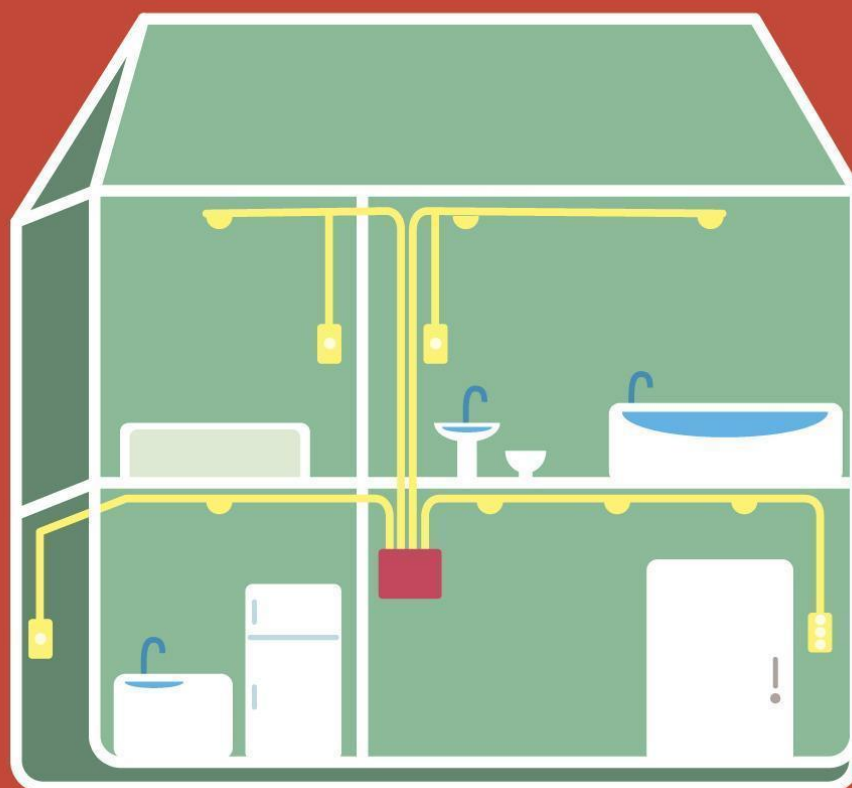


SABERES DIGITALES



AUTOMATIZACIÓN DE SISTEMA LUMÍNICO

AUTORIDADES

Presidente de la Nación

Mauricio Macri

Vicepresidenta de la Nación

Marta Gabriela Michetti

Jefe de Gabinete de Ministros

Marcos Peña

Ministro de Educación, Cultura, Ciencia y Tecnología

Alejandro Finocchiaro

Titular de la Unidad de Coordinación General del Ministerio de Educación, Cultura, Ciencia y Tecnología

Manuel Vidal

Subsecretario de Coordinación Administrativa

Javier Mezzamico

Director Ejecutivo INET

Leandro Goroyesky

Gerenta General de EDUCAR Sociedad del Estado

Liliana Casaleggio

Directora Nacional de Asuntos Federales

María José Licio Rinaldi

Director Nacional de Educación Técnico - Profesional

Fabián Prieto

Coordinador de Secundaria Técnica

Alejandro Anchava

Responsable de Formación Docente Inicial y Continua INET

Judit Schneider

Coordinador General En FoCo

Pablo Trangone

AUTORIDADES	¡Error! Marcador no definido.
AUTOMATIZACIÓN DE SISTEMA LUMÍNICO	3
Ficha técnica	3
Presentación	4
Desarrollo	5
Nivel Inicial	5
Paso 1 - Conectar el relé	5
Paso 2 - Programar el encendido y apagado de las luces	6
Paso 3 - Subir el código a la placa Arduino	9
Paso 4 - Conectar el sensor piroeléctrico	10
Paso 5 - Calibración del sensor piroeléctrico	11
Paso 6 - Activar el envío de datos a la consola	12
Paso 7 - Agregar el módulo relé	13
Paso 8 - Encender las luces en función del movimiento	14
Nivel Avanzado	15
Paso 1 - Conectar la fotorresistencia (sensor LDR)	15
Paso 2 - Obtener la lectura del sensor LDR	16
Paso 3 - Encender y apagar las luces en función del sensor LDR	17
Cierre	19
Glosario	20
Reconocimientos	¡Error! Marcador no definido.

AUTOMATIZACIÓN DE SISTEMA LUMÍNICO

Ficha técnica

Nivel educativo	Secundario. Ciclo Básico.
------------------------	---------------------------

Descripción general	Diseño y construcción de una maqueta/prototipo de un sistema de iluminación domiciliario.
Niveles de complejidad	Nivel inicial: instalación y programación de un sistema de luces de encendido automático con sensor de movimiento. Nivel avanzado: incorporación de un sensor del nivel de luz del exterior en función de cuyas mediciones se encenderá o apagará el sistema.

Insumos	<ul style="list-style-type: none">• 1 x Arduino UNO R3• 1 x Protoboard• 1 x Cable USB tipo B• 1 x Fuente de 9V 1A (plug centro positivo, 5.5x2.1mm)• 20 x Cables dupont macho hembra• 20 x Cables dupont macho macho• Filamento para impresora 3D (PLA)• 1 x Módulo Relé• 1 x Sensor PIR• 1 x Sensor LDR <p>Extra</p> <ul style="list-style-type: none">• Hasta 4 lámparas de 220V y portalámparas• Enchufe y cable.
Herramientas	<ul style="list-style-type: none">• Computadora• Soldador• Estaño• Alicata• Pinza de punta

	<ul style="list-style-type: none"> • Brusela
Otros requisitos	<ul style="list-style-type: none"> • Conexión a internet • Descargar el programa “mblock3” http://www.mblock.cc/software-1/mblock/mblock3/

Presentación

Descripción ampliada del proyecto

Se propone armar la maqueta/prototipo de un sistema automatizado de iluminación para un hogar. Este contará con un sensor de movimiento que detectará la presencia de personas en los ambientes. En base a esta medición, el sistema encenderá la luz cuando los espacios se encuentren habitados y la apagará cuando no lo estén.

En el nivel avanzado, se incorpora un sensor LDR que permitirá determinar la cantidad de luz que proviene del exterior. El sistema se programará para que las luces del interior se enciendan o se apaguen automáticamente en función de la medición del sensor.

Al final de esta guía se puede encontrar un glosario donde se provee la información técnica necesaria para poder poner el proyecto en funcionamiento. El mismo cuenta con aclaraciones sobre los diversos elementos electrónicos involucrados así como también conceptos claves

Objetivos

- Aproximarse al conocimiento y al manejo de distintos componentes electrónicos mediante la construcción de un sistema automatizado de encendido y apagado de luces.
- Introducirse en el armado de circuitos utilizando placas Arduino, sensores de movimiento y actuadores.
- Analizar y desarrollar la programación de estructura secuencial de un programa que permita el encendido y apagado de luces del interior de una casa en función de la presencia de personas en los ambientes (nivel inicial) y de las condiciones lumínicas del exterior (nivel avanzado).

Desarrollo

Nivel Inicial

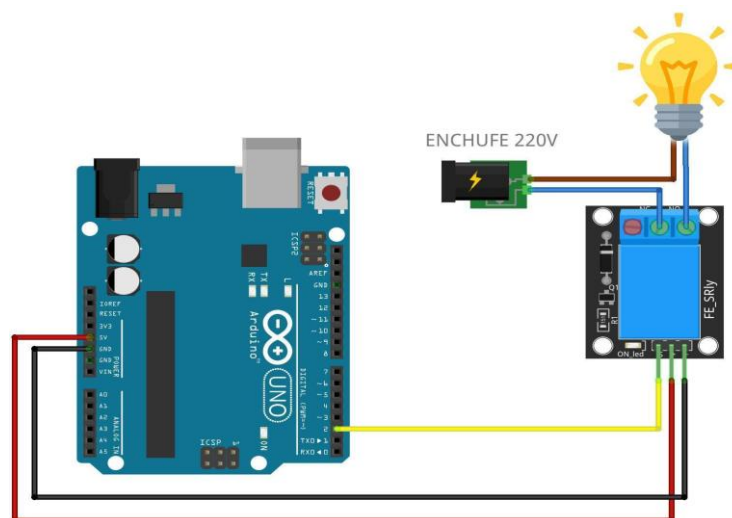
Agustín es un chico que se preocupa mucho por el cuidado ambiental. Por eso, tuvo la idea de aplicar sus conocimientos en programación y electrónica para desarrollar un sistema de ahorro de energía para su hogar, que controle el encendido y apagado de las luces de manera automática.

Este cuenta con un sensor PIR de movimiento que detecta el ingreso y el egreso de las personas de cada habitación, permitiéndole encender las luces solamente cuando haya alguien adentro. De esta forma sólo se utilizará energía en iluminación cuando alguien lo necesite y se eliminará el riesgo de olvidar de apagar las luces cuando las personas salgan de los ambientes de la casa.

En esta instancia del proyecto, se propone utilizar un sensor PIR (piroeléctrico) que detecta el movimiento de personas, y permite la automatización del encendido y el apagado de las luces, utilizando un módulo relé para la conexión.

Paso 1 - Conectar el relé

Para armar el sistema de luz automatizado y poder optimizar el consumo energético de un hogar, controlaremos el encendido y apagado de las luces a través de una placa Arduino y un módulo relé.



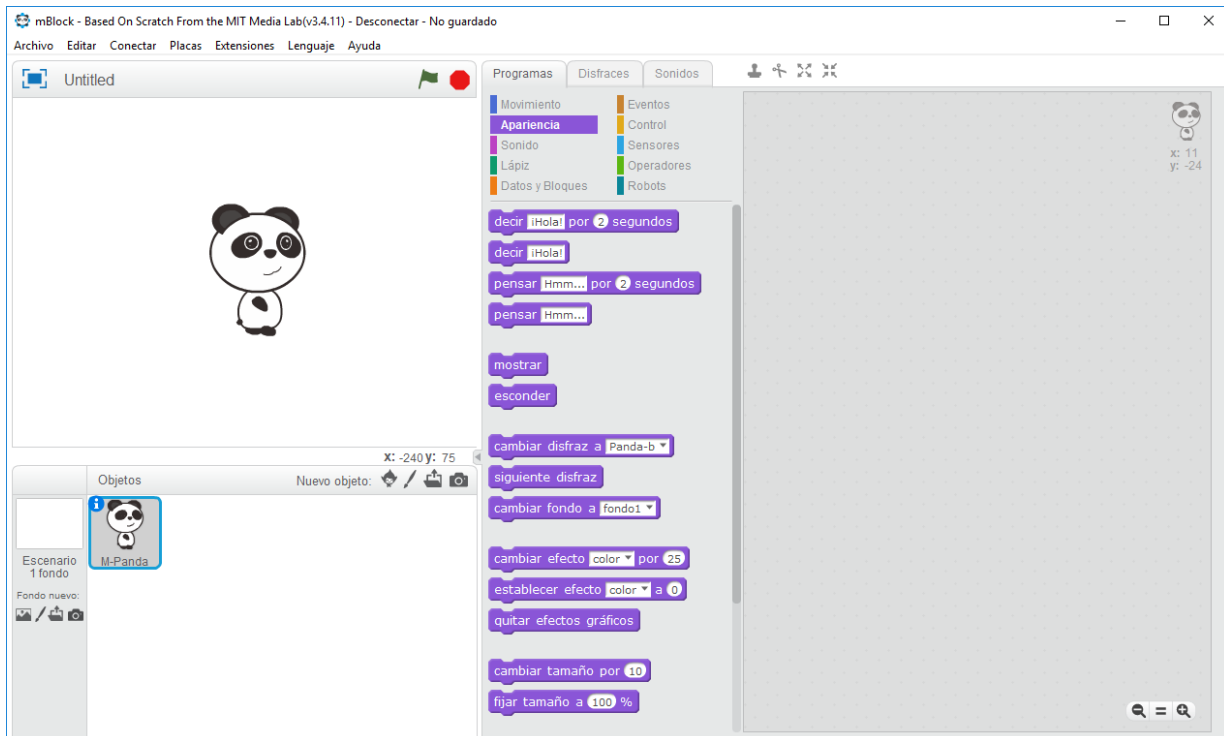
¡Atención! Para construir este dispositivo trabajaremos con una tensión de 220V. Si se está trabajando con protoboard, se recomienda no incluir en el mismo las conexiones a relé y 220V.

El código que subamos a la placa Arduino controlará la apertura y el cierre del relé y, en consecuencia, el encendido y apagado de las luces.

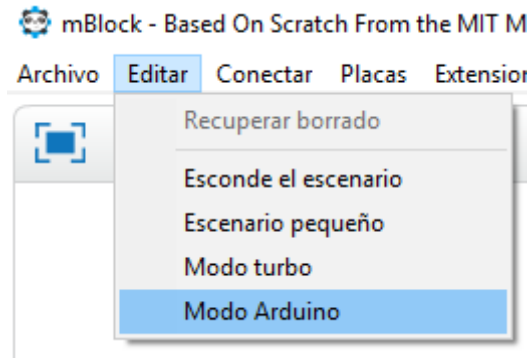
Paso 2 - Programar el encendido y apagado de las luces

La programación la realizaremos con mBlock3, entorno de programación basado en Scratch2 que permite programar proyectos de Arduino utilizando bloques. Pueden descargarlo siguiendo este enlace: <http://www.mblock.cc/software-1/mblock/mblock3/>

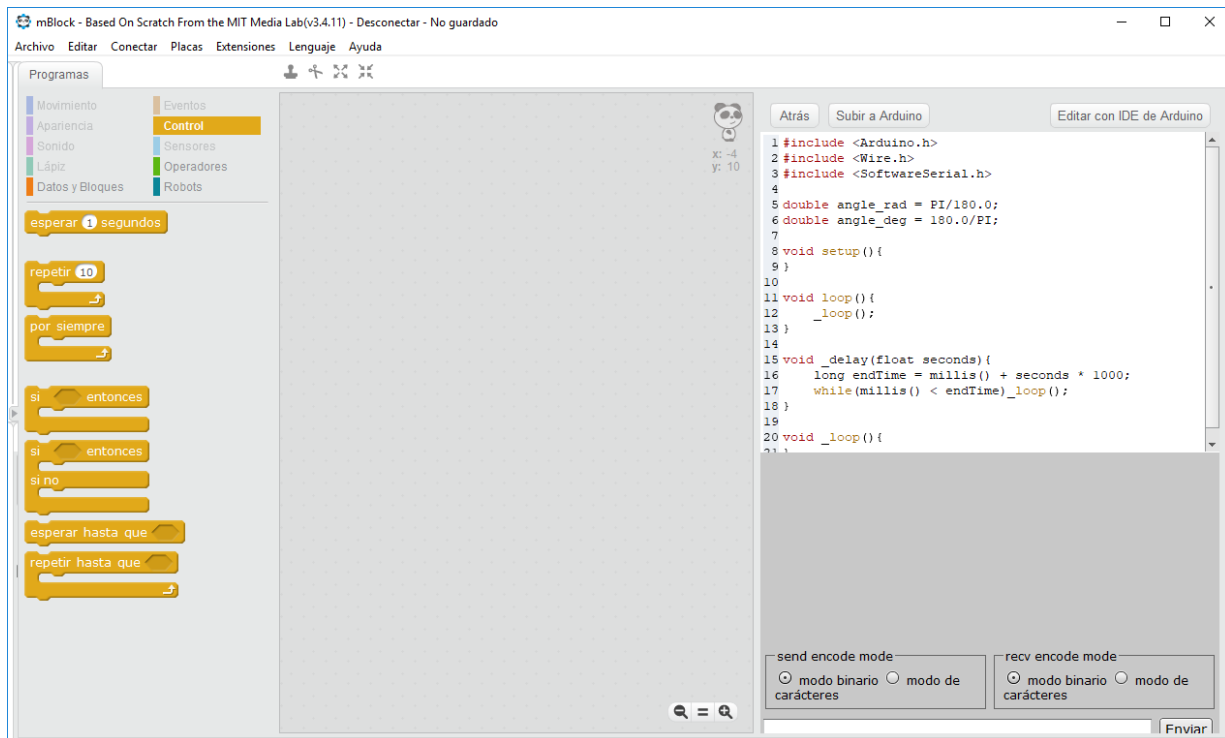
Cuando abrimos mBlock3, veremos una pantalla como la siguiente:



Para programar un proyecto de Arduino con mBlock3 debemos seleccionar el “Modo Arduino” desde el menú.



Al seleccionar este modo, el programa cambiará de aspecto. Se verá un área en el centro que es la que utilizaremos para programar con bloques. A la derecha se verá un campo donde aparecerá el código escrito que le corresponde a los bloques que están en el centro. Este código se irá escribiendo automáticamente a medida que se vaya armando el programa con los bloques.



Los bloques están agrupados por categorías. En este caso, se usarán las siguientes: “**Robots**”, “**Control**”, “**Operadores**” y “**Datos y Bloques**”. Cuando seleccionamos una de estas categorías, se pueden visualizar todos los bloques que pertenecen a ese grupo.



Para poder encender las luces debemos accionar el módulo relé. Realizaremos una prueba de encendido y apagado intermitente para verificar que todo funcione de forma correcta.



Al asignar el estado “ALTO” a la salida del relé (“*pin* digital 2”) estamos indicando que se aplique una tensión de 5V al relé, de manera que este permita el paso de corriente para que la lámpara se encienda.

Al asignarle el estado “BAJO”, estamos indicando que no se aplica tensión al módulo relé, de modo que no permita el paso de corriente y la lámpara se apague.

Veremos que a la derecha se muestra el código escrito que corresponde a este programa:

```
#include <Wire.h>
#include <SoftwareSerial.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;

void setup(){
  pinMode(2,OUTPUT);
}
```

```

void loop(){
    digitalWrite(2,1);
    _delay(1);
    digitalWrite(2,0);
    _delay(1);
    _loop();
}

void _delay(float seconds){
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime)_loop();
}

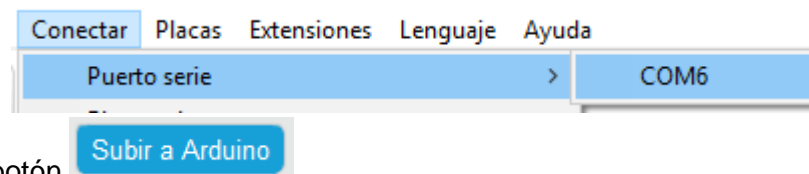
void _loop(){
}

```

Paso 3 - Subir el código a la placa Arduino

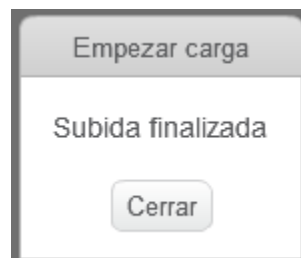
Para subir el código de nuestro programa a la placa Arduino, necesitamos:

1. Conectar la placa Arduino a la entrada USB.
2. Chequear que en el menú "Placas" esté seleccionado "Arduino Uno".
3. Seleccionar el puerto serie al que está conectada la placa.



4. Clickear el botón

Al terminar de subir nuestro código, veremos este mensaje



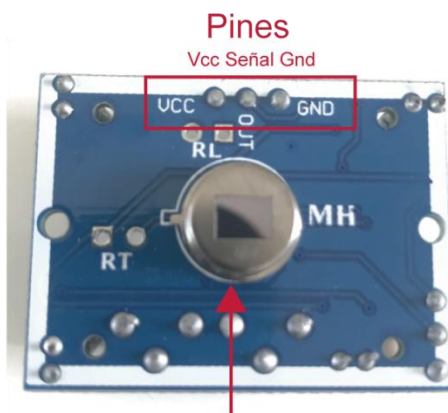
Paso 4 - Conectar el sensor piroeléctrico

Si el sistema de encendido y apagado que probamos en los pasos anteriores funciona correctamente, podemos pasar a la instalación del sensor de movimiento. Para poder controlar que las luces de los ambientes de la casa se enciendan solo cuando haya alguien adentro, utilizaremos un sensor piroeléctrico.



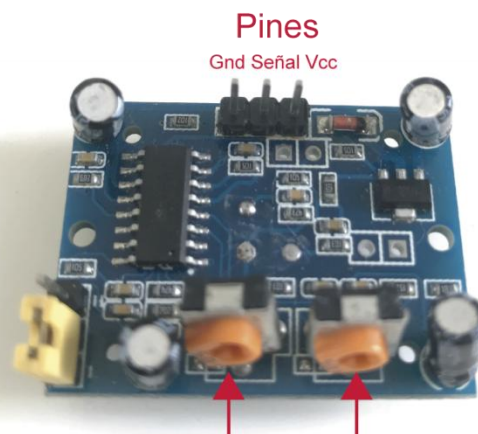
Los detectores piroeléctricos (habitualmente llamados PIR) son sensores de movimiento que reaccionan sólo ante determinadas fuentes de energía, tales como el calor que emite el cuerpo humano o de los animales. Así, pueden reconocer el movimiento de un cuerpo en un ambiente, dado que sienten la diferencia de calor entre el cuerpo y el resto del espacio.

Frente del sensor



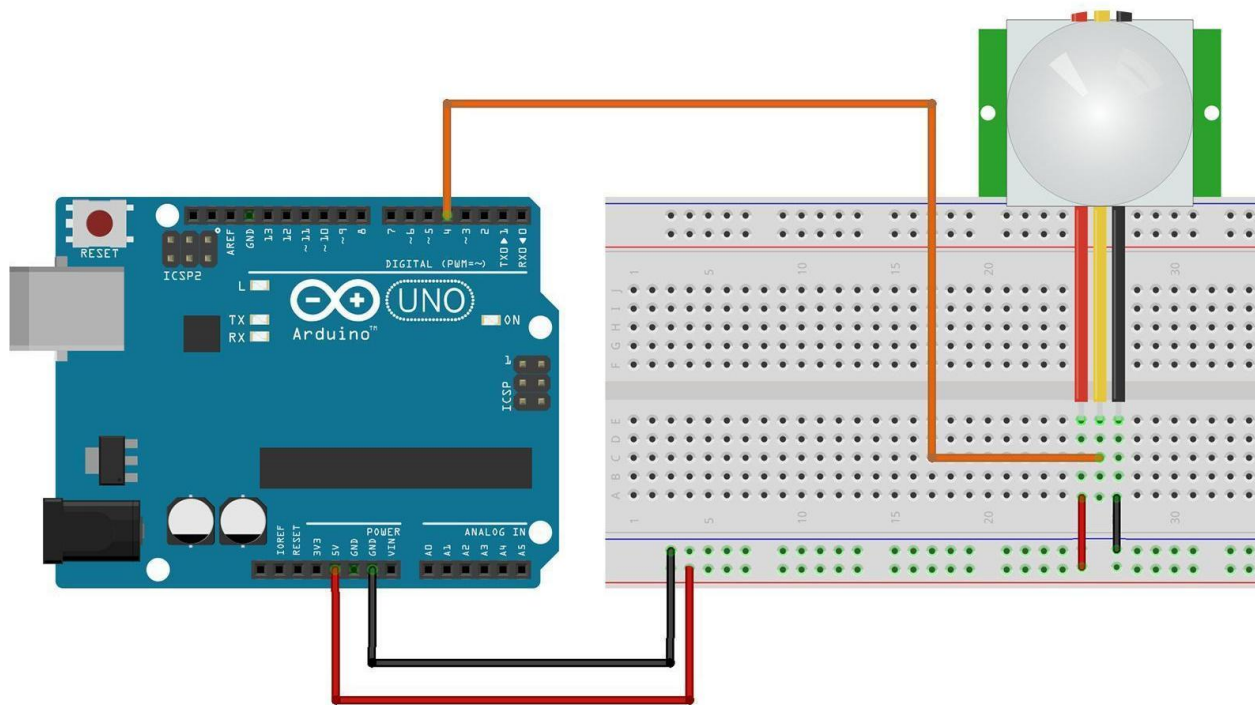
Sensor Pir

Reverso del sensor



Ajuste de Sensibilidad

Ajuste de Tiempo



Paso 5 - Calibración del sensor piroeléctrico

En primer lugar, debemos familiarizarnos con la forma de medición del sensor PIR.

El PIR indicará el valor “1” cuando detecte movimiento y “0” cuando no lo haga.

Podemos visualizar los valores de registro del sensor en la consola del programa, que se encuentra en la esquina inferior derecha de la pantalla. Es importante seleccionar “modo de caracteres” para el modo de recepción de los datos, como se muestra en la imagen:

```
24    long endTime = millis() + seconds * 1000;
avrdude: load data flash data from input file /var/folders/
avrdude: input file /var/folders/04/f5mr_mss4dz3fybhznr3rz
avrdude: reading on-chip flash data:

Reading | #####

avrdude: verifying ...
avrdude: 4502 bytes of flash verified

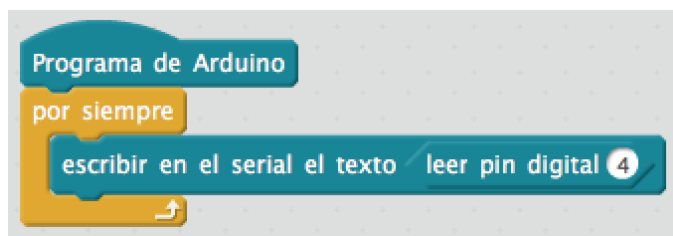
avrdude done. Thank you.
```

send encode mode: ☒ modo binario ☐ modo de caracteres

recv encode mode: ☐ modo binario ☒ modo de caracteres

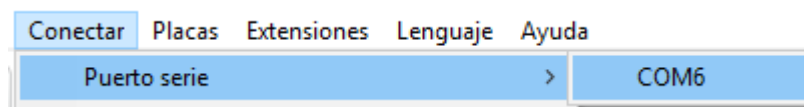
Enviar

Para enviar los datos que toma el sensor a la consola se utiliza el bloque “Escribir en el serial el texto”. A este bloque se le agrega el bloque “Leer pin digital”, asociado a la lectura del *pin* en el que esté conectado el sensor PIR (en este caso, el número 4). Es importante que el bloque “Por siempre” contenga a los dos bloques anteriores, ya que necesitamos que el sistema actualice constantemente el valor de lectura del sensor.

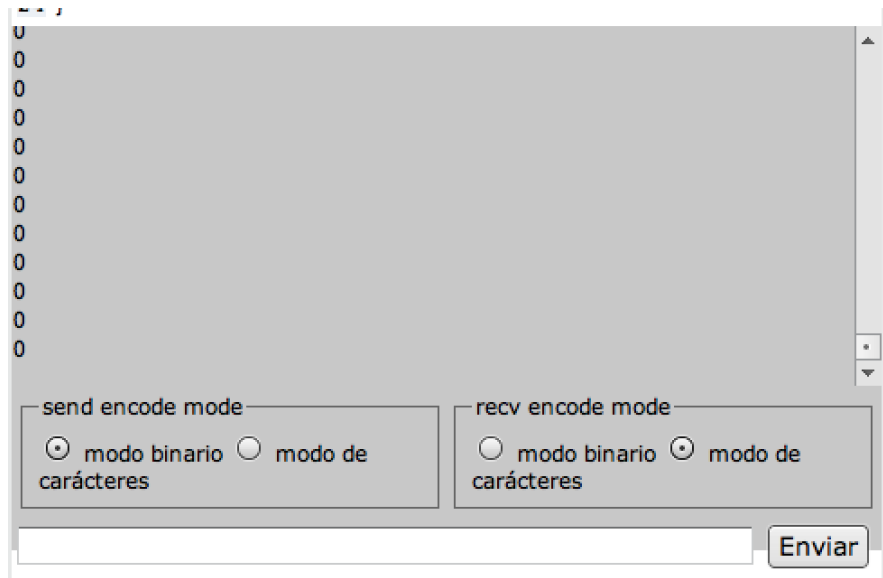


Paso 6 - Activar el envío de datos a la consola

Una vez que nuestro programa esté cargado, debemos volver a conectar nuestra placa para que se envíen los datos a la consola.

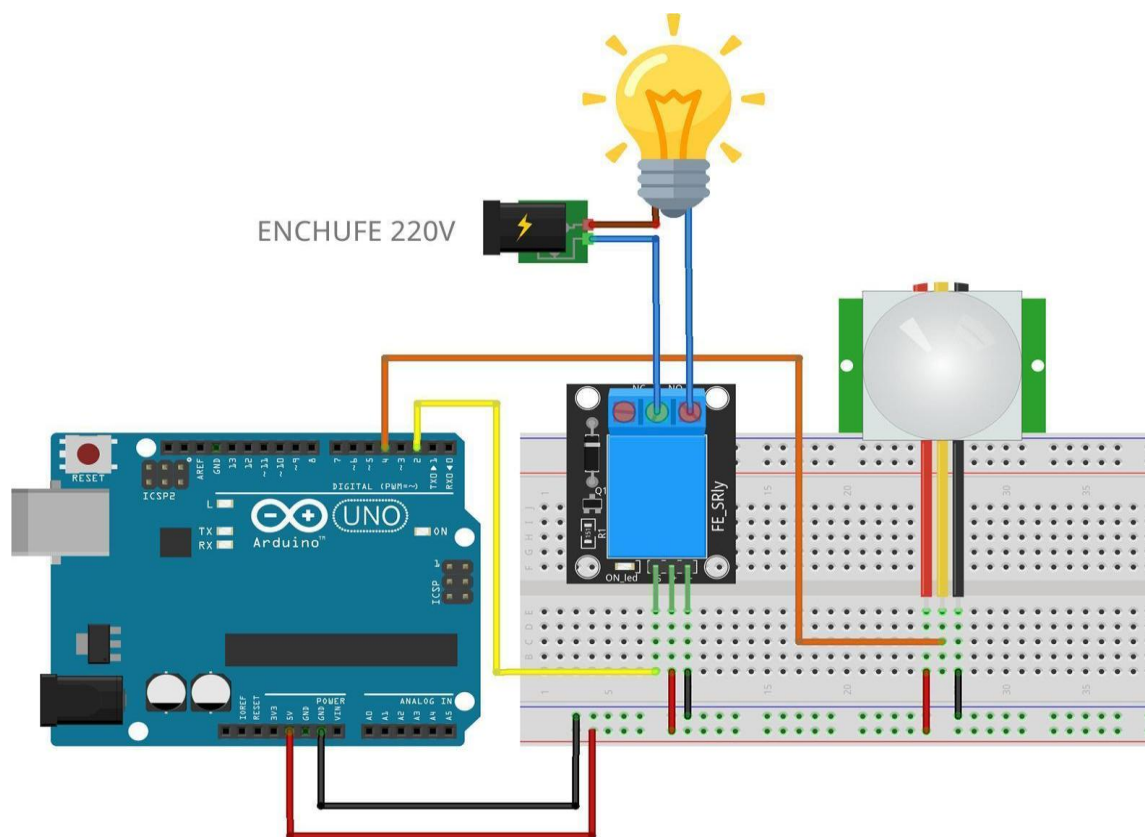


Finalmente podemos visualizar los valores del sensor en el programa. Con los trimmers de ajuste podemos calibrar la sensibilidad de medición y el tiempo que habrá entre cada lectura hasta que los parámetros se ajusten a nuestras necesidades.



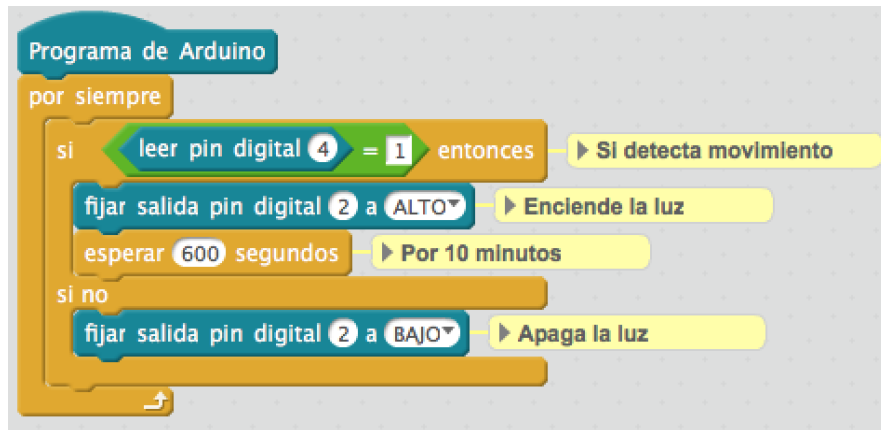
Paso 7 - Agregar el módulo relé

Con el sensor calibrado de forma correcta, es tiempo de agregar a nuestro sistema la conexión del módulo relé que hicimos previamente.



Paso 8 - Encender las luces en función del movimiento

Utilizaremos el bloque “Si... entonces/si no...” para accionar el relé cuando el sensor detecte movimiento (es decir, cuando la lectura sea igual a “1”) y para desactivarlo en caso contrario (cuando la lectura sea igual a “0”).



Para evitar que el sistema esté continuamente revisando si hay movimiento, establecemos un intervalo de tiempo de espera de, por ejemplo, 10 minutos (600 segundos) entre cada medición.

Veremos que a la derecha se muestra el código escrito que corresponde a este programa:

```
#include <Wire.h>
#include <SoftwareSerial.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;

void setup(){
  pinMode(4,INPUT);
  pinMode(2,OUTPUT);
}

void loop(){
  if(((digitalRead(4))==1)){
    digitalWrite(2,1);
    _delay(600);
  }else{
```

```

        digitalWrite(2,0);
    }
    _loop();
}

void _delay(float seconds){
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime)_loop();
}

void _loop(){
}

```

Nivel Avanzado

Observando atentamente, Agustín pudo apreciar que el nivel de luz natural presente en el exterior era un factor importante para establecer en qué momentos se necesitaba encender o apagar las luces de su casa. Entonces, se propuso complementar su sistema automático de control de luces agregando sensores ubicados afuera de la casa que midieran la intensidad de la luz solar.

Ahora, además de tener en cuenta la presencia de personas en los ambientes, el sistema encenderá las luces cuando el nivel de luz exterior sea bajo, y las apagará cuando este sea alto. De esta manera, se incrementará la optimización del uso de energía en su hogar.

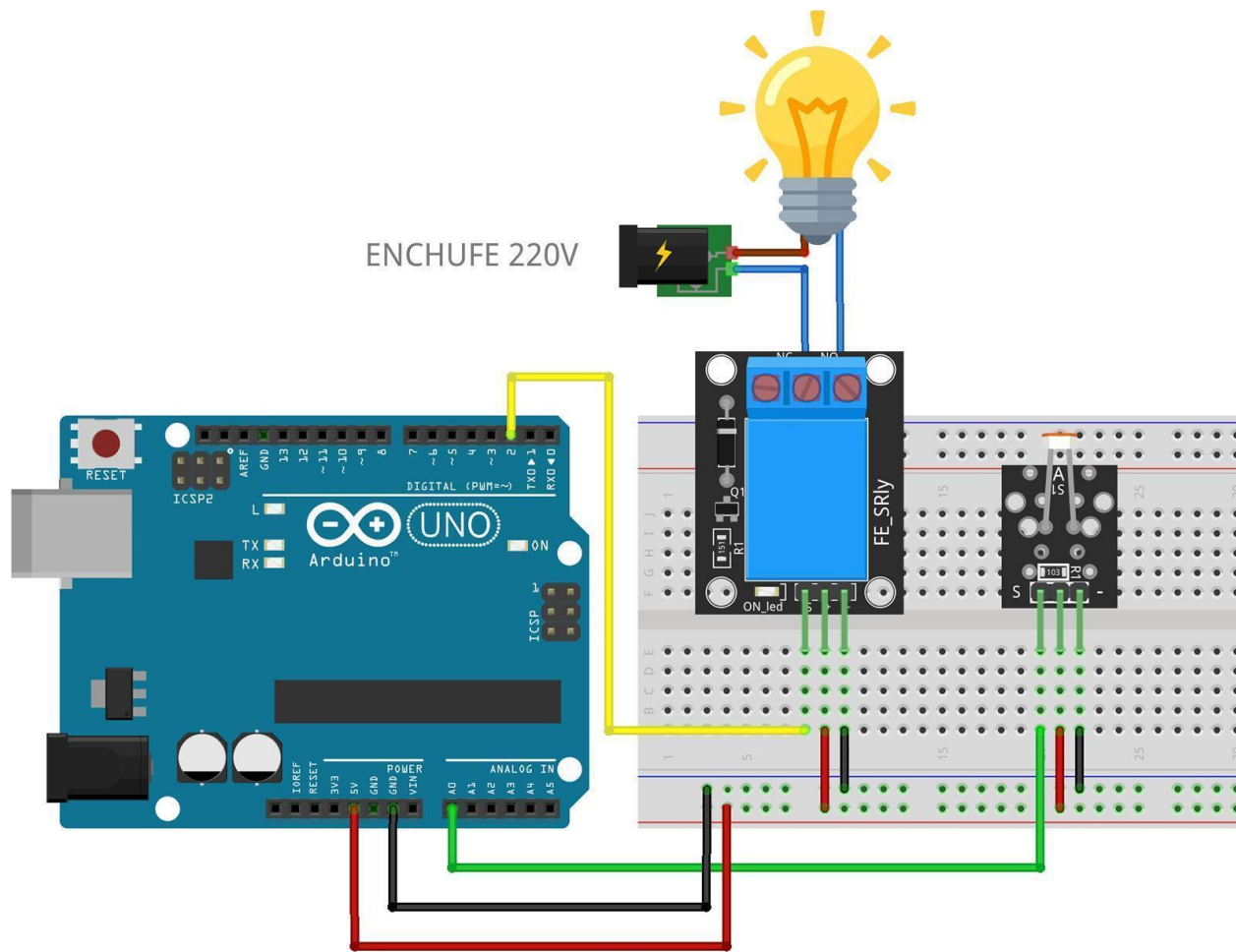
En esta actividad se propone utilizar un sensor LDR (fotorresistencia) para medir la intensidad de luz exterior, y encender o apagar las luces del hogar, a través de un módulo relé

Paso 1 - Conectar la fotorresistencia (sensor LDR)

Para poder medir la luz del exterior es necesario conectar una fotorresistencia (sensor LDR).

Un sensor LDR o una fotorresistencia es un componente electrónico cuya resistencia varía de acuerdo al cambio que perciba en la intensidad de la luz a la que está expuesto.

El valor de resistencia de un LDR es bajo cuando hay luz incidiendo en él (puede descender hasta 50 ohm) y muy alto cuando está a oscuras (1 megaohm o más).



Paso 2 - Obtener la lectura del sensor LDR

Al igual que hicimos con el sensor PIR, primero prepararemos nuestro programa para leer el *pin* y visualizar sus mediciones en la consola. Como el LDR es un sensor analógico, su lectura nos brindará valores entre 0 (cuando no haya luz) a 1023 (cuando haya mucha luz).

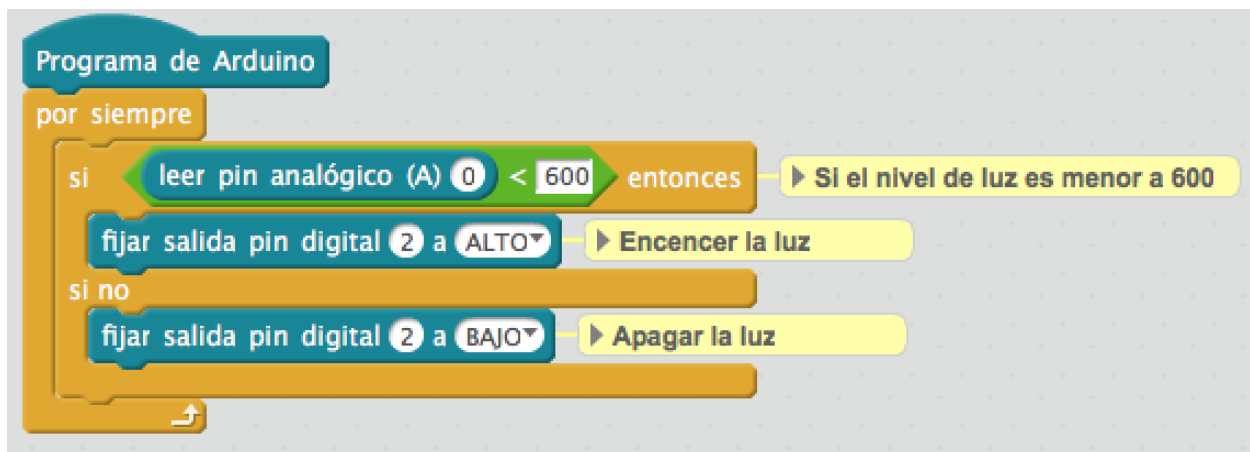
Las entradas analógicas de la placa Arduino que van desde el *pin* A0 al A5 nos permiten saber el voltaje de la entrada con una resolución que va de **0** cuando hay **0V** a **1023** cuando hay **5V**. Si, por ejemplo, el voltaje en la entrada es de **1.25V**, estos *pines* leen un valor intermedio de **255**.



Como nuestro objetivo es detectar el nivel luz natural presente en el exterior, sería recomendable poder realizar mediciones en distintos momentos del día e ir observando qué valores obtenemos en cada uno de ellos. De todas formas, para probar el funcionamiento del sensor LDR, se puede utilizar una luz artificial variando su intensidad.

Paso 3 - Encender y apagar las luces en función del sensor LDR

Tomando los valores que obtuvimos al medir los niveles de la luz del exterior, podemos establecer ante qué valor queremos que se enciendan y se apaguen las luces. Para poder programarlo, debemos utilizar la siguiente condición: si el valor de la lectura del sensor es menor al valor que determinamos (en este caso 600), la luz se encenderá; si no, se apagará. Para esto, utilizaremos el bloque “Si.. entonces/ si no...”, como se ve a continuación:



El código escrito que se puede ver a la derecha debería ser similar al siguiente:

```
#include <Wire.h>
#include <SoftwareSerial.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;

void setup(){
    pinMode(A0+0,INPUT);
    pinMode(2,OUTPUT);
}

void loop(){
    if((analogRead(A0+0)) < (600)){
        digitalWrite(2,1);
    }else{
        digitalWrite(2,0);
    }
    _loop();
}

void _delay(float seconds){
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime)_loop();
}

void _loop(){
}
```

Cierre

Una vez finalizado este proyecto, si se quiere continuar, es posible extenderlo. Estas son algunas opciones sugeridas:

- Agregar un display LCD que nos permita visualizar los ambientes de la casa donde las luces se encuentran encendidas.
- Programar un “modo de vacaciones”, utilizando un módulo reloj, para determinar distintos horarios de encendido y apagado de las luces.
- Controlar el encendido y apagado de luces a través de IoT.

El proceso de resolución de problemas, como los que se han planteado aquí, permite la movilización y la integración de distintos saberes en la búsqueda de soluciones posibles a una situación dada. Si bien la información aquí fue presentada a modo de instructivo, se espera que sean los estudiantes organizados en pequeños grupos quienes vayan encontrando las mejores formas para construir los dispositivos. Esto implica preparar los materiales para que cada grupo cuente con todo lo necesario para la construcción del proyecto. Además, al interior de cada grupo, los estudiantes deben distribuirse los roles y las tareas de acuerdo a las demandas que van teniendo en las actividades.

Es importante que los docentes acompañen las producciones de cada grupo monitoreando los avances de todos los estudiantes y presentando la información que se considere necesaria para continuar la tarea. Pero, al mismo tiempo, es necesario que habiliten espacios para que los alumnos realicen hipótesis, planteen interrogantes, indaguen, prueben y realicen ajustes de acuerdo a lo que ellos mismo van pensando sobre cómo llevar a cabo el proyecto.

En este sentido, registrar lo que se va haciendo, las preguntas de los alumnos, las pruebas, los errores, y cómo se fueron construyendo los dispositivos, permite reflexionar sobre la propia práctica, reforzar los aprendizajes construidos a lo largo de este proceso y poder volver a ese material disponible para próximos proyectos que se realicen.

Una vez terminado el proyecto, se sugiere reunir y organizar con el grupo el registro que se hizo del proceso realizado. Esta instancia de sistematización también permite movilizar capacidades vinculadas a la comunicación, porque implica tomar decisiones respecto a cómo se quiere mostrar el proyecto a otros (otros grupos, otras escuelas, otros docentes, a la comunidad, etc.).

Glosario

Electrónica y arduino

Arduino: Placa electrónica que contiene un microcontrolador programable y sistema de comunicación (USB y serial) que permite al usuario cargarle diversos programas así como también comunicarse con la misma. Del lado de la computadora se utiliza un IDE de programación para generar el código, compilarlo y quemarlo en la placa. Existen múltiples IDE compatibles con las placas Arduino.

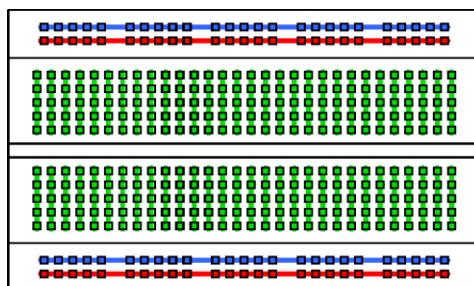
El microcontrolador posee entradas analógicas y digitales así como salidas digitales, PWM y servo. Las entradas y salidas digitales son las que permiten leer o escribir estados del tipo binarios. Pueden adoptar la forma de 0 ó 1, alto o bajo, verdadero o falso. Para prender y apagar los LED del semáforo utilizamos salidas digitales, las mismas están nombradas con números desde el 0 al 13.

Las entradas analógicas permiten leer información que puede adoptar diferentes niveles de tensión, tal como la lectura de un termómetro analógico, la posición de un potenciómetro, etc. Las mismas están identificadas en la placa como A0 a A5.

Puerto COM: Es el puerto de comunicaciones a través del cual un sistema operativo informático se comunica con un dispositivo externo tal como una placa Arduino. La asignación de los mismos suele realizarse de forma automática al conectar la placa via USB. Dicha asignación suele ser dinámica, lo que significa que a veces cambia el número al conectar una misma placa en otro puerto USB o al conectar varias placas. En todos los IDE de programación es necesario especificar el puerto COM a través del cual nos comunicaremos con la placa Arduino.

Protoboard: Es una placa experimental que permite el prototipado rápido de circuitos electrónicos. Tiene orificios para insertar las patas de los componentes permitiendo que se conecten sin tener que recurrir a la soldadura.

El mismo posee una grilla de orificios que se encuentran conectados entre sí siguiendo el esquema de la imagen. Las líneas de conexión superior e inferior recorren la placa de punta a punta y suelen utilizarse para la alimentación del circuito, mientras que las líneas verdes se suelen utilizar para interconectar componentes. Tomar en cuenta que las líneas verdes se interrumpen en el centro de la placa. Generalmente se utilizan cables del tipo dupont para realizar conexiones en la protoboard



Sensor de movimiento infrarrojo (habitualmente llamados PIR) son sensores que reaccionan sólo ante determinadas fuentes de energía, tales como el calor que emite el cuerpo humano o de los animales. Pueden reconocer movimiento de una persona en un rango máximo de un par de metros. Solo detectan presencia o no del movimiento, sin realizar ningún tipo de dimensionamiento del mismo. Suelen tener un trimmer para regular la sensibilidad del mismo y tiempo que dura el disparo cada vez que se detecta algo.

Se suelen utilizar para sistemas de alarma domiciliaria o para reflectores automáticos activados por movimiento. Generalmente tiene dos pines para la alimentación (VCC y GND) y un tercer pin para la salida de información que se conecta a una entrada digital del Arduino.

Fotoresistencia (LDR): es componente electrónico cuya propiedad es ofrecer resistencia al flujo de corriente. El LDR se distingue de los resistores tradicionales por estar elaborado de un material que es sensible a la luz. Al variar la intensidad de luz a la que se expone varía proporcionalmente la resistencia eléctrica del componente. El mismo tiene dos patas de conexión. Algunos casos de aplicación pueden ser un robot seguidor de luz, el control automático de iluminación, control automático de brillo en una pantalla o barreras láser.

Relé: Dispositivo electromagnético que funciona como un interruptor controlado por un circuito eléctrico. Por medio de un electroimán se acciona uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes. Se utiliza comúnmente para aislar eléctricamente dos circuitos así como también permitir controlar con bajo voltaje a dispositivos que utilizan mayor voltaje. Arduino trabaja con señales de 5 V que mediante un relé permiten controlar la activación o desactivación de dispositivos de 220 V.

Reconocimientos

Este trabajo es fruto del esfuerzo creativo de un enorme equipo de entusiastas y visionarios de la pedagogía de la innovación, la formación docente, la robótica, la programación, el diseño y la impresión 3D. Les agradecemos por el trabajo en equipo inspirador para traer a la realidad la obra que, en forma conjunta, realizamos INET y EDUCAR del Ministerio de Educación, Cultura, Ciencia y Tecnología de la Nación Argentina.

Contenidos

Equipo INET

Alejandro Anchava
Joreliz Andreyana Aguilera Barragán
Omar Leandro Bobrow
Alejandro Cesar Cáceres
Ezequiel Luberto
Gustavo Roberto Mesiti
Alejandro Palestrini
Judit Schneider
Pablo Trangone

Equipo Educar:

Pablo Aristide
Mayra Botta
Anabela Cathcarth
Eduardo Chiarella
María Laura Costilla
Diego Dorado
Facundo Dyszel
Federico Frydman
Matías Rinaldi
Uriel Rubilar
Camila Stecher
Carolina Sokolowicz
Nicolás Uccello

Para la confección de esta obra se contó con el apoyo de la Universidad Pedagógica Nacional "UNIPE". En particular en el desarrollo de los capítulos 1 y 2, los cuales estuvieron a cargo de los profesores Fernando Raúl Alfredo Bordinon y Alejandro Adrián Iglesias.

Producción y comunicación

Juliana Zugasti

Diseño y edición

Leonardo Frino
Mario Marrazzo

Corrección de estilo

María Cecilia Alegre

Agradecimientos especiales

Mariano Consalvo. Equipo ABP

Damián Olive. Equipo de ABP

María José Licio Rinaldi, Directora Nacional de Asuntos Federales INET, quien siempre acompañó a este equipo en todas las gestiones para su implementación

Estamos comprometidos en instalar la innovación en la escuela secundaria técnica: la robótica, la programación, el pensamiento computacional, los proyectos tecnológicos, el ABP, la impresión 3D, de manera más accesible para todos.

Agradecemos enormemente, docente, tu continua dedicación y compromiso con el futuro de tus estudiantes.

¡Estamos ansiosos por saber qué es lo que vamos a crear juntos!