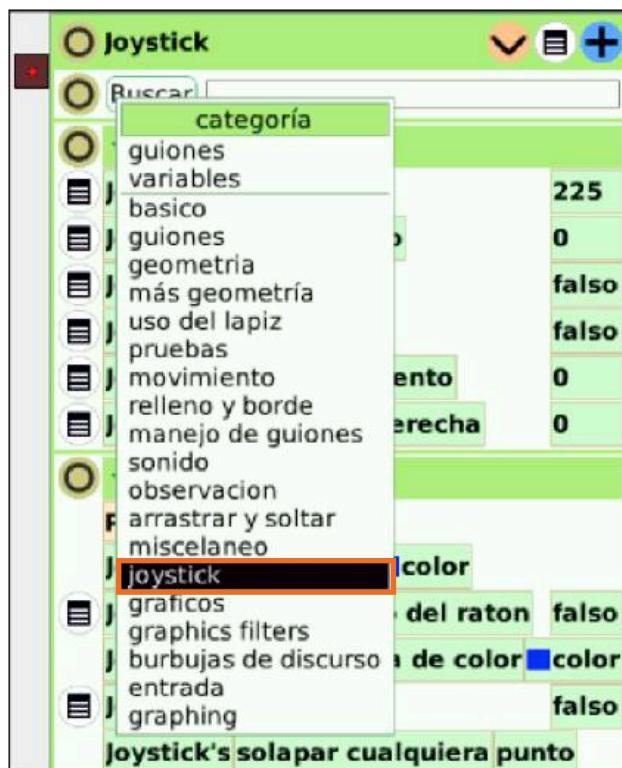
Agregar la herramienta **Joystick** desde la **Caja de Provisiones**.



Abrir el halo **Visor** del **Servomotor** para que aparezcan los comandos y órdenes para programarlo. Arrastrar el comando **Servomotor grados** desde la flecha de asignación al **mundo** ya que le asignaremos un valor.



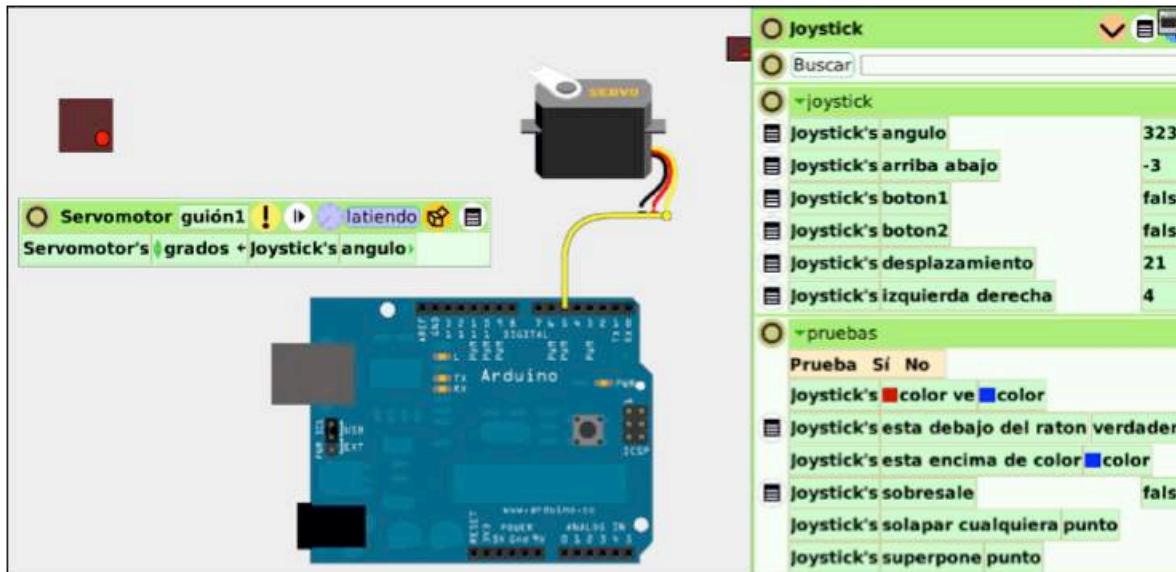
Abrir el halo **Visor** del **Joystick** y dentro de las categorías seleccionar **Joystick**.



Asignar el valor del ángulo del joystick a los grados del servomotor.

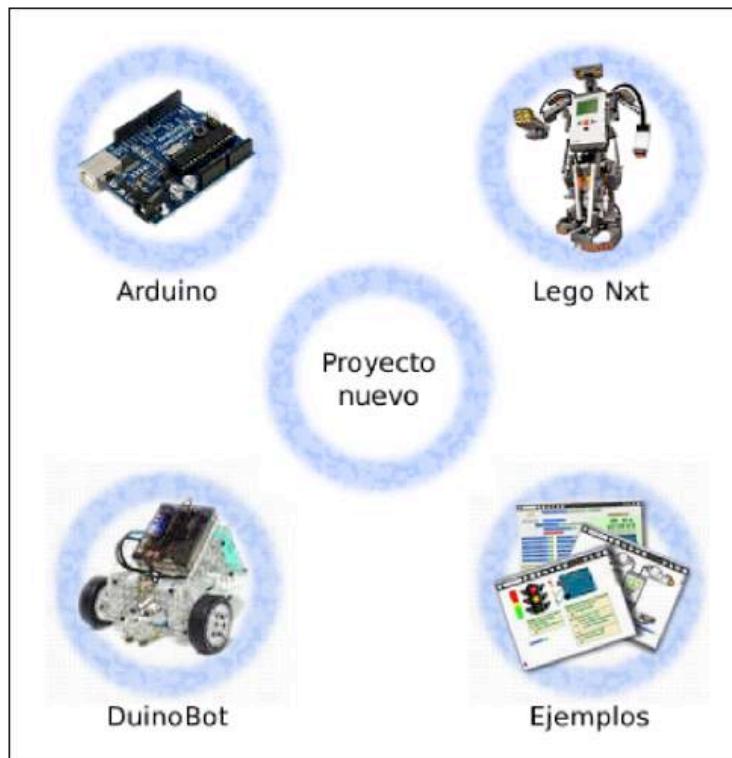


Mover el punto rojo del centro del joystick para comandar la dirección de giro del servomotor.



► Ayuda en Physical Etoys

Desde el programa existen diversas formas de obtener ayuda. En la página de inicio se encuentran instructivos de conexión con diferentes kits de robótica.



Arduino-Example

El "Hola Mundo" de la Electrónica

Empieza seleccionando el tipo y el puerto serie correspondiente a tu placa Arduino y haz clic en "Conectar".

Arduino board's arduino type = **uno**

Arduino board's port name = **c o m1** **Conectar** **Desconectar**

Estado de la conexión: **Desconectado**

Luego ejecuta el siguiente guión haciendo clic en el ícono correspondiente a un reloj.

Led script1 ! normal

Prueba Led's está el led prendido
Sí Led's está el led prendido +**falso**
No Led's está el led prendido +**verdadero**

Arduino

Click here to learn more!

También se puede acceder a ejemplos para comenzar a trabajar a partir de producciones finalizadas.

Cargá un proyecto

...examples

- Arduino-StreetLight.002.pe
- Arduino-TrafficLight.006.pe
- DuinoBot-Concurrency.002.pe
- DuinoBot-Graph.004.pe
- DuinoBot-Greenhouse.003.pe
- DuinoBot-Motors.008.pe
- DuinoBot-RemoteControl.002.pe
- LegoNxt-Car.005.pe
- LegoNxt-DodgeWalls.006.pe
- Translation.003.pe



Arduino-LuzCallejera

Este proyecto muestra el comportamiento de una luz callejera que detecta automáticamente si tiene que estar prendida o apagada dependiendo de la luz ambiente.

By Richo

arduino, led, photoresistor, directo

Abrir

Enlaces de interés

Sitio oficial: <http://tecnodacta.com.ar/gira/projects/physical-etoys/>

Videos sugeridos desde la página oficial: <http://tecnodacta.com.ar/gira/videos/>

Contacto:
ayuda.pedagogico.digital@bue.edu.ar



Esta obra se encuentra bajo una Licencia Attribution-NonCommercial-ShareAlike 2.5 Argentina de Creative Commons. Para más información visite <http://creativecommons.org/licenses/by-nc-sa/2.5/ar/>

