



Software de desarrollo de Arduino

INTEGRANTES:

- CÉSAR ANTONIO IX SÁNCHEZ
- ADALBERTO ENAI PECH US
- FOSTER MEDINA SERGIO DAVID
- CARLOS ADRIAN SALAZAR SANTOYO
- IVÁN JOSÉ RODRIGUEZ MALDONADO

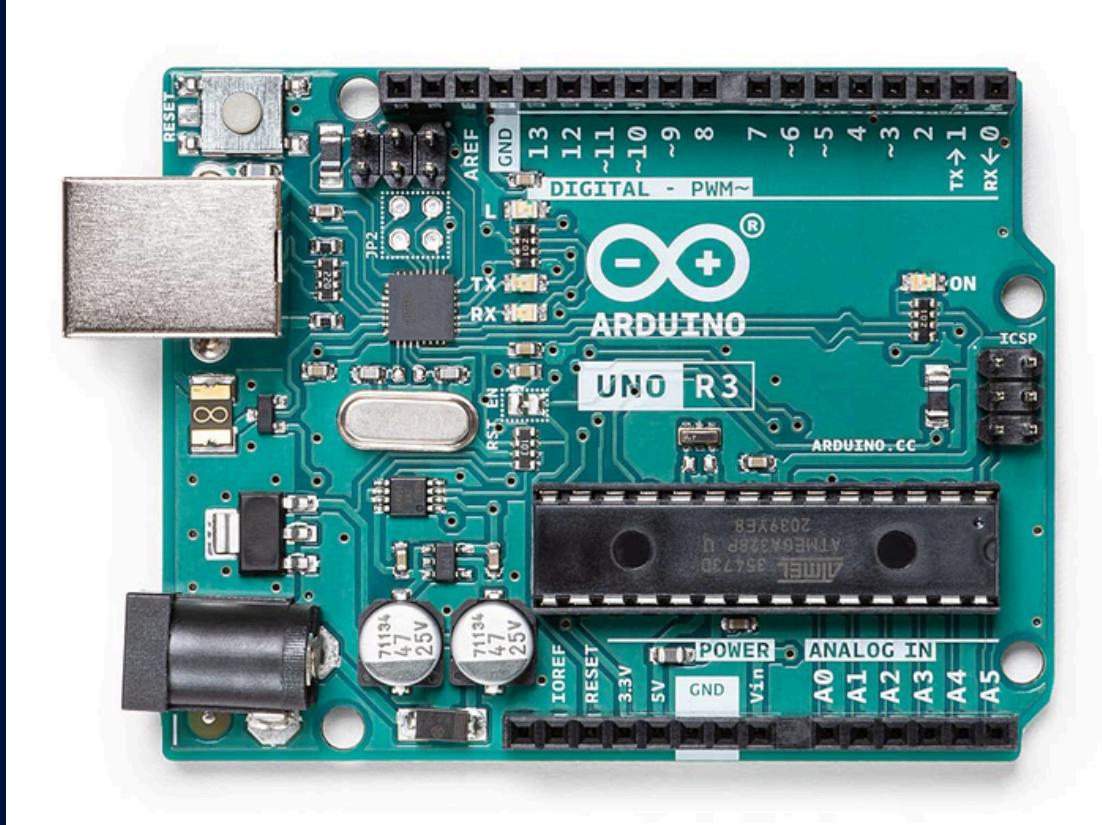
1. Introducción a Arduino IDE

El Arduino IDE (Integrated Development Environment) es el software utilizado para escribir, compilar y cargar programas a la placa Arduino. Es ideal para principiantes, ya que se puede programar utilizando un lenguaje basado en C++ de manera sencilla.

El IDE Arduino permite interactuar con la placa, acceder a bibliotecas y cargar el código fácilmente.

La facilidad de conexión, el código simple y el gran número de bibliotecas disponibles hacen que el Arduino UNO sea una opción excelente para empezar con proyectos de hardware.

Tiene una amplia comunidad.



2. Características del Software

Lenguaje de Programación:

- El Arduino IDE utiliza una variación simplificada de C/C++ para programar las placas Arduino. Aunque se basa en C++.



Funciones Principales:

El código de Arduino se estructura en dos funciones principales:

- **setup()**: Se ejecuta una vez al inicio del programa para realizar la configuración inicial.
- **loop()**: Se ejecuta de manera continua y contiene el código que se ejecuta repetidamente.

Funciones específicas de Arduino como **digitalWrite()**, **analogRead()**, y **delay()**, son parte del núcleo del software, permitiendo el control de pines, la lectura de sensores y la temporización.

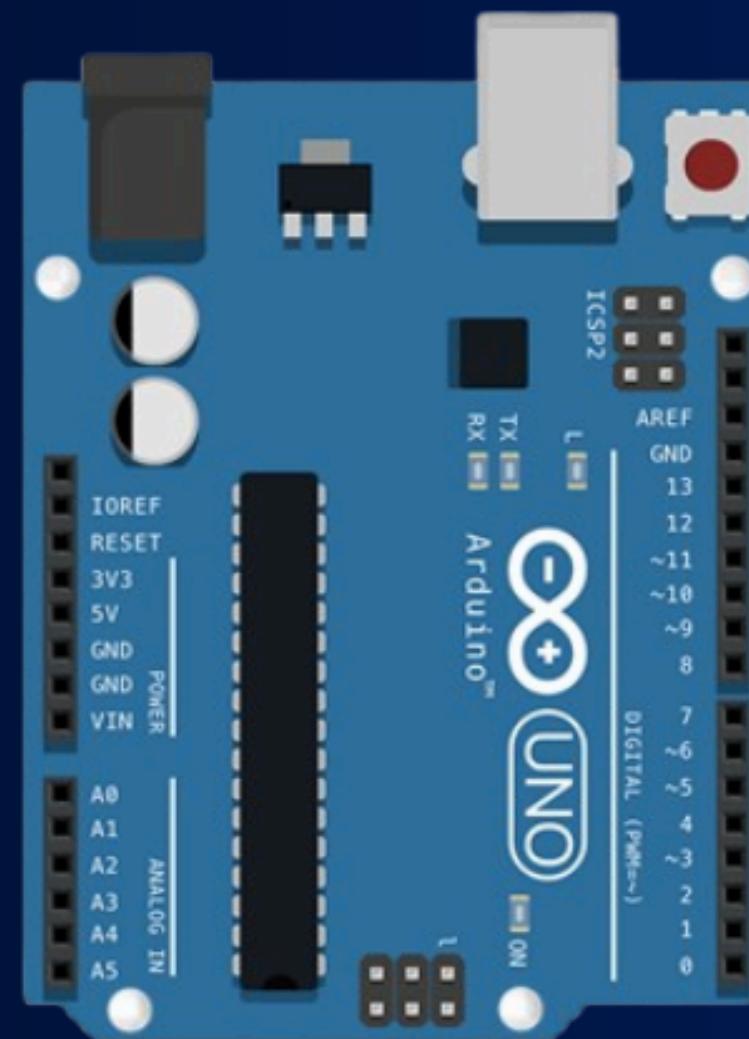
2. Características del Software

Clases y Estructuras:

- No soporte completo de OOP (Programación Orientada a Objetos).
- Estructuras: En lugar de clases, se utilizan estructuras (struct) y funciones para organizar el código de forma modular.

Capacidades del Software:

- Editor de Código
- Compilación: Transforma el código fuente en un archivo binario que puede ser cargado en la placa.
- Depuración: A través del monitor serial, puedes ver los datos que envía el Arduino a la computadora.
- Compatibilidad con bibliotecas: Puedes instalar y usar bibliotecas predefinidas que amplían las funcionalidades del Arduino



3. Proceso de programación

- Paso 1: Escribir el código en el Arduino IDE utilizando C/C++:

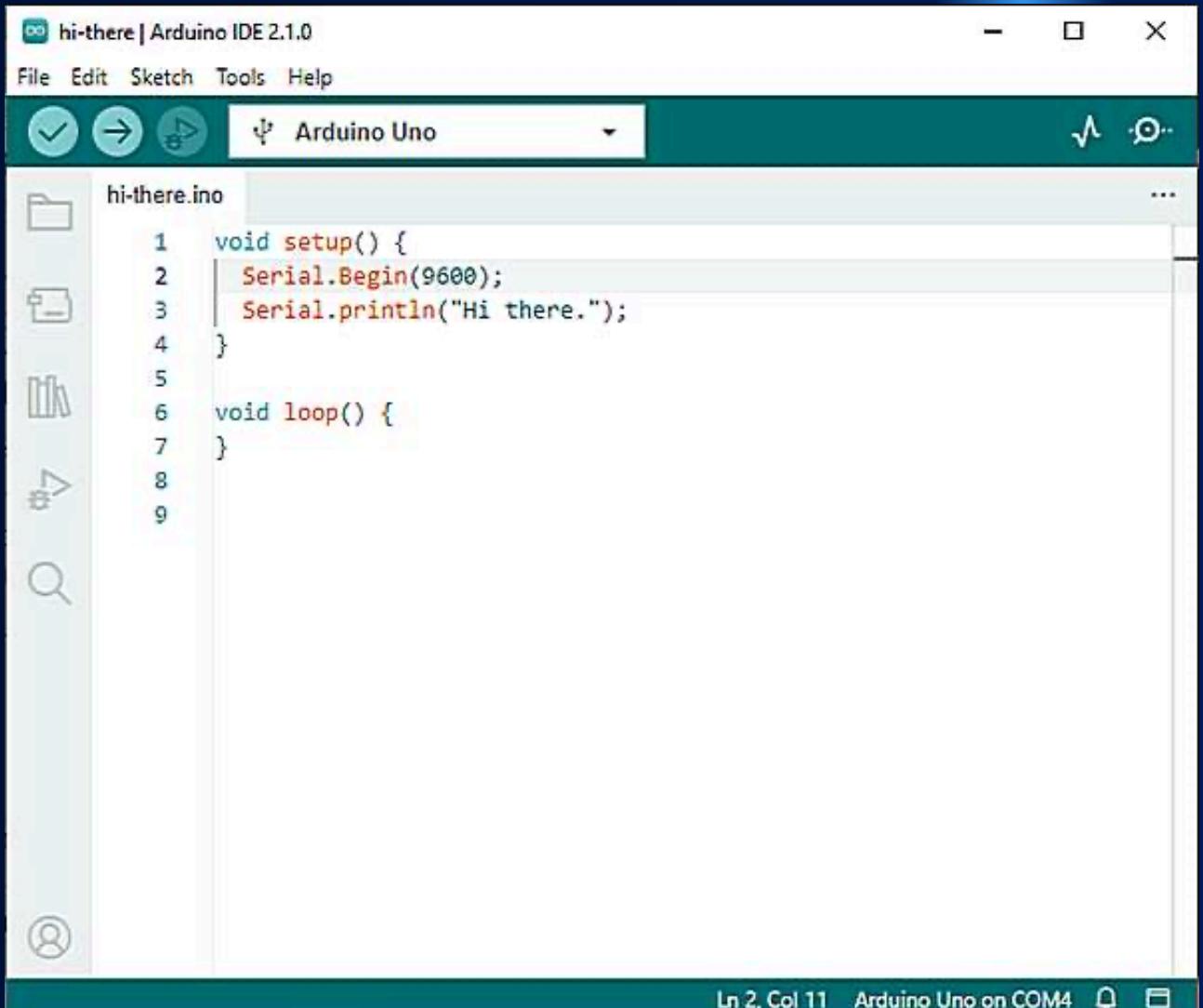
El código debe estar compuesto por dos funciones principales: `setup()` (que se ejecuta una vez) y `loop()` (que se ejecuta continuamente).

- Paso 2: Seleccionar la placa y el puerto desde el menú del IDE:

Esto se realiza a través de la opción "Herramientas" > "Placa" y "Puerto".

- Paso 3: Compilar el código:

El IDE verifica si hay errores en el código antes de enviarlo a la placa.



The screenshot shows the Arduino IDE interface with the title bar 'hi-there | Arduino IDE 2.1.0'. The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for Open, Save, and Upload. The board selector dropdown shows 'Arduino Uno'. The left sidebar has icons for File, Folder, Bookmarks, and Help. The main code editor window displays the following code:

```
1 void setup() {
2     Serial.Begin(9600);
3     Serial.println("Hi there.");
4 }
5
6 void loop() {
7 }
8
9
```

The status bar at the bottom right shows 'Ln 2, Col 11' and 'Arduino Uno on COM4'.

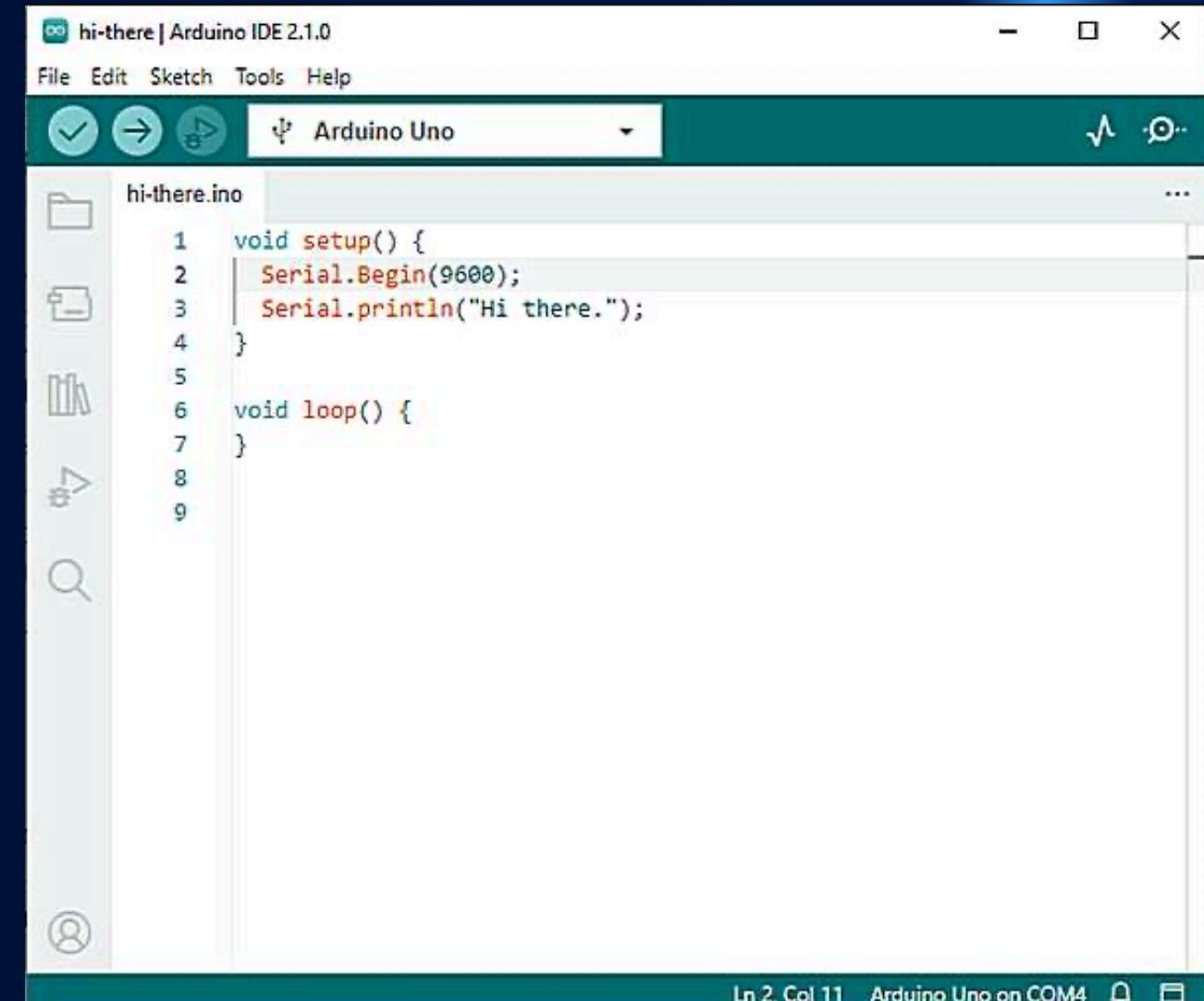
3. Proceso de programación

- Paso 4: Cargar el código a la placa Arduino UNO mediante el cable USB:

El IDE transfiere el código al microcontrolador en el Arduino UNO, permitiendo que el hardware ejecute las instrucciones.

- Paso 5: Verificación y depuración:

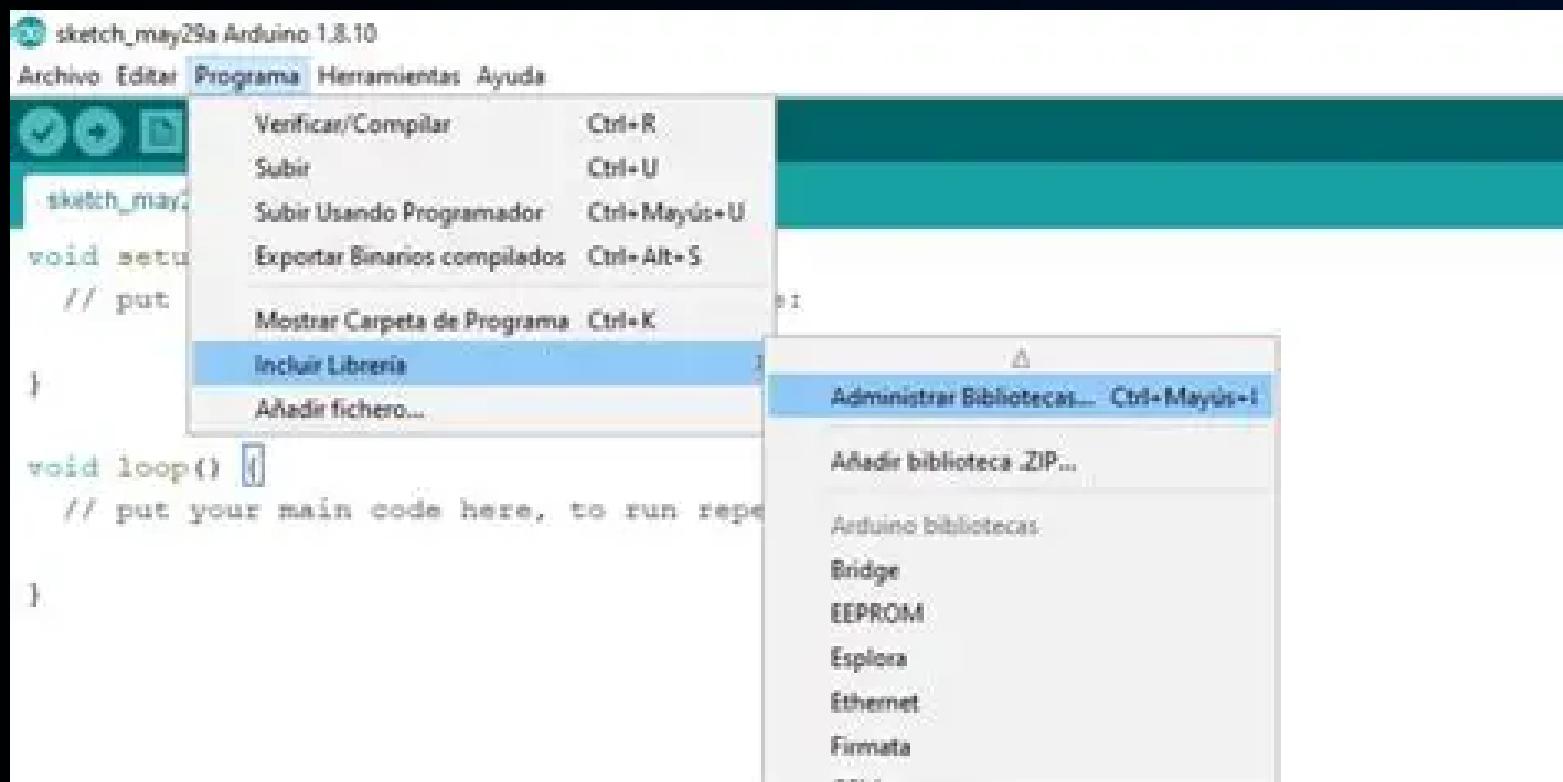
Uso del Monitor Serial para verificar la salida del código y depurar posibles errores en tiempo real.



```
void setup() {  
  Serial.begin(9600);  
  Serial.println("Hi there.");  
}  
  
void loop() {}
```

The screenshot shows the Arduino IDE interface with the title bar "hi-there | Arduino IDE 2.1.0". The toolbar includes icons for file operations, sketch tools, and serial monitoring. The dropdown menu shows "Arduino Uno" selected. The main area displays the code for "hi-there.ino". The status bar at the bottom right indicates "Ln 2, Col 11" and "Arduino Uno on COM4".

4. Uso de Bibliotecas en el Arduino



Las bibliotecas son colecciones de funciones predefinidas que amplían las capacidades del Arduino. Permiten utilizar componentes como sensores, pantallas y módulos sin necesidad de escribir todo el código desde cero.

Puedes poner tus propias librerías.

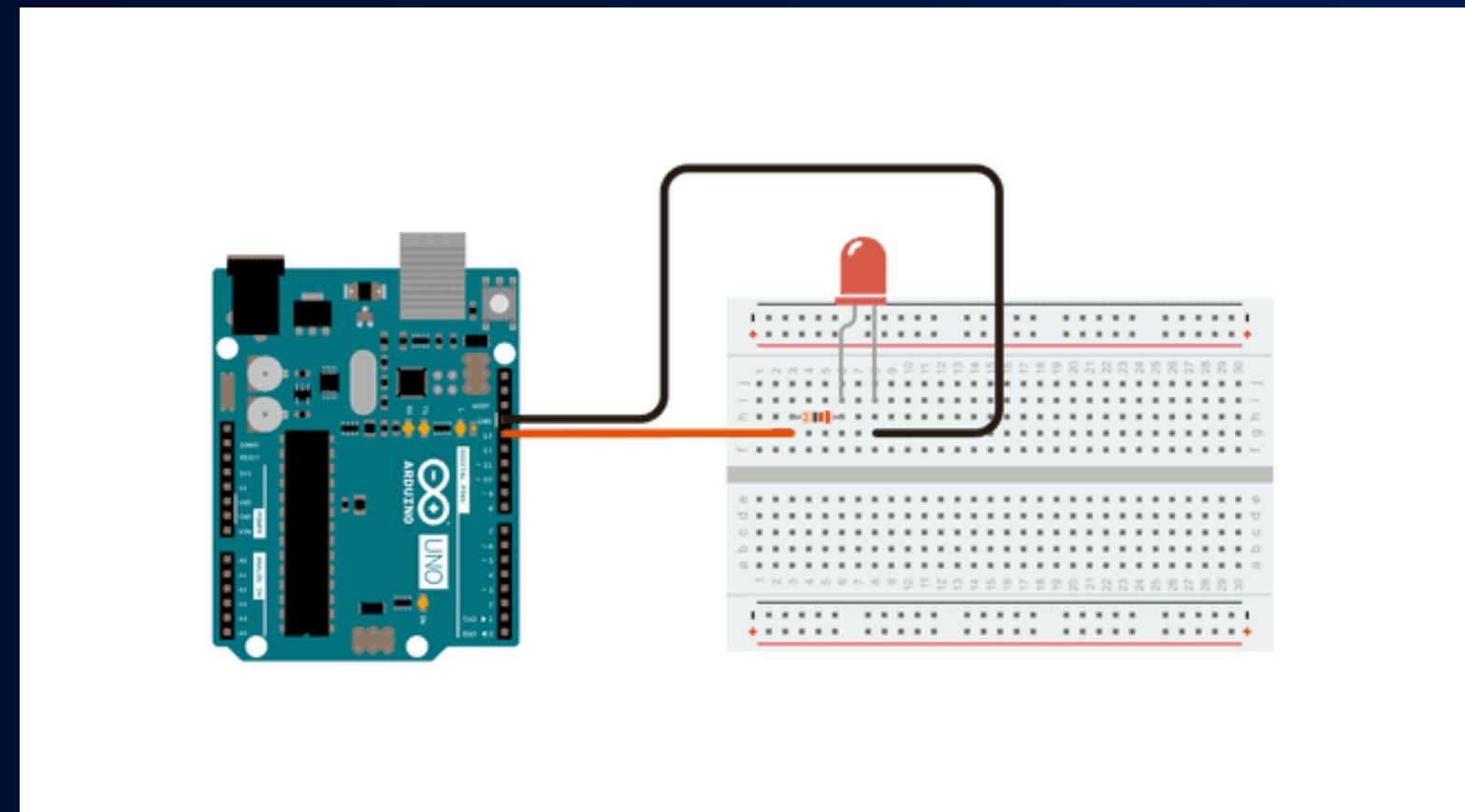
Algunas bibliotecas populares incluyen:

- **Wire:** Para comunicación I2C.
- **Servo:** Para controlar servomotores.
- **LiquidCrystal:** Para manejar pantallas LCD.
- **SPI:** Para comunicación con dispositivos mediante el protocolo SPI.

5. Proyectos comunes con Arduino

Proyectos para principiantes:

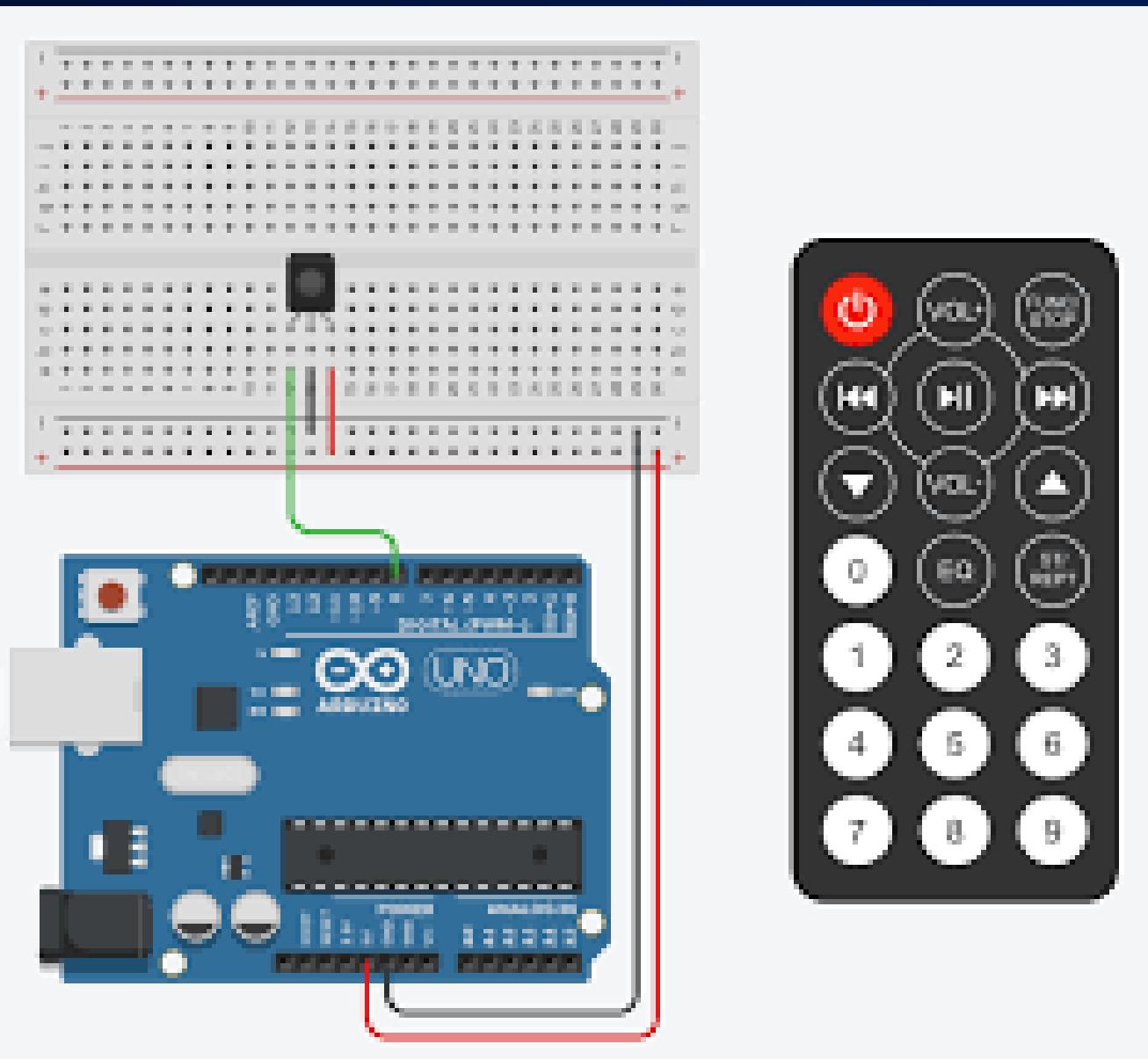
- Blink: Parpadeo de un LED, ejemplo básico de control de salidas.
- Semáforo: Control de luces de tráfico utilizando LEDs.
- Control de Servomotor: Mover un servo motor en función de un sensor o un botón.



5. Proyectos comunes con Arduino

Proyectos intermedios:

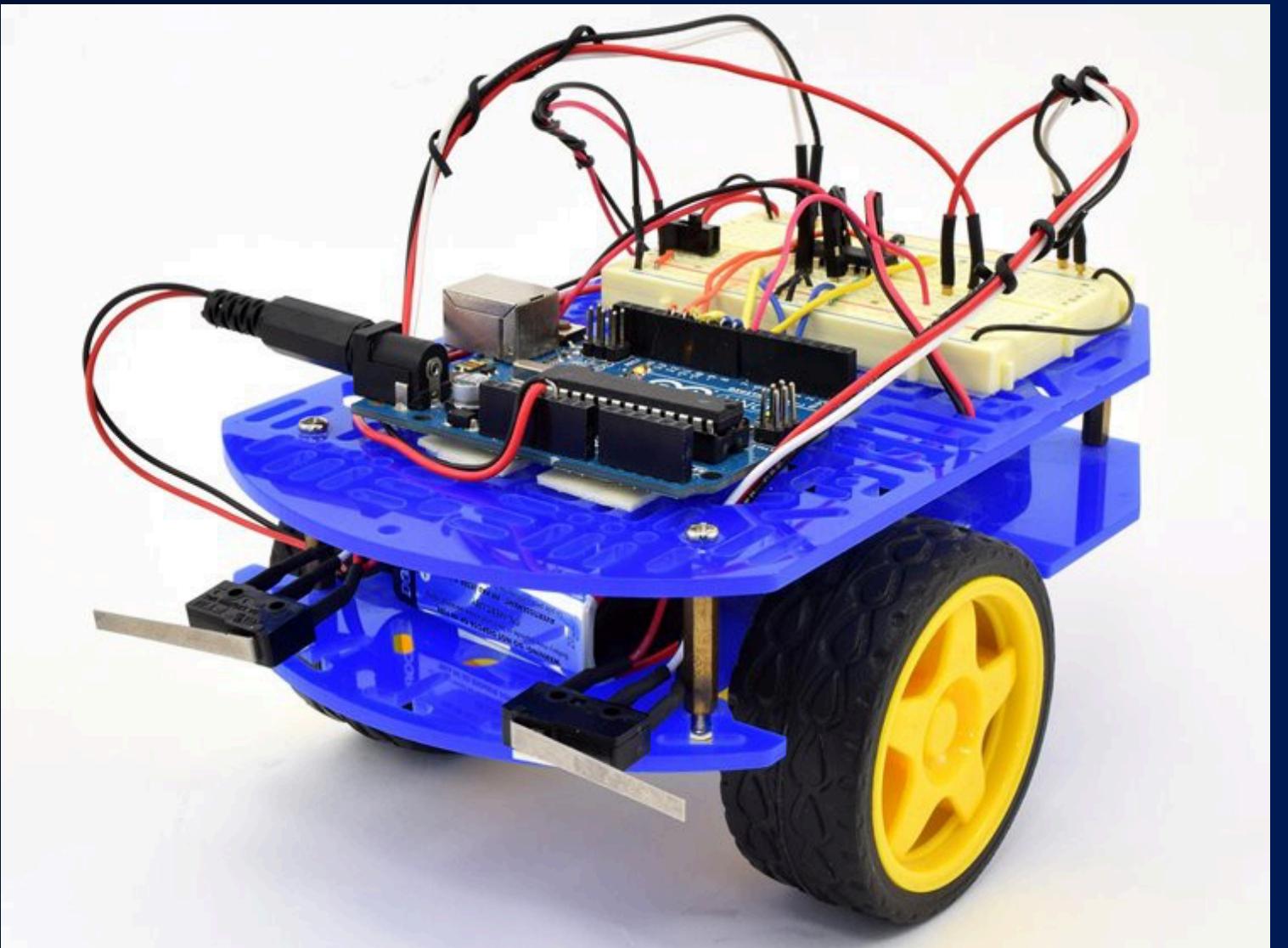
- Control remoto IR: Usar un control remoto para controlar dispositivos mediante infrarrojos.
 - Estación meteorológica: Medir temperatura y humedad con sensores y mostrar los resultados en una pantalla LCD.



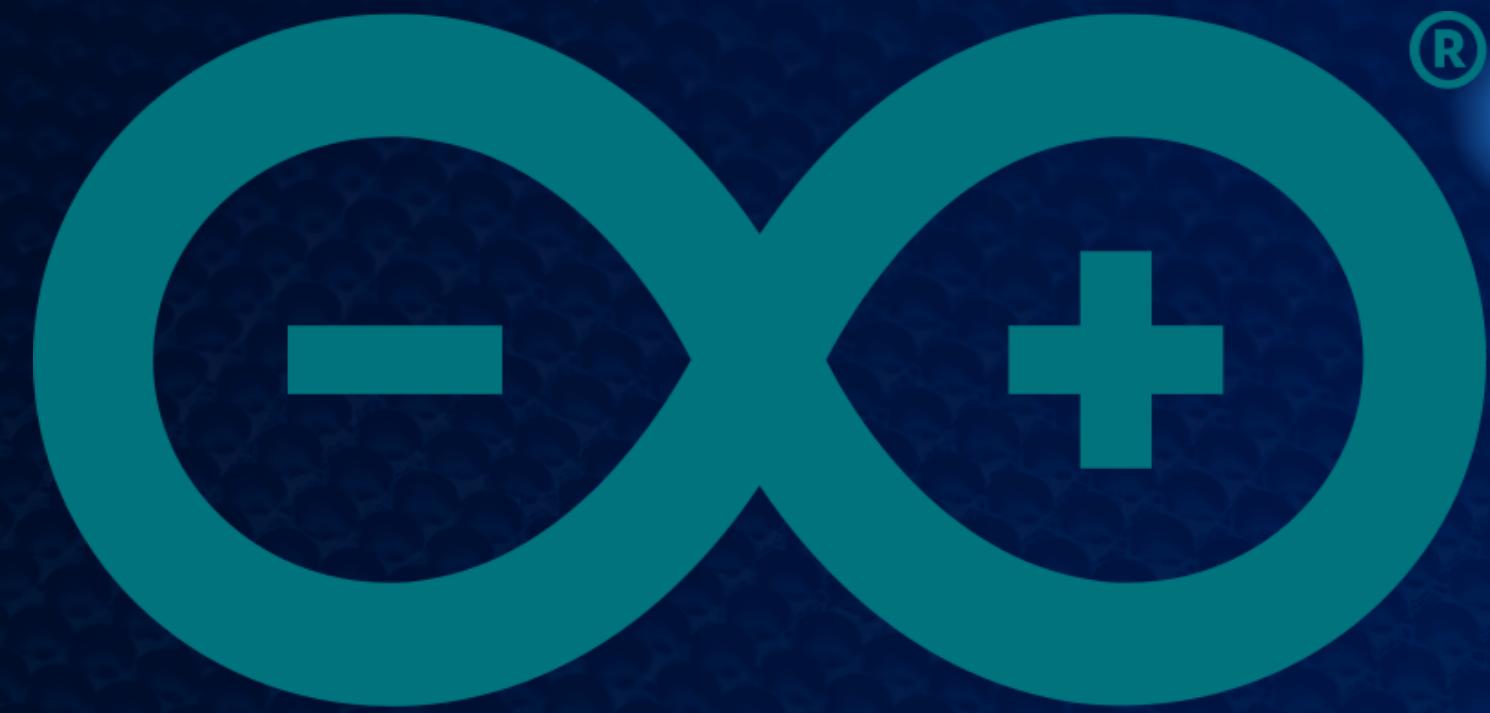
5. Proyectos comunes con Arduino

Proyectos avanzados:

- Robots autónomos: Integrar sensores de distancia para la navegación sin intervención humana.
- Sistema de automatización del hogar: Controlar luces y dispositivos a través de la web o por voz usando módulos WiFi.



Gracias



ARDUINO