

Taller Introducción a la Ingeniería Eléctrica Robótica y comunicaciones basado en Microcontrolador Arduino

Instituto de Ingeniería Eléctrica

Taller 0: Marco del curso e introducción al HW y SW.

12 de Marzo de 2025

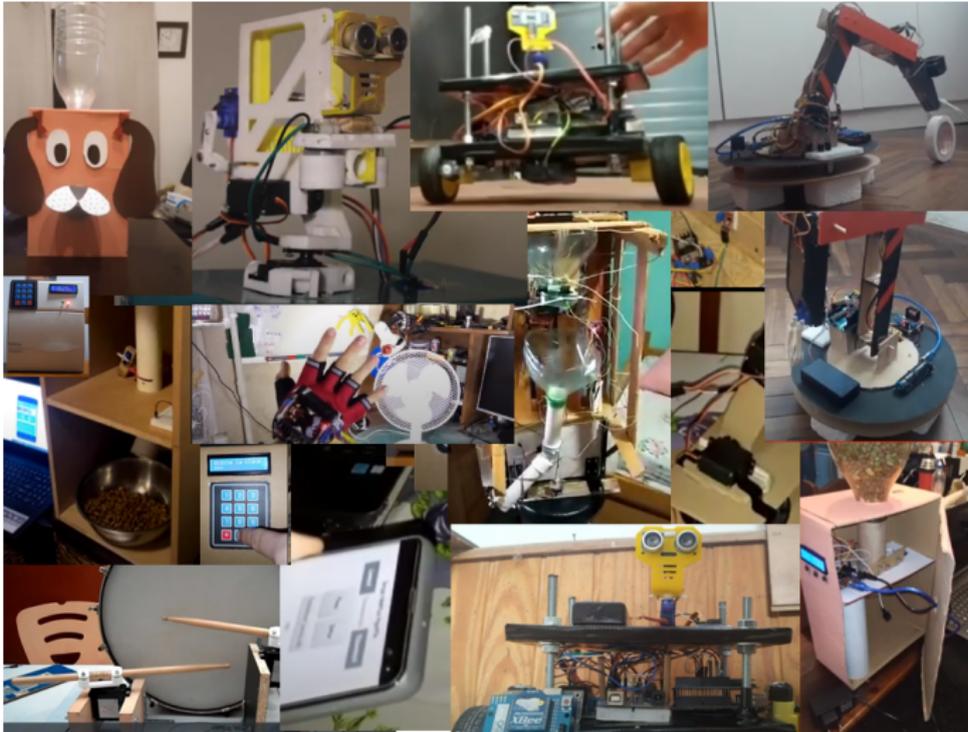
- Objetivos y características.
- Equipo docente.
- Sitio de cursos y foro de consultas.
- Metodología, horario de clases y consulta.
- Cronograma.
- Evaluación.
- Reglas internas básicas.

- Actividad orientada a la generación de ingreso.
- Pretende introducir a la Ingeniería Eléctrica, sus métodos, sus herramientas, sus actores.
 - Página del Instituto de Ingeniería Eléctrica (IIE):
<https://iie.fing.edu.uy/>
- De paso - datos básicos de la carrera:
 - Página de la carrera: <http://www.fing.edu.uy/carreras/grado/ingenieriaelectrica>
 - Noticias de Instituto de Ingeniería Eléctrica:
<https://iie.fing.edu.uy/noticias-del-iie/>
 - Director de carrera: Julian Oreggioni (juliano@fing.edu.uy).

- Pretende aportar motivación, experiencia de trabajo en equipo, comunicación oral, escrita y digital, a través de una metodología de enseñanza activa.
- **No se viene a escuchar, se viene a HACER!**
- ¿Qué se va a hacer?

- Pretende aportar motivación, experiencia de trabajo en equipo, comunicación oral, escrita y digital, a través de una metodología de enseñanza activa.
- **No se viene a escuchar, se viene a HACER!**
- **¿Qué se va a hacer?**

Objetivos y características



Proyectos finales TallerInE Robótico 20/21

Primer Semestre 2021

Primer Semestre 2020

Segundo Semestre 2020

- Se denotan dos etapas bien claras: *Introducción* y *Proyecto final*.
 - **Introducción:** serie de talleres para ir descubriendo el mundo de Arduino. Con énfasis en programación, aplicada al conocimiento y utilización de sensores y actuadores.
 - **Proyecto Final:** cada estudiante, o eventualmente cada grupo de estudiantes, desarrollaran proyectos basados en Arduino, con una tutoría docente "personalizada".
- Tanto para llevar a cabo los ejercicios de la primera etapa como para realizar los proyectos finales, se utilizará la plataforma Arduino, complementando con el simulador Tinkercad.
- Para la etapa final, recibirán un kit de materiales para que desarrollen el proyecto.
- Cuidar el material es responsabilidad de cada estudiante/grupo de estudiantes!!
(ver las *reglas* en el EVA -Reglas básicas-).

- Se denotan dos etapas bien claras: *Introducción* y *Proyecto final*.
 - **Introducción:** serie de talleres para ir descubriendo el mundo de Arduino. Con énfasis en programación, aplicada al conocimiento y utilización de sensores y actuadores.
 - **Proyecto Final:** cada estudiante, o eventualmente cada grupo de estudiantes, desarrollaran proyectos basados en Arduino, con una tutoría docente "personalizada".
- Tanto para llevar a cabo los ejercicios de la primera etapa como para realizar los proyectos finales, se utilizará la plataforma Arduino, complementando con el simulador Tinkercad.
- Para la etapa final, recibirán un kit de materiales para que desarrollen el proyecto.
- **Cuidar el material es responsabilidad de cada estudiante/grupo de estudiantes!!**
(ver las *reglas* en el EVA [-Reglas básicas-](#)).

Profesores

- Juan Sanchez
- Guillermo Airaldi
- Sebastián Montes de Oca

Ayudantes estudiantiles

- Cesar Azambuya
- Natalia Dominguez
- Alejo Bravo
- Mateo Guerrero

- El sitio de cursos: <https://eva.fing.edu.uy>
- Allí se busca el EVA del Tallere Robótico del primer semestre dentro de las asignaturas del Instituto de Ingeniería Eléctrica (IIE) (<https://eva.fing.edu.uy/course/view.php?id=1247>).
- Dentro del EVA, hay foros de información y foros de consulta.
- Además hay material de apoyo, están los ejercicios a realizar y es la plataforma a través de la cual se realiza buena parte de las evaluaciones.
- Y lo más importante: **es el canal de comunicación oficial del curso.**

- Las clases serán los ***Miércoles de 17:00 a 20:00*** horas en **Modalidad Presencial**, en el Laboratorio de Software del IIE.
- 1 Hora extra de consulta (a confirmar horario y modalidad)
- Serán en un formato que combina presentaciones de temas con mucho trabajo por parte del alumnado.
- Durante la primer etapa, habrá cuestionarios breves e individuales cada semana para afirmar lo incorporado en cada taller.
- Durante la segunda etapa habrá semanas con un esquema de trabajo más libre por parte de cada proyecto.

Cuatro Módulos.

- **Módulo 1:** Introducción a la programación y a Arduino (3-4 talleres).
- **Módulo 2:** Sensores y actuadores (2-3 talleres).
- **Módulo 3:** Comunicaciones (1-2 talleres).
- **Módulo 4:** Proyecto.

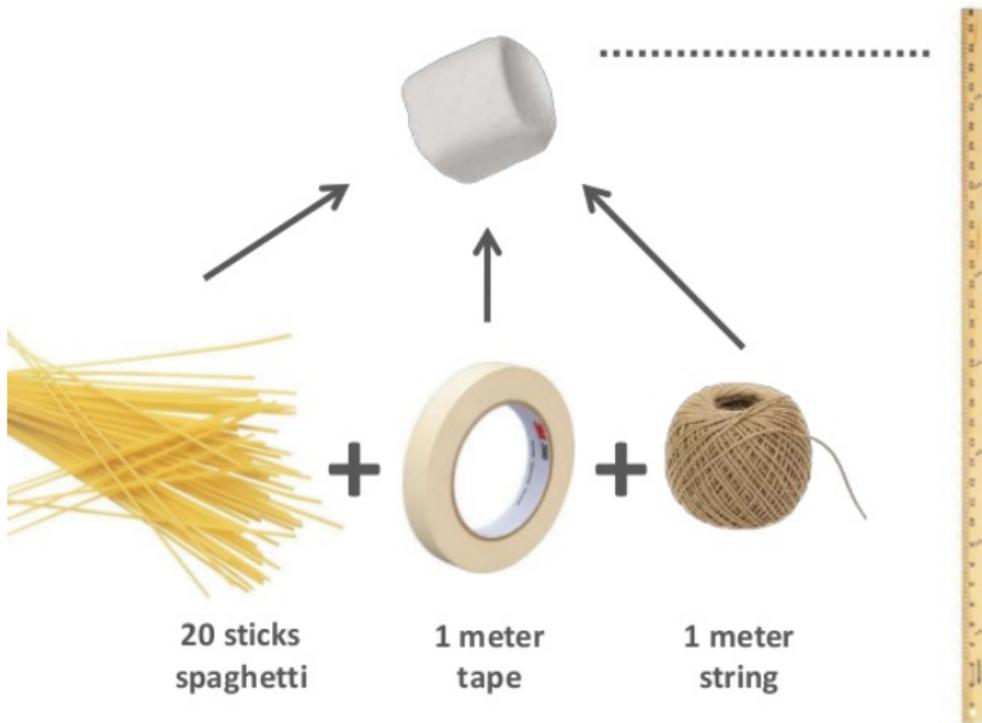
Evaluación:

cuestionarios, ejercicios, participación en clase, proyecto final: entregables y prototipo

- La evaluación de la asignatura es INDIVIDUAL y GRUPAL
- La metodología es la siguiente:
 - Cuestionarios (obligatorios) (5 %)
 - Ejercicios entregables (obligatorios) (15 %)
 - Prueba Parcial integradora (30 %)
 - Proyecto Final (50 %)
- En general, se indicarán lecturas previas a los talleres.
- Se darán ejercicios prácticos domiciliarios a ser compartidos al comienzo del siguiente taller.
- El módulo final tendrá como entregables: un informe, una presentación de apoyo visual y un vídeo.

Rompamos el hielo!- El desafío del Malvadisco

Cuenta regresiva - 18 minutos



[Video de cierre](#)

- Ante cualquier duda usar el *Foro de consultas*. Así, se pueden ayudar entre los estudiantes. Además una duda personal, puede cubrir la duda de más estudiantes.
- Recordar que la asistencia a las clases es **obligatoria** (Max. 2 faltas no justificadas).

1 Motivación

2 Hardware

3 Software

- Introducción
- Entorno de desarrollo: Arduino IDE
- Simulador on-line: TinkerCad
- Ejercicios para hacer en el taller
- Ejercicio para hacer en casa:

1 Motivación

2 Hardware

3 Software

- Introducción
- Entorno de desarrollo: Arduino IDE
- Simulador on-line: TinkerCad
- Ejercicios para hacer en el taller
- Ejercicio para hacer en casa:

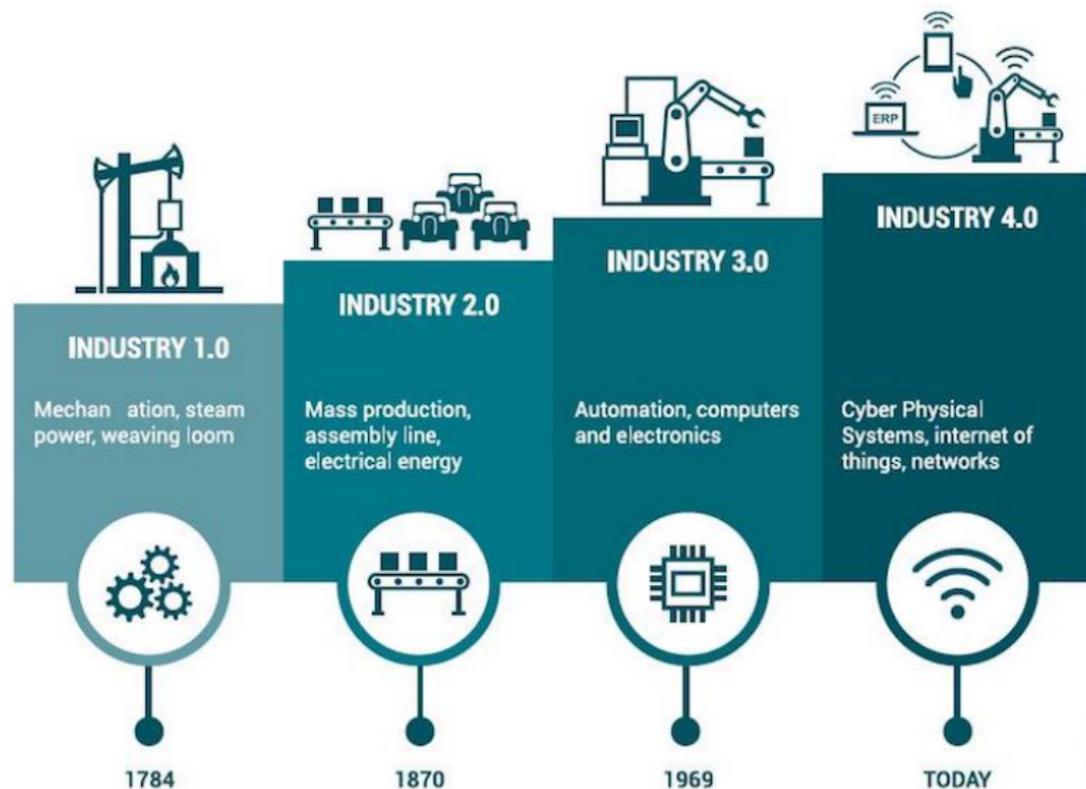
Piensen en cómo se fabricaban los autos hace 50 años y cómo se fabrican hoy. ¿Qué cambió?



¿Cómo creen que se activan las alarmas o los sensores en una línea de producción?



Industria 4.0 - Áreas de aplicación



SHUTTERSTOCK



INDUSTRIAL 4.0



Automation



Big Data



Cloud Computing



Autonomous



IoT



Data Management

Alarma de Seguridad

"Tu equipo de ingeniería debe instalar un sistema de alarma en una bodega. Cuando alguien presiona un botón de emergencia, se debe encender un LED rojo y sonar un buzzer."

- ¿Cómo podemos hacer esto con Arduino?
- ¿Qué elementos necesitamos?

Alarma de Seguridad

"Tu equipo de ingeniería debe instalar un sistema de alarma en una bodega. Cuando alguien presiona un botón de emergencia, se debe encender un LED rojo y sonar un buzzer."

- ¿**Cómo** podemos hacer esto con Arduino?
- ¿**Qué** elementos necesitamos?

Alarma de Seguridad

"Para resolver este problema, necesitamos algo que **procese las señales** del botón y **controle** el LED y el buzzer. Aquí es donde entra un **microcontrolador**."

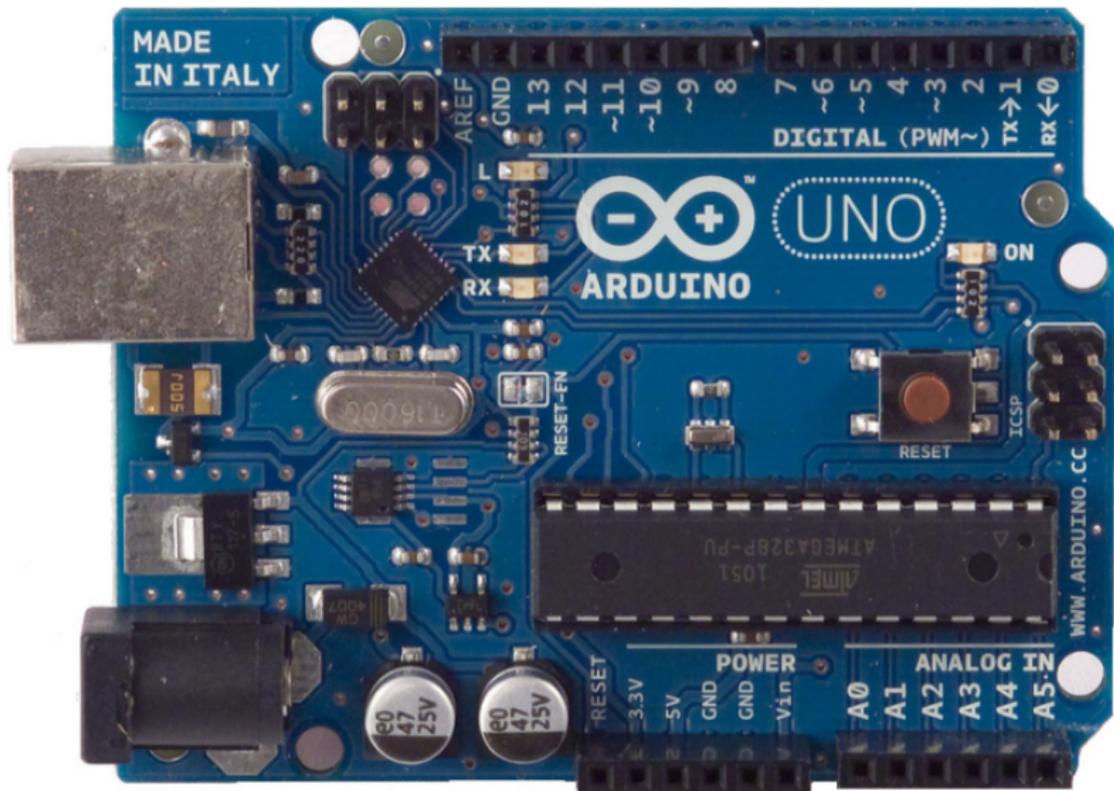
- ¿Qué es un microcontrolador?
- ¿Para qué sirve?
- ¿Qué es Arduino?

Alarma de Seguridad

"Para resolver este problema, necesitamos algo que **procese las señales** del botón y **controle** el LED y el buzzer. Aquí es donde entra un **microcontrolador**."

- ¿Qué es un microcontrolador?
- ¿Para qué sirve?
- ¿Qué es Arduino?

Arduino Uno



¿Para qué sirve un microcontrolador (μC)?

Algunas de las muchas posibles aplicaciones:

- Robótica.
- Adquisición de datos
- Control de un invernadero.
- Control de vehículos.
- Control de un proceso industrial: temperatura, pH, nivel de un tanque, motores, etc.
- Sintetizador de sonidos.
- Domótica.

Basada en el **software/hardware libre**. Permite:

- Estudiar el hardware para entender cómo funciona.
- Hacer modificaciones al hardware.
- Poder compartir esas modificaciones con la comunidad.



WHAT IS ARDUINO?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects.

[Learn more about Arduino](#)



ARDUINO BOARD

Arduino senses the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors, and other actuators.

[Discover the official Arduino boards](#)



ARDUINO SOFTWARE

You can tell your Arduino what to do by writing code in the Arduino programming language and using the Arduino development environment.

[Download the Arduino Software](#)

CLOSE

BUY AN ARDUINO

LATEST



Consecuencias:

- Comunidad activa de usuarios y desarrolladores (foros, ejemplos, tutoriales, etc.).
- Gran cantidad de software y hardware disponible para autoconstruir.
- Precios bajos.
- Existe mucho material en el Web y muchos proyectos interesantes que pueden hacer.

1 Motivación

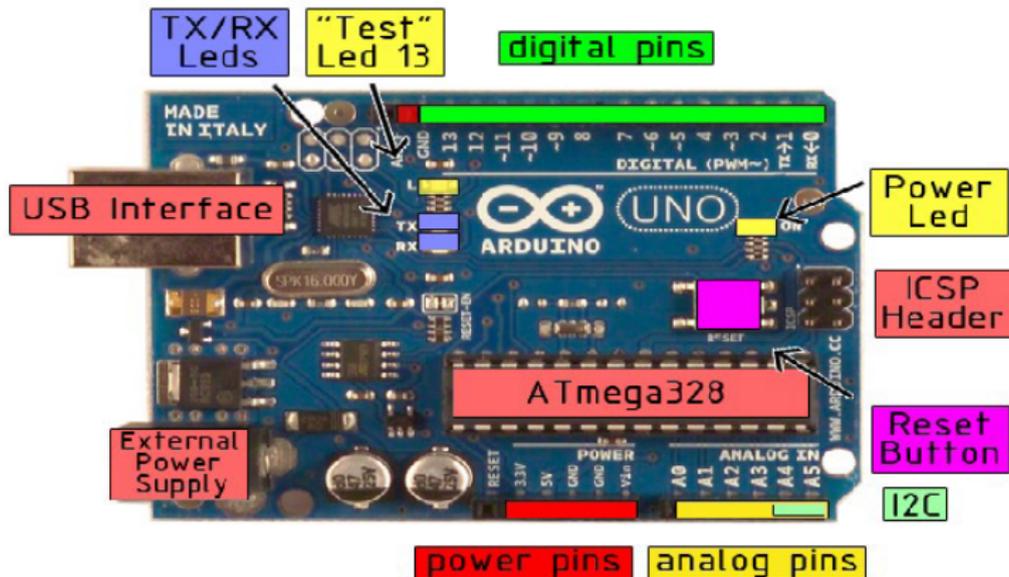
2 Hardware

3 Software

- Introducción
- Entorno de desarrollo: Arduino IDE
- Simulador on-line: TinkerCad
- Ejercicios para hacer en el taller
- Ejercicio para hacer en casa:

Microcontrolador

Arduino UNO



Microcontrolador

Características

- Microcontrolador: ATmega328 (**8 bits**).
- Alimentación via USB (5 V) o independiente (7-12 V).
- 14 pines de entrada/salida (I/O) digitales (6 con Pulse Width Modulation: PWM).
- Corriente máxima por pin I/O: **40 mA**.
- 6 pines de entrada analógica.
- Permite comunicación serial.
- Memoria flash: 32 KB.
- Frecuencia del reloj: **16 MHz**.

1 Motivación

2 Hardware

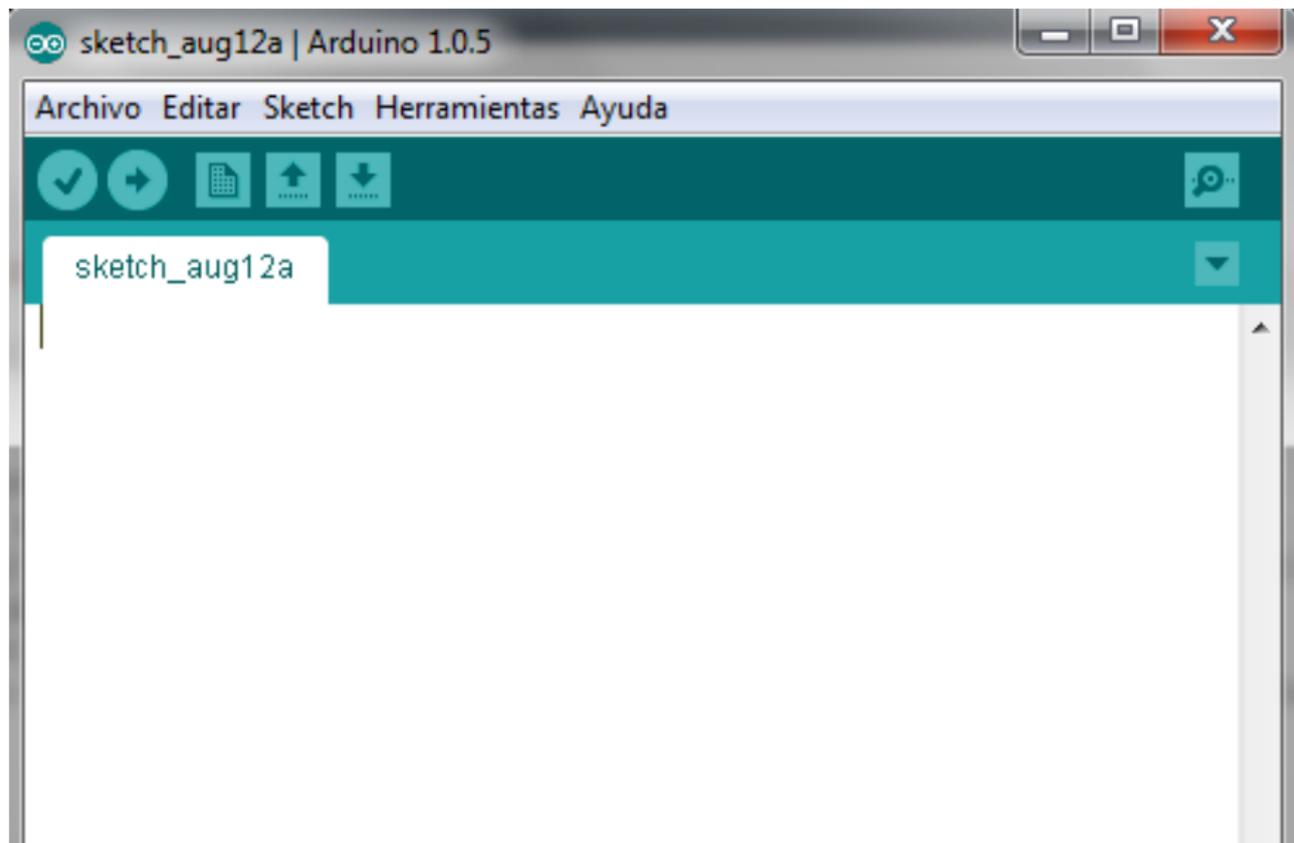
3 Software

- Introducción
- Entorno de desarrollo: Arduino IDE
- Simulador on-line: TinkerCad
- Ejercicios para hacer en el taller
- Ejercicio para hacer en casa:

- Basado en Wiring y similar a C++.
- Sketches = Código fuente.
- Lenguaje compilado.

Entorno de desarrollo: Arduino IDE

-a utilizarse durante la etapa de proyectos-



Entorno de desarrollo: Arduino IDE

Proceso: Código fuente



The screenshot shows the Arduino IDE 1.0.5 interface. The title bar reads "sube_y_baja_brillo_con_pote_comunicacion_serial | Arduino 1.0.5". The menu bar includes "Archivo", "Editar", "Sketch", "Herramientas", and "Ayuda". The toolbar contains icons for a checkmark, a right arrow, a grid, an upload arrow, a download arrow, and a help icon. The sketch name "sube_y_baja_brillo_con_pote_comunicacion_serial" is displayed in the top right of the editor area. The code in the editor is as follows:

```
const int led = 3;
const int pot = 0;

int brillo;

void setup(){
  Serial.begin(9600);
  pinMode(led, OUTPUT);
  //los pines analogicos se declaran como entradas automaticamente
}

void loop(){
  brillo = analogRead(pot)/4; //entre 0 y 1023
```

Entorno de desarrollo: Arduino IDE

Proceso: Código fuente -> Compilación ("verify")



```
sube_y_baja_brillo_con_pote_comunicacion_serial | Arduino 1.0.5
Archivo Editar Sketch Herramientas Ayuda
sube_y_baja_brillo_con_pote_comunicacion_serial
const int led = 3;
const int pot = 0;

int brillo;

void setup(){
  Serial.begin(9600);
  pinMode(led, OUTPUT);
  //los pines analogicos se declaran como entradas automaticamente
}

void loop(){
  brillo = analogRead(pot)/4; //entre 0 y 1023
```

Entorno de desarrollo: Arduino IDE

Proceso: Código fuente -> Compilación ("verify") -> Programarlo en la placa ("upload")



```
sube_y_baja_brillo_con_pote_comunicacion_serial | Arduino 1.0.5
Archivo Editar Sketch Herramientas Ayuda
sube_y_baja_brillo_con_pote_comunicacion_serial
const int led = 3;
const int pot = 0;

int brillo;

void setup(){
  Serial.begin(9600);
  pinMode(led, OUTPUT);
  //los pines analogicos se declaran como entradas automaticamente
}

void loop(){
  brillo = analogRead(pot)/4; //entre 0 y 1023
```

Encender y apagar un LED

- Para este ejercicio se debe utilizar **ChatGPT** para encender y apagar un LED cada un segundo.
- Cargar el programa al controlador Arduino y probar su funcionamiento.
- Utilizar primero el pin 13 y luego armar el circuito utilizando el pin de control 2 para conectar el LED.

Simulador on-line: TinkerCad

Comenzar un diseño -asumiendo que cada estudiante ya tiene su cuenta creada-

The screenshot shows the Tinkercad website interface. At the top left is the logo for TINKERCAD, part of the Autodesk ecosystem. The top right contains navigation links for 'Galería', 'Blog', 'Aprendizaje', and 'Enseñanza', along with a search icon and a user profile icon. On the left side, there is a user profile section with a search bar labeled 'Buscar diseños...', a list of categories including 'Diseños 3D', 'Circuits' (highlighted with a yellow arrow), 'Bloques de código', and 'Lecciones', and a 'Proyectos' section with a 'Crear proyecto' button. The main content area is titled 'Circuits' and features a green button labeled 'Crear nuevo circuito' (highlighted with a yellow arrow) and a 'Select' checkbox. Below this are eight circuit design thumbnails, each with a title, creation date, and privacy status. The thumbnails include: 'TRtaller4Ej1d' (Publico, 5 meses), 'Copy of Ej2' (Privado, 5 meses), 'Super Leelo-Robo' (Privado, 5 meses), 'Magnificent Lappi-Wolt' (Privado, 5 meses), 'L293ybotonesOK' (Privado, 5 meses), 'TRtaller4Ej1a' (Publico, 5 meses), 'TR_T2ej2' (Privado, 5 meses), and 'TR_T2ej1' (Privado, 5 meses).

Simulador on-line: TinkerCad

Renombrar el nuevo proyecto, buscar e instanciar el microcontrolador

miProyecto01

Se han guardado todos los cambios.

Código ▶ Iniciar simulación Exportar Compartir

Componentes
Todos

arduino

Microcontroladores

Arduino Uno R3 ATTiny

Simulador on-line: TinkerCad

Acceder a la sección de codificación -en general se usará solo *texto*-

miProyecto01

Se han guardado todos los cambios.

Código ▶ Iniciar simulación ▶ Exportar ▶ Compartir

Bloques + texto

Bloques

- Bloques + texto
- Texto

```
1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10  digitalWrite(13, LOW);
11  delay(1000); // Wait for 1000 millisecond(s)
12 }
```

Simulador on-line: TinkerCad

Compilar código y condicionalmente empezar la simulación

miProyecto01

Se han guardado todos los cambios.

Código ▶ Iniciar simulación Exportar Compartir

1 (Arduino Uno R3)

```
1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10  digitalWrite(13, LOW);
11  delay(1000); // Wait for 1000 millisecond(s)
12 }
```

Bloques + texto

- Salida
- Entrada
- Notación
- Control
- Matemáticas
- Variables

definir LED integrado en ALTA

definir pasador 0 en ALTA

definir pasador 3 en 0

girar servo en el pasador 0 a 0

reproducir altavoz en el pasador 0

desactivar el altavoz en pasador 0

imprimir en monitor en serie hello world

definir LED RGB de pasadores 3

esperar 1 segundos

definir LED integrado en BAJA

esperar 1 segundos

Monitor en serie

Simulador on-line: TinkerCad

Detener simulación para volver a editar código

The screenshot shows the TinkerCad online simulator interface. At the top, the project name is 'miProyecto01'. The status bar indicates 'Se han guardado todos los cambios.' and shows the simulation time as 'Hora de simulador: 00:00:21'. The 'Detener simulación' button is highlighted with a yellow arrow. The code editor on the right shows the following code:

```
1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10  digitalWrite(13, LOW);
11  delay(1000); // Wait for 1000 millisecond(s)
12 }
```

The left sidebar contains a 'Bloques + texto' panel with various blocks for defining pins, controlling LEDs, and setting delays. A yellow highlight is also present on the 'Hora de simulador' text.

Simulador on-line: Tinkercad

¿Cómo compartir un proyecto adecuadamente?



Circuits

Crear nuevo circuito

Modificar

miProyecto01
hace 6 minutos
Privado

Opciones

TF
ha
Pú



Circuits

Crear nuevo circuito

Modificar

miProyecto01
hace 6 minutos
Privado

Propiedades...
Duplicar
Mover a proyecto...
Eliminar

hace 5 meses
Público

Privacidad

Público Viewable and discoverable by everyone



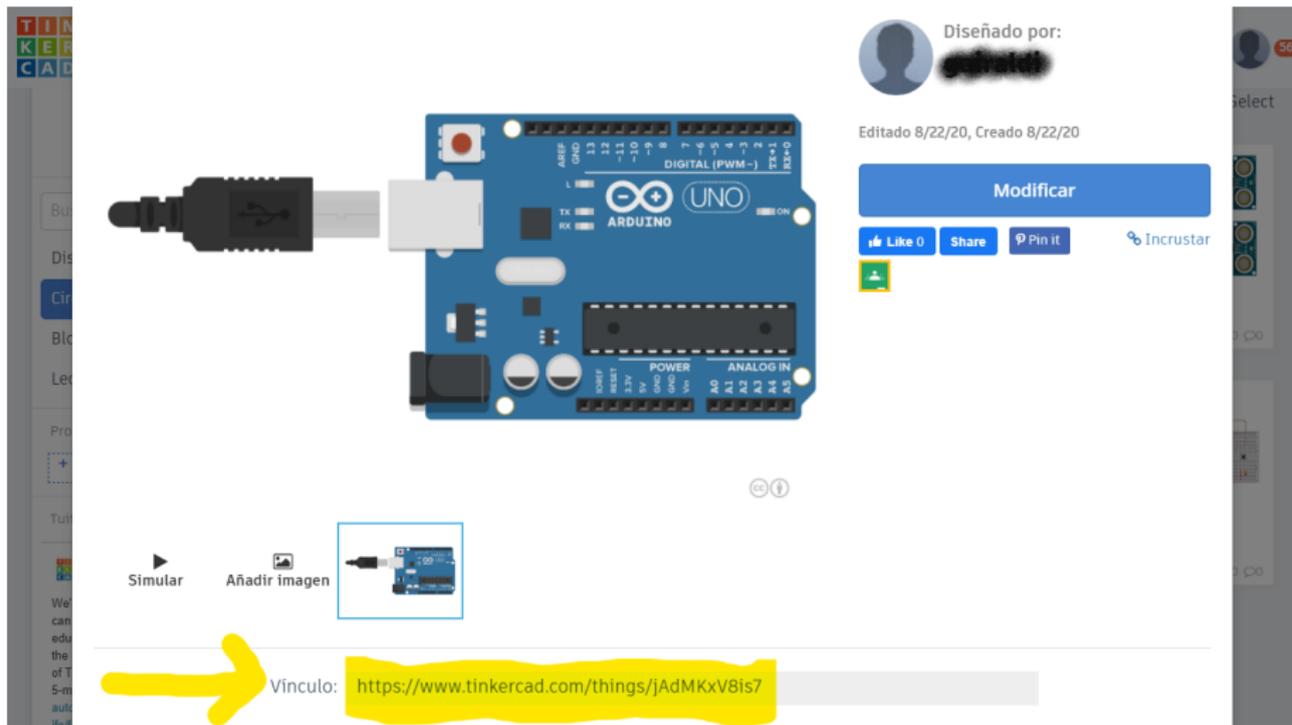
Circuits

Crear nuevo circuito

miProyecto01
hace 6 minutos
Público

Simulador on-line: TinkerCad

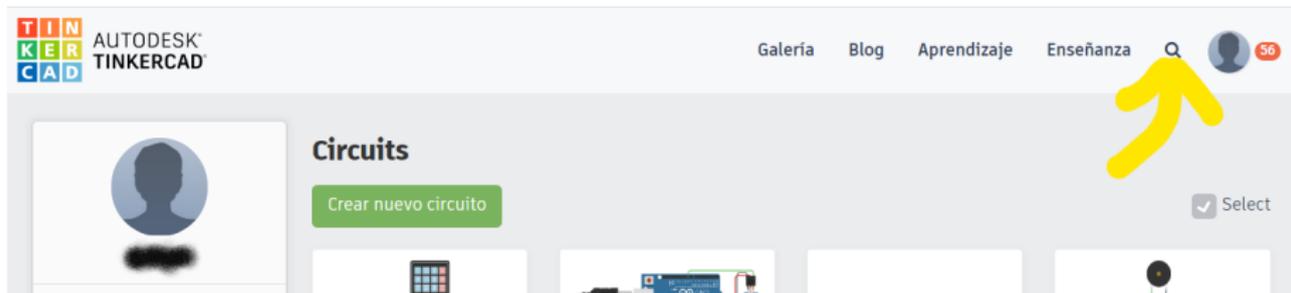
¿Cómo compartir un proyecto adecuadamente?



The screenshot displays the TinkerCad online simulator interface. In the center, a blue Arduino Uno board is shown with a USB Type-C cable connected to its left side. Below the board, there are two buttons: "Simular" (with a play icon) and "Añadir imagen" (with a plus icon). To the right of the "Añadir imagen" button is a small thumbnail of the Arduino board, which is highlighted with a blue border. Below these buttons, a yellow arrow points to a text field labeled "Vínculo:" containing the URL <https://www.tinkercad.com/things/jAdMkxV8is7>. The URL is highlighted in yellow. On the right side of the interface, there is a user profile section with a silhouette icon, the text "Diseñado por:", and a blurred name. Below this, it says "Editado 8/22/20, Creado 8/22/20". There is a blue "Modificar" button, and below it are buttons for "Like 0", "Share", "Pin it", and "Incrustar".

Simulador on-line: Tinkercad

¿Cómo buscar un proyecto por su nombre?



Ejercicios para hacer en el taller

Ejercicio 1:

Se visitarán distintos proyectos brindados por los docentes (listados en las siguiente slides), con el objetivo que, repartidos en grupos de 3 (-4) estudiantes, puedan:

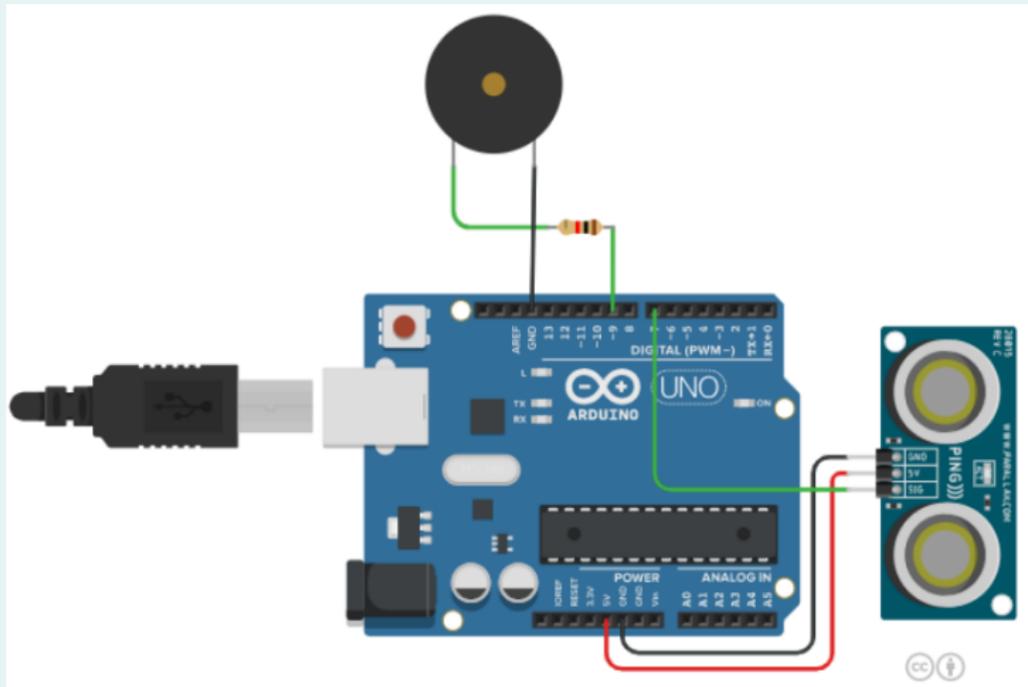
- 1 comenzar a interactuar con el simulador,
- 2 simular el comportamiento y observar la funcionalidad de los distintos circuitos,
- 3 interpretar los códigos, tratando de reconocer la estructura general de éstos y descubriendo la funcionalidad de las distintas instrucciones,
- 4 conocer y clasificar los materiales, interpretando cuáles le dan información al microcontrolador y sobre cuáles el microcontrolador actúa.

A continuación, se presentan los distintos proyectos a visitar. **Para cada uno, buscar en Tinkercad por el nombre del proyecto indicado.**

Ejercicio 1

a. Ultrasonido y Buzzer

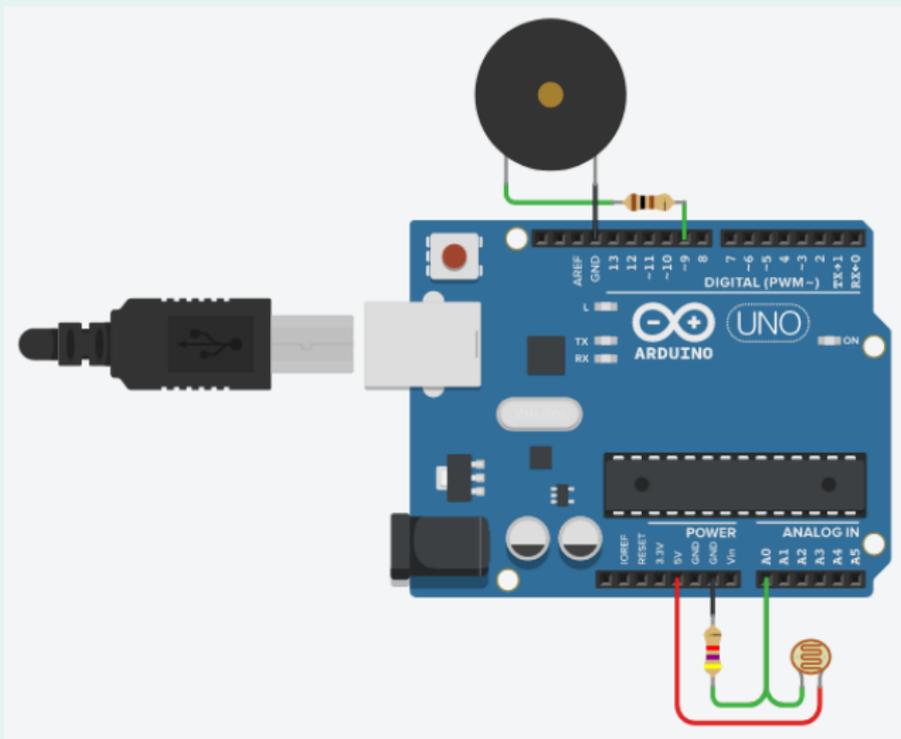
Nombre proyecto: "TRtaller0Ej1a"



Ejercicio 1

b. LDR y Buzzer

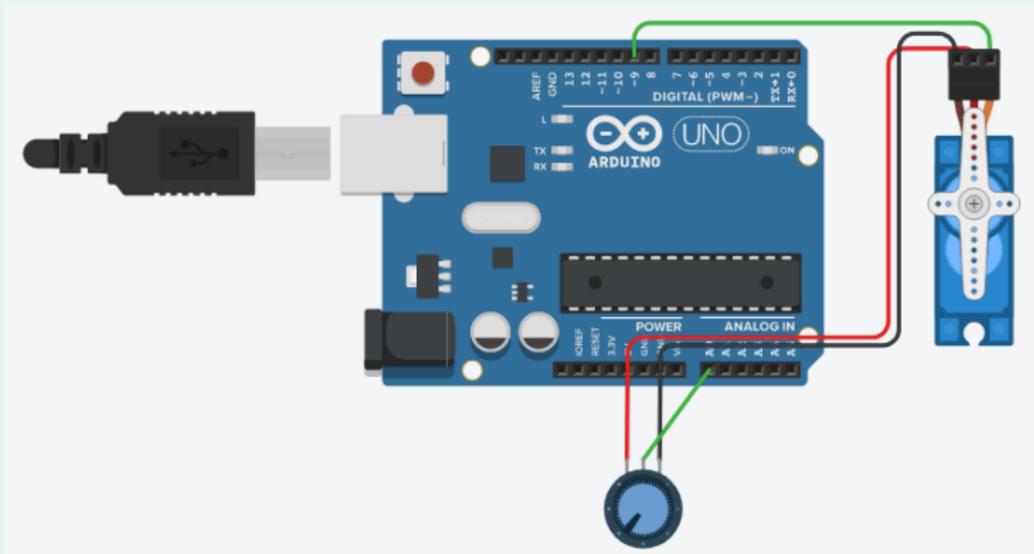
Nombre proyecto: "TRtaller0Ej1b"



Ejercicio 1

c. Potenciómetro y Servomotor

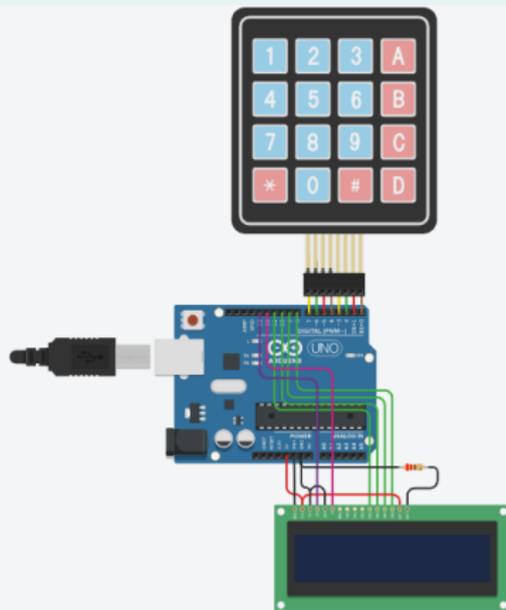
Nombre proyecto: "TRtaller0Ej1c"



Ejercicio 1

d. Calculadora

Nombre proyecto: "TRtaller0Ej1d"



Problema

Tu equipo de ingeniería está desarrollando un sistema de señales para un satélite en órbita. Para verificar que la transmisión de datos sigue activa, deben hacer que un LED parpadee con una secuencia específica.

Objetivo: Implementar en el simulador el circuito de la figura y escribir un programa para encender y apagar un LED con una secuencia especial.

Niveles de Desafío:

- **Nivel 1:** Hacer que el LED parpadee cada 1 segundo usando `delay(x)`.
- **Nivel 2:** Modificarlo para que haga 3 parpadeos rápidos y luego una pausa de 2 segundos.
- **Extra 1:** Agregar un botón que, al presionarlo, haga que el LED parpadee más rápido.
- **Extra 2 1:** Si el botón se mantiene presionado más de 3 segundos, el LED entra en un **modo secreto** con un nuevo patrón de luces.

Resumen para la próxima clase:

- 1 Si no se terminaron los ejercicios para hacer en este taller, terminarlos.
- 2 Tener funcionando el Ejercicio 1 de deberes para compartir en la siguiente clase (por dudas utilizar el *Foro de consultas*).
- 3 Queda disponible un cuestionario sobre esta clase, que deberá ser completado en el sitio EVA. Lo deberá hacer cada estudiante individualmente!!
- 4 Se recomienda continuar con la lectura de la documentación sugerida en la sección de *Introducción* en el sitio de EVA. En particular, leer sobre tipos de datos, variables y control de flujo.