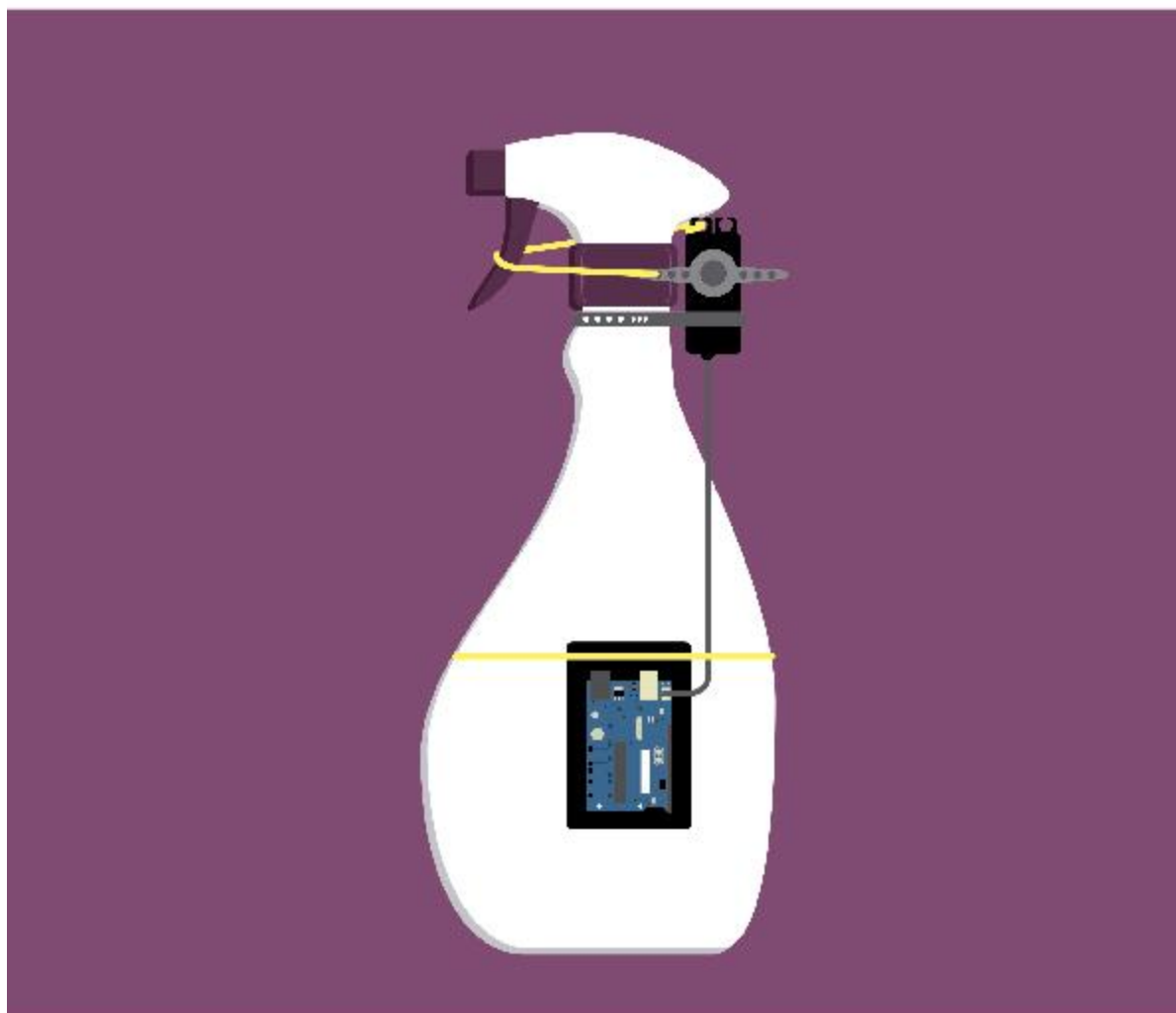


# SABERES DIGITALES

---



**PULVERIZADOR AUTOMATIZADO**

---

# 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.
PULVERIZADOR AUTOMATIZADO	4
Ficha técnica	4
Presentación	5
Desarrollo	5
<b>Nivel Inicial</b>	5
<b>Paso 1 - Conectar el servomotor.</b>	6
<b>Paso 2 - Programar el servomotor</b>	7
<b>Paso 3 - Subir el código a la placa Arduino.</b>	9
<b>Paso 4 - Movimiento del servomotor.</b>	10
<b>Paso 5 - Impresión de las piezas del mecanismo</b>	11
<b>Paso 6 - Ensamblaje del sistema mecánico con la parte electrónica y el contenedor estándar</b>	13
<b>Nivel Avanzado</b>	15
<b>Paso 1 - Conexión del sensor PIR.</b>	15
<b>Paso 2 - Calibración del sensor PIR</b>	17
<b>Paso 3 - Activar el envío de datos a la consola</b>	18
<b>Paso 4 - Accionar el pulverizador</b>	18
<b>Paso 5 - Ciclos repetitivos</b>	19
<b>Paso 6 - Esperar antes de accionar el pulverizador</b>	20
Cierre	22
Glosario	23
Reconocimientos	¡Error! Marcador no definido.

# PULVERIZADOR AUTOMATIZADO

## Ficha técnica

<b>Nivel educativo</b>	Secundario. Ciclo Básico.
<b>Descripción general</b>	Diseño y construcción de un sistema de pulverización de agua.
<b>Niveles de complejidad</b>	Nivel inicial: Construir un mecanismo automatizado de pulverización de agua con una <b>placa Arduino</b> y un <b>temporizador</b> . Nivel avanzado: Programar el pulverizador para que entre en acción en función de los datos obtenidos con un sensor PIR de movimiento.
<b>Insumos</b>	<ul style="list-style-type: none"><li>• 1 x Gatillo Pulverizador estandar</li><li>• 1 x Servo SG5010</li><li>• 1 x rollo PLA</li><li>• 20 cables dupont macho hembra</li><li>• 20 cables dupont macho macho</li><li>• 1 x Arduino UNO R3</li><li>• 1 x Protoboard</li><li>• 1 x Cable usb tipo B</li><li>• 1 x Fuente de 9v 1 A (plug centro positivo, 5.5x2.1mm)</li><li>• 1 x Sensor PIR (piroeléctrico)</li></ul>
<b>Equipamiento</b>	<ul style="list-style-type: none"><li>• Computadora</li><li>• Soldador</li><li>• Estaño</li><li>• Alicata</li><li>• Pinza de punta</li><li>• Brusela</li></ul>
<b>Otros requisitos</b>	<ul style="list-style-type: none"><li>• Conexión a internet</li><li>• Descargar el programa "mblock3" <a href="http://www.mblock.cc/software-1/mblock/mblock3/">http://www.mblock.cc/software-1/mblock/mblock3/</a></li></ul>

# Presentación

## Descripción ampliada del proyecto

El proyecto propone la construcción y la programación de un prototipo de un pulverizador de agua. En el nivel inicial, se propone la utilización de un servomotor que esté programado para accionar un gatillo cada cierto tiempo, que será determinado de antemano, y así realizar la pulverización.

En el nivel de complejidad avanzado, se propone agregar al sistema un sensor PIR de movimiento y programar el pulverizador para que se accione en función de la información que este recoja.

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 el manejo de sistemas mecánicos mediante la construcción de un prototipo de un pulverizador automático de agua
- Conocer los componentes de la interfaz de Arduino.
- Introducirse en la tecnología de impresión 3D.
- Analizar y desarrollar la programación secuencial de un programa que permita el funcionamiento de un prototipo del pulverizador automático, programando el tiempo en que se accionará (nivel inicial) e incorporando un sensor de movimiento (nivel avanzado).

# Desarrollo

## **Nivel Inicial**

El dueño de un supermercado desea incorporar un sistema de pulverización de agua a sus góndolas de frutas y verduras. Necesita que el pulverizador se accione de forma automática, respetando intervalos de tiempo previamente definidos, así él puede realizar otras tareas y mantener simultáneamente el cuidado de estos productos.

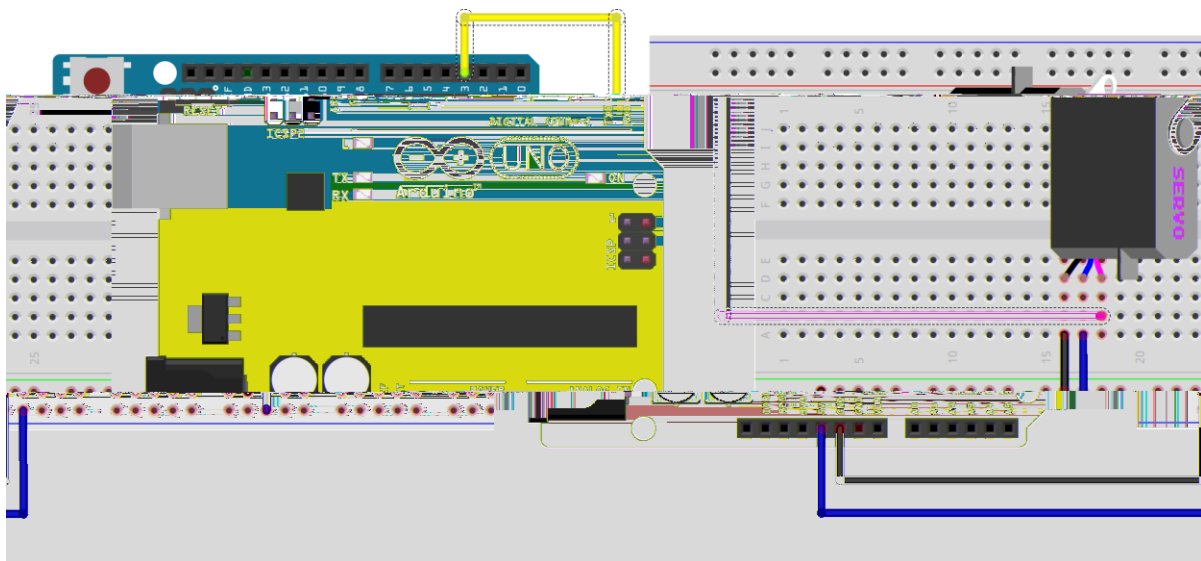
**En esta instancia del proyecto se propone construir un pulverizador con un sistema automático, utilizando una placa Arduino y un servomotor, y programar sus tiempos de acción.**

La pulverización de líquidos consiste en hacerlos pasar a través de un dispositivo que los expulsa en forma de pequeñas gotas. Esto puede ser aprovechado con distintos fines, como modificar las condiciones de humedad y temperatura de un ambiente (por ejemplo, un invernadero) a través de la pulverización de agua, realizar una aplicación selectiva de aceites lubricantes en maquinaria industrial o aplicar perfumes de ambiente. En la mayoría de los casos, esta técnica representa una utilización eficiente de los líquidos pulverizados, evitando posibles excesos.

Dependiendo de las necesidades del usuario, en algunos casos puede ser útil contar con un sistema de pulverización de accionar automático. Esto permite alcanzar los mismos objetivos realizando un menor esfuerzo y ocupando un menor tiempo de trabajo. Es necesario, en estos casos, establecer la frecuencia de accionar del pulverizador y la cantidad total de pulverizaciones que se precisan para cumplir con los objetivos planteados.

### Paso 1 - Conectar el servomotor.

Comenzamos realizando la conexión del servomotor, que será el encargado de activar el gatillo del pulverizador. Primero, se conecta el cable de señal (amarillo) al *pin* 3 de la placa Arduino. Luego, la alimentación eléctrica (cable negro y rojo) se conecta a los pines GND y 5v, como se muestra en la imagen.



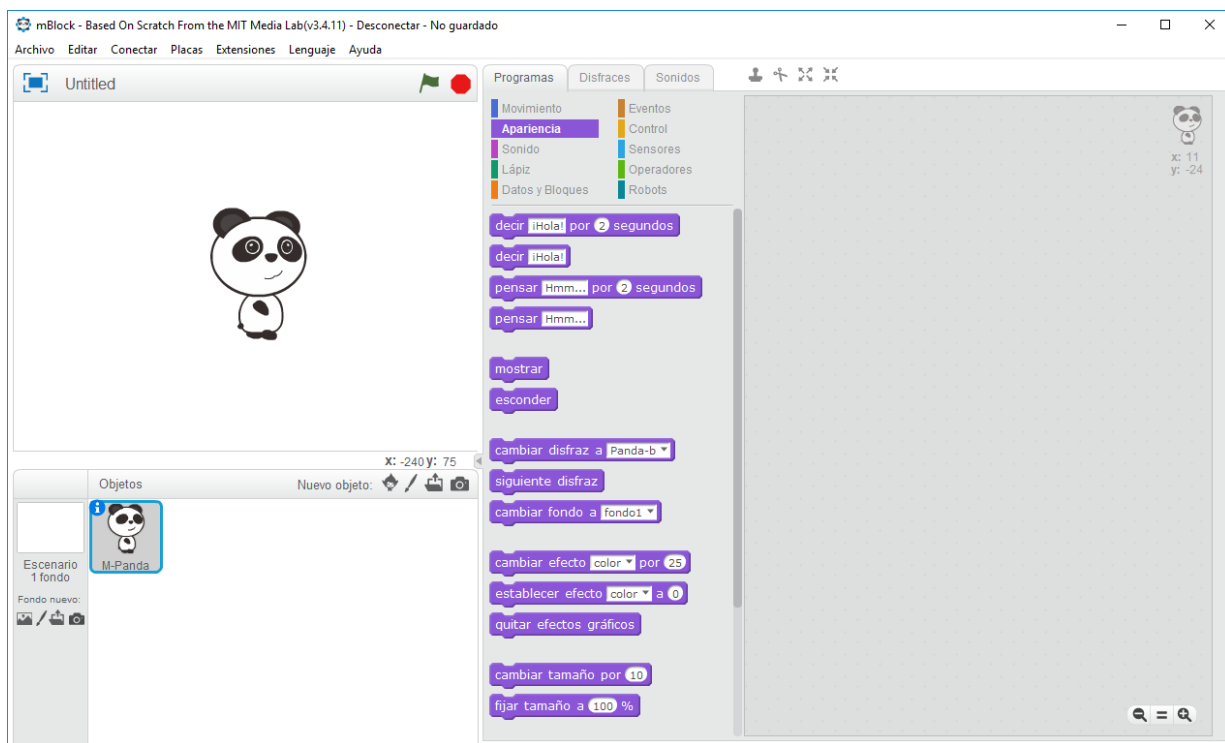
El servomotor (habitualmente llamado “*servo*”) es un motor de corriente continua con un sistema de control que le permite ubicarse en una posición específica y mantenerse estable.

Normalmente pueden moverse 180° o 360° dependiendo del modelo. Poseen tres cables: negativo (negro o marrón), positivo (rojo) y señal (naranja o amarillo) .

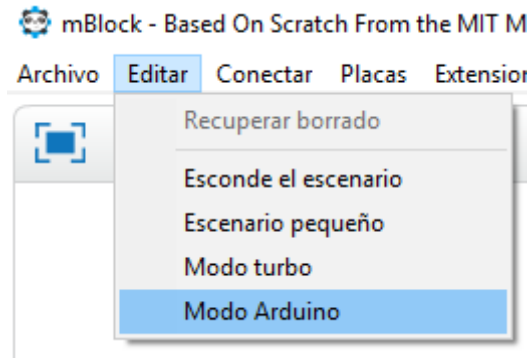
## Paso 2 - Programar el servomotor

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

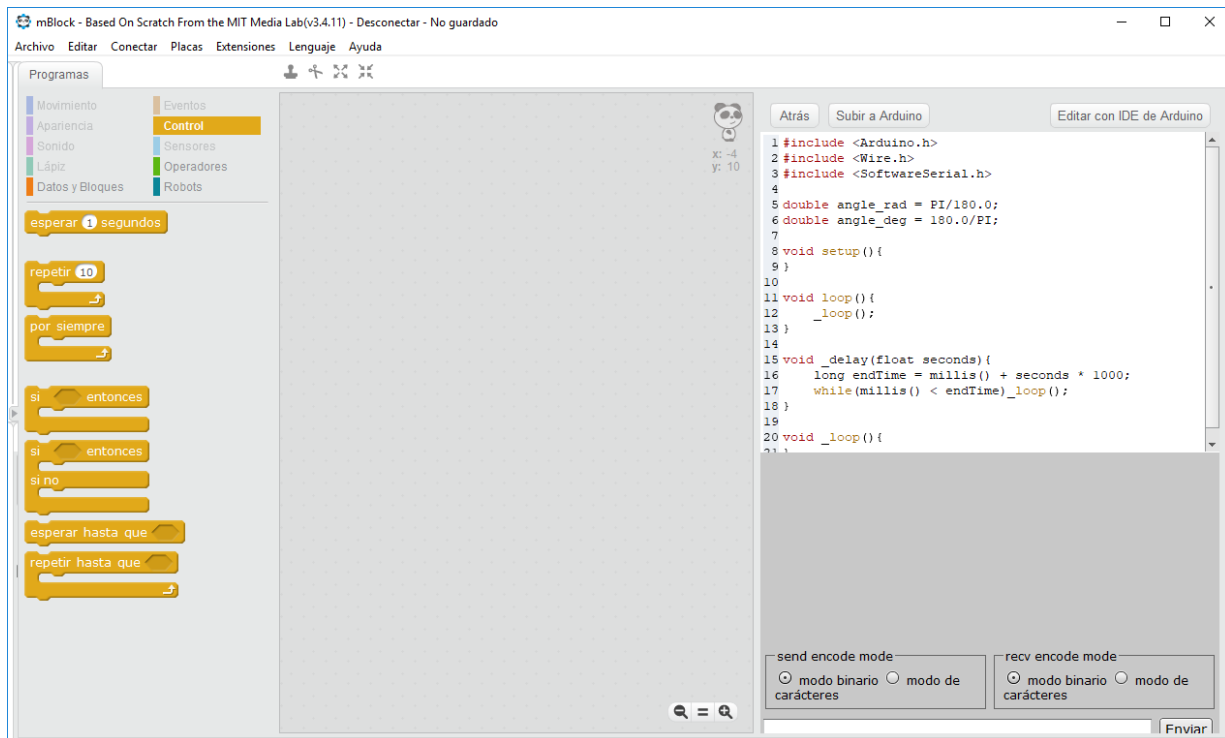
Cuando abrimos mBlock3, veremos una pantalla como la siguiente:



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 bloques de las categorías “**Robots**”, “**Control**”, “**Operadores**” y “**Datos y Bloques**”. Cuando seleccionamos una de estas categorías, se pueden visualizar todos los bloques que pertenecen a ese grupo.





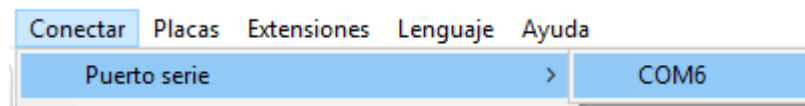
Para comenzar el programa necesitamos agregar el bloque “Programa de Arduino” que se encuentra dentro de la categoría **Robots**. Luego, colocamos el bloque “fijar ángulo del servo” y completamos sus campos. En el campo que está después de la palabra “pin” se debe colocar el mismo número del *pin* al que está conectado el servomotor en la placa Arduino. En el campo final, designado al ángulo, elegimos 90°.



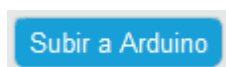
### Paso 3 - Subir el código a la placa Arduino.

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

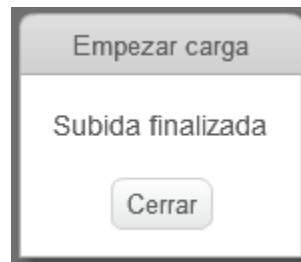
1. Conectar la placa Arduino a la entrada USB.
2. Chequear que en el menu “Placas” esté seleccionado “Arduino Uno”.
3. Seleccionar el puerto serie al que está conectada la placa.



4. Clickear el botón



Al terminar de subir nuestro código, veremos este mensaje



Con la conexión realizada previamente, el servo deberá moverse a la posición de 90°.

#### Paso 4 - Movimiento del servomotor.

Para que se accione el gatillo, necesitamos que el servomotor gire de 90° a 0°, respetando intervalos de tiempo previamente determinados. Para asegurarnos de que funciona correctamente, haremos una prueba y lo programaremos para que se mueva cada 5 segundos. Más adelante, podremos modificar este intervalo de acuerdo a nuestras necesidades. En esta instancia, todos los bloques “fijar ángulo del servo” deberán estar contenidos en el bloque “por siempre”, dado que necesitamos que esta acción se repita constantemente.



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

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

#include <Servo.h>
```

```

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
Servo servo_3;

void setup(){
    servo_3.attach(3); // init pin
}

void loop(){
    servo_3.write(90); // write to servo
    _delay(1);
    servo_3.write(0); // write to servo
    _delay(5);
    _loop();
}

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

void _loop(){
}

```

## Paso 5 - Impresión de las piezas del mecanismo

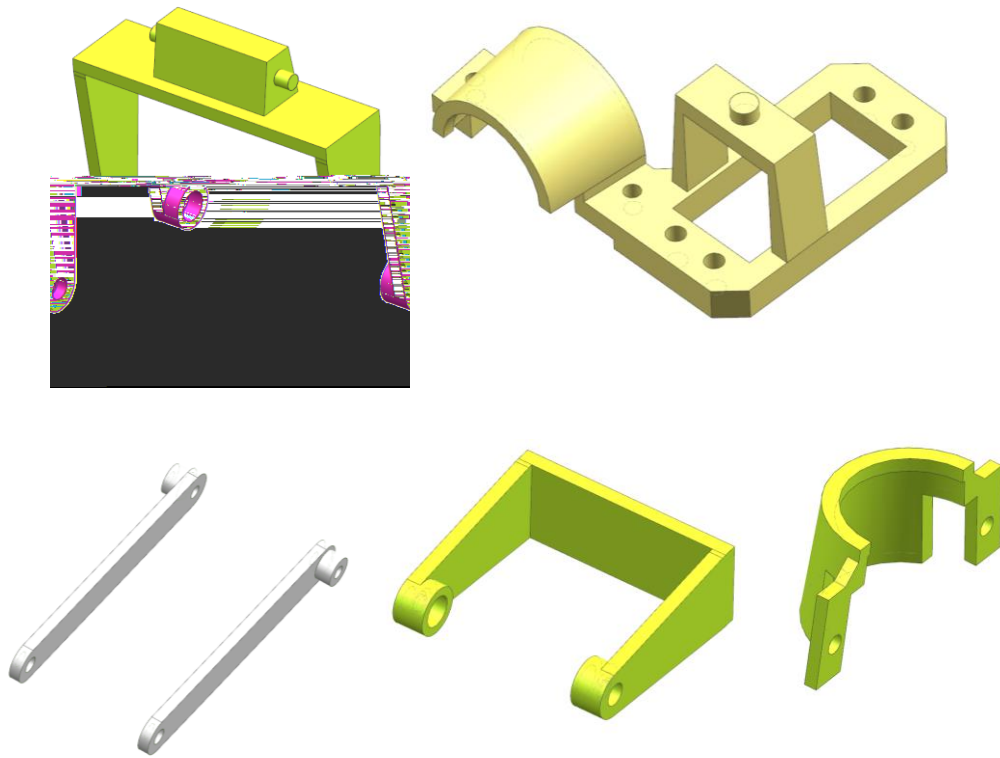
Para armar el mecanismo, se deben imprimir las piezas en la impresora 3D. El modelo 3D del dispositivo se puede descargar de forma libre y gratuita en el siguiente enlace: <https://enfoco.net.ar/sd>.

Una vez descargado el modelo, lo imprimimos con la impresora 3D. Cuando estén listas todas las piezas, pasamos al ensamblado.

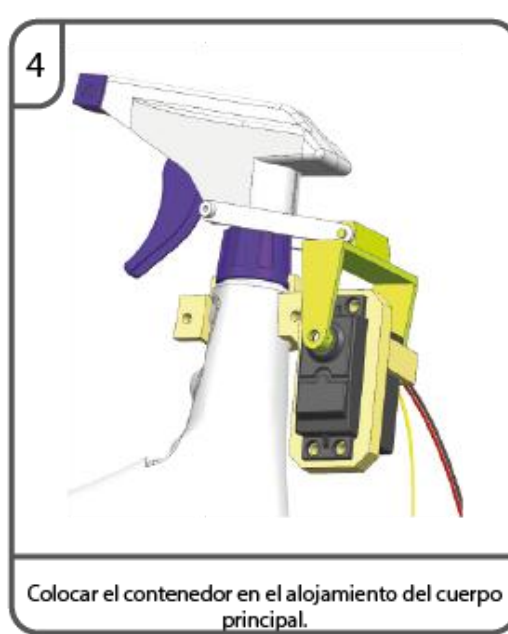
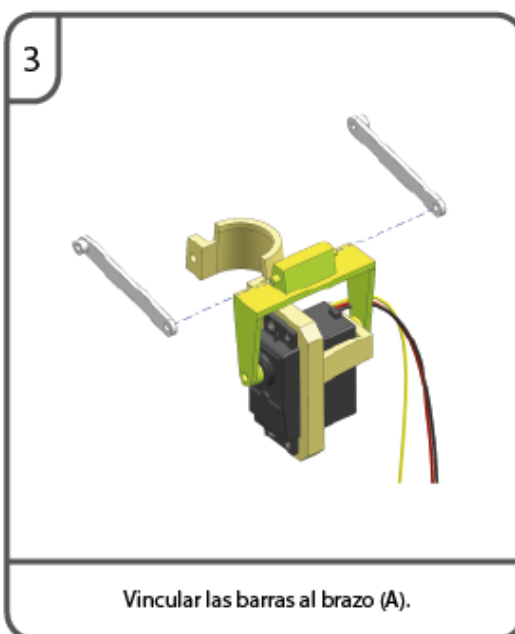
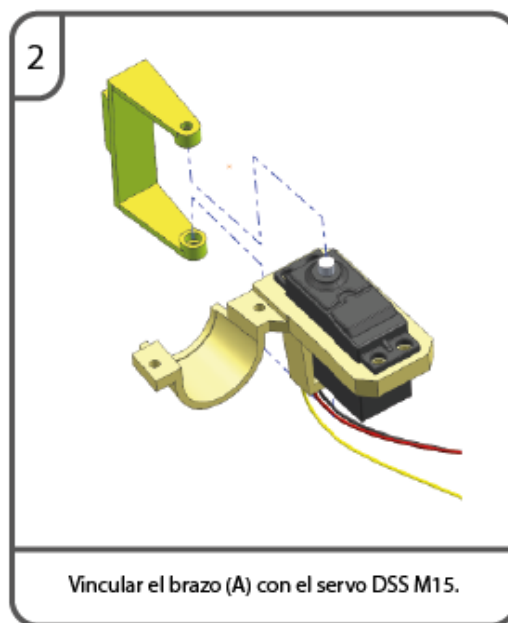
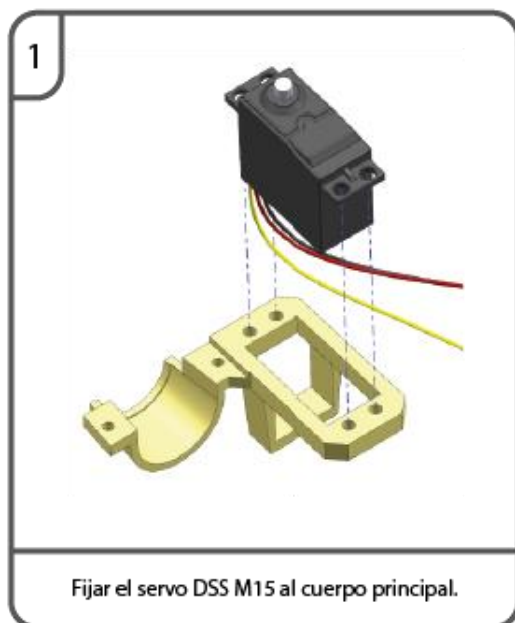
En caso de querer realizar una modificación en el modelo, independientemente del programa de modelado que utilicemos, debemos exportar nuestras piezas en formato .stl.

El .stl es un formato de archivo informático de diseño asistido por computadora (CAD) que define la geometría de objetos 3D, excluyendo cierta información, como color, texturas o propiedades físicas, que se incluye en otros formatos CAD. Este es el formato que necesitamos abrir desde el software de impresión para poder desde allí exportar el modelo en código G.

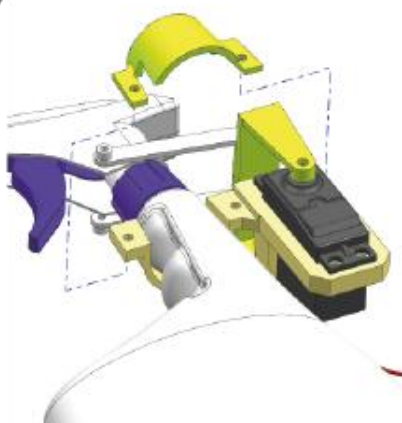
Código G es el nombre que habitualmente recibe el lenguaje de programación más usado en control numérico (CN). Es por medio de él que la impresora 3D logrará convertir el modelado en trayectorias para poder conformar la pieza que precisamos.



## Paso 6 - Ensamblaje del sistema mecánico con la parte electrónica y el contenedor estándar

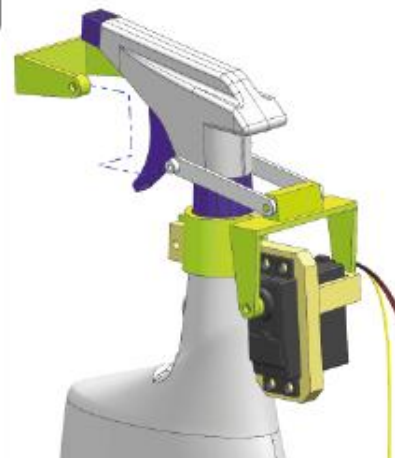


5

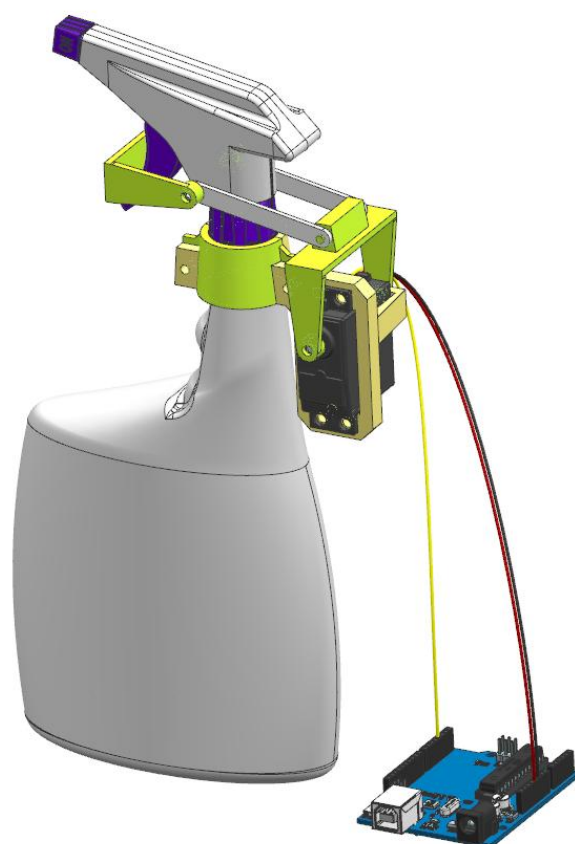


Montar la brida con el cuerpo principal.

6



Vincular el brazo (B) con las barras.



## Nivel Avanzado

Para volver el sistema más eficiente y no desperdiciar agua, el dueño del supermercado solicitó que el pulverizador se conecte a un sensor que detecte cuándo un cliente está sacando frutas o verduras de las góndolas. Así, a medida que los clientes retiren frutas o verduras, el pulverizador se activará para rociar las frutas que van quedando en la parte superior. Es importante también que el sistema de pulverización espere un tiempo para activarse, para no rociar a los clientes con el líquido.

**En esta instancia avanzada del proyecto, se propone sumar un sensor PIR (piroeléctrico) para detectar el movimiento de las manos de los clientes. Además se programará un temporizador para determinar el tiempo de activación de nuestro servomotor.**

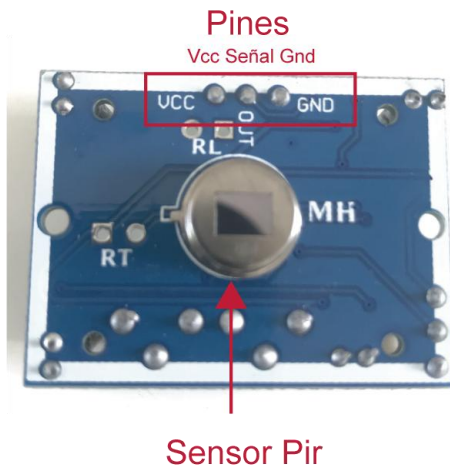
### Paso 1 - Conexión del sensor PIR.

Para poder detectar el movimiento utilizaremos un sensor "PIR".

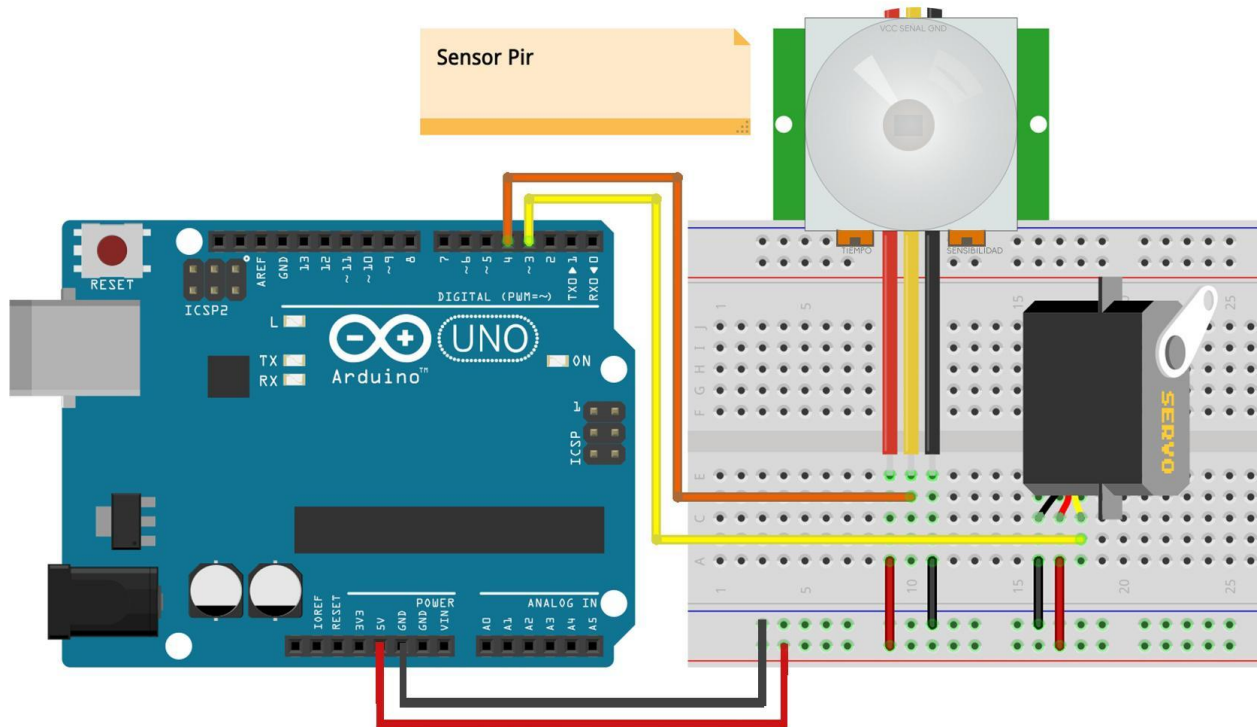
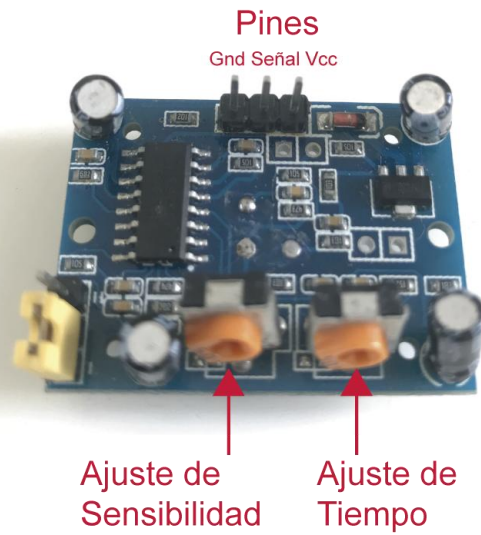


Los sensores de movimiento infrarrojo (habitualmente llamados PIR) son sensores que reaccionan sólo ante determinadas fuentes de energía, tales como el calor que emite el cuerpo humano o de los animales. Pueden reconocer movimiento de una persona en un rango máximo de un par de metros. Solo detectan presencia o no del movimiento, sin realizar ningún tipo de dimensionamiento del mismo.

Frente del sensor



Reverso del sensor



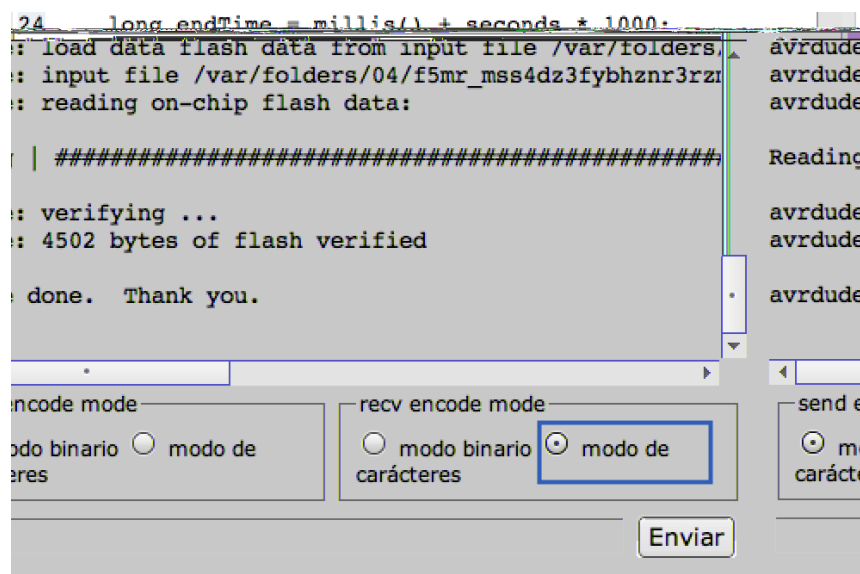


## Paso 2 - Calibración del sensor PIR

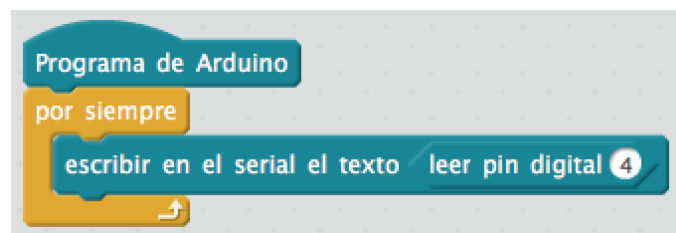
En primer lugar, debemos familiarizarnos con la forma de medición del sensor PIR.

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:

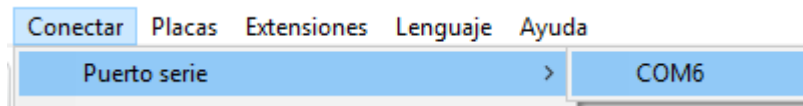


Para enviar los datos que toma el sensor a la consola se utiliza el bloque “Escribir en el serial el texto”. A este bloque se le agrega el bloque “Leer pin digital”, asociado a la lectura del pin en el que esté conectado el sensor PIR (en este caso, el número 4). Es importante que el bloque “Por siempre”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

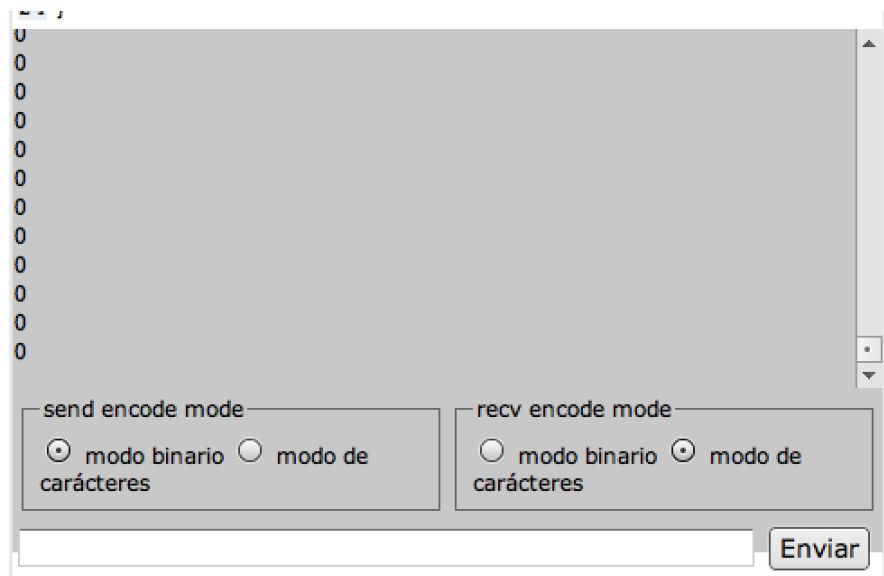


### Paso 3 - Activar el envío de datos a la consola

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 potenciómetros de ajuste podemos calibrar la sensibilidad de medición y el tiempo que habrá entre cada lectura hasta los parámetros se ajusten a nuestras necesidades.



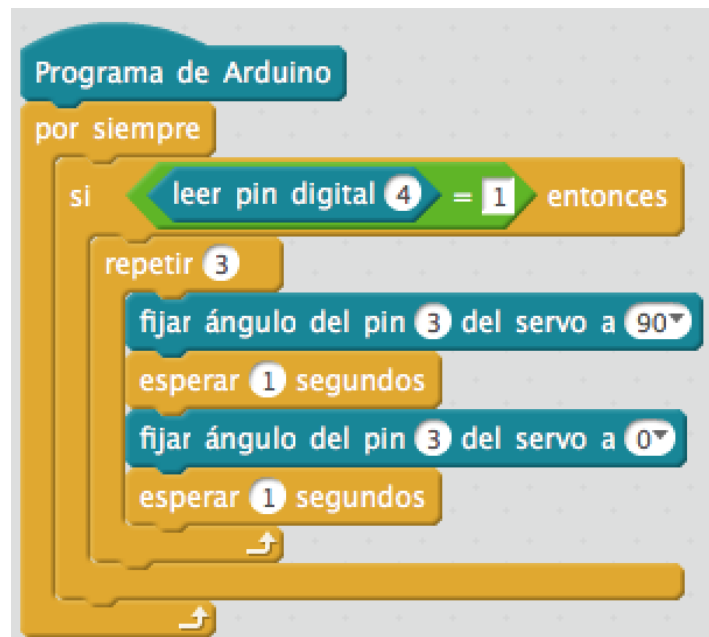
### Paso 4 - Accionar el pulverizador

Utilizaremos el bloque “Si.../ entonces...” para accionar el servomotor cuando el sensor detecte movimiento, es decir, cuando la lectura sea igual al valor 1.



## Paso 5 - Ciclos repetitivos

Para que el pulverizador se accione más de una vez necesitamos programar un ciclo de repeticiones. Para esto utilizaremos el bloque “Repetir” y determinaremos la cantidad de repeticiones en 3 veces. El tiempo entre cada accionar lo reduciremos a un segundo para que no haya tanta espera entre una pulverización y la otra.



## Paso 6 - Esperar antes de accionar el pulverizador

Hay que tener en cuenta que el pulverizador se activa automáticamente al detectar movimiento y es probable que las personas todavía se encuentren cerca de las frutas y las verduras de las góndolas. Por esta razón es necesario que dejemos pasar un intervalo de tiempo antes de que se accione el pulverizador.



Y el código escrito que se puede ver a la derecha, debería ser similar al siguiente:

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

#include <Servo.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
Servo servo_3;

void setup(){
  pinMode(4,INPUT);
  servo_3.attach(3); // init pin
```

```
}

void loop(){
    if(((digitalRead(4))==(1))){
        _delay(60);
        for(int __i__=0;__i__<3;++__i__)
        {
            servo_3.write(90); // write to servo
            _delay(1);
            servo_3.write(0); // write to servo
            _delay(1);
        }
    }
    _loop();
}

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

void _loop(){
}
```

## Cierre

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

- Programar un sensor de humedad del ambiente para determinar, de forma automática, el accionar del dispositivo según las necesidades ambientales.
- Agregar un sensor del nivel de líquido para determinar la cantidad de contenido que le resta al pulverizador e incorporar un zumbador que avise cuando haya que rellenarlo.
- Calcular, teniendo en cuenta la capacidad total del reservorio de agua del pulverizador y el volumen de agua expulsado con las pulverizaciones, cuánto líquido queda disponible; o, dicho de otra forma, cuántas pulverizaciones restan para agotar el reservorio. Luego, se puede armar un programa que recupere esa información y la publique 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.

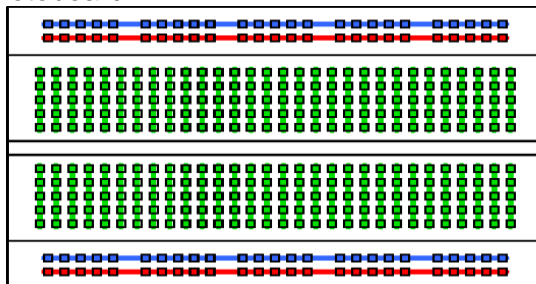
El microcontrolador posee entradas analógicas y digitales así como salidas digitales, PWM y servo. Las entradas y salidas digitales son las que permiten leer o escribir estados del tipo binarios. Pueden adoptar la forma de 0 ó 1, alto o bajo, verdadero o falso. Para prender y apagar los LED del semáforo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



**Servomotor:** Un servo es un dispositivo que se compone de un motor y un sistema de control que le permite ubicarse en una posición específica. Los servos más comunes pueden moverse

en un rango de 0° a 180°, sin poder girar de forma continua. Se suelen utilizar en aplicaciones tipo barreras o brazos mecánicos. La programación de los mismos es muy simple, teniendo que especificar únicamente el ángulo al que se lo quiere posicionar.

Existen también servos de “rotación continua” que permiten realizar un control relativamente preciso del movimiento así como también que el eje de giros continuos sin estar acotado a un rango de movimiento como el caso de los servos estándares. Este tipo de servo requiere una lógica de programación un poco más compleja que el caso anterior.

Los servos tienen 3 pines de conexión, dos de ellos se ocupan en alimentación eléctrica (VCC marrón y GND negro) y un tercer pin que se conecta a una salida digital del Arduino (cable naranja). Para controlar el servo el arduino genera una señal con una frecuencia particular y un método de modulación de pulsos cuyo ciclo de trabajo equivale a el ángulo que se desea posicionar el servo, no confundir este método de modulación con el PWM.

**Sensor de movimiento infrarrojo** (habitualmente llamados PIR): 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 potenciómetro 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.

## Impresión 3D

**Formato .stl:** El .stl es un formato de archivo que contiene la forma de un objeto sólido. Este formato de archivo no contiene información tal como color, texturas o propiedades físicas. Los archivos STL son objetos ya consolidados por lo que resulta útil para el intercambio e impresión de los mismos. Este formato de archivo no resulta práctico en caso de necesitar editar la geometría del objeto. Esto se debe a que no contiene información paramétrica sobre la generación de las diversas formas, curvas o capas que se utilizan a la hora de diseñar. Se lo puede considerar como la bajada o exportación final de un diseño, en ciertos aspectos equivalente a pensar en exportar un documento PDF partir de un documento de texto. Es posible generar archivos STL partiendo desde distintos tipos de entornos de diseño y modelado en 3D.

**Código G:** Es el nombre que recibe el conjunto de acciones que va a tener que realizar la impresora 3D, o cualquier otro tipo de máquina CNC, para completar el trabajo solicitado. Estas instrucciones se generan partiendo del análisis de un archivo STL y realizando el cálculo de todos los movimientos y trayectorias que realizará cada componente de la impresora (motores, avance de filamento, calentador de extrusor, calentador de la base, etc) para materializar el objeto en cuestión. Debido a que cada marca y modelo de impresora 3D tiene diferentes características mecánicas, el código G generado para imprimir cierto objeto solo va a servir para ejecutarse en un modelo de impresora específico.



## 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 "UNIPED". 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!***