

Taller Arduino: la placa mágica



Las tres incógnitas de Arduino

- ¿Qué es Arduino?
- ¿Por qué?
- ¿Cómo funciona?

¿Que es Arduino?

- **Arduino** es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos, automatización, pequeños proyectos domésticos o robótica.

¿Por qué Arduino?

1. Coste.

Precio de mercado muy bajo.

2. Disponibilidad.

Ebay, Amazon, DealeXtreme etc..

3. Flexibilidad.

Versiones desde muy sencillas a complejas.

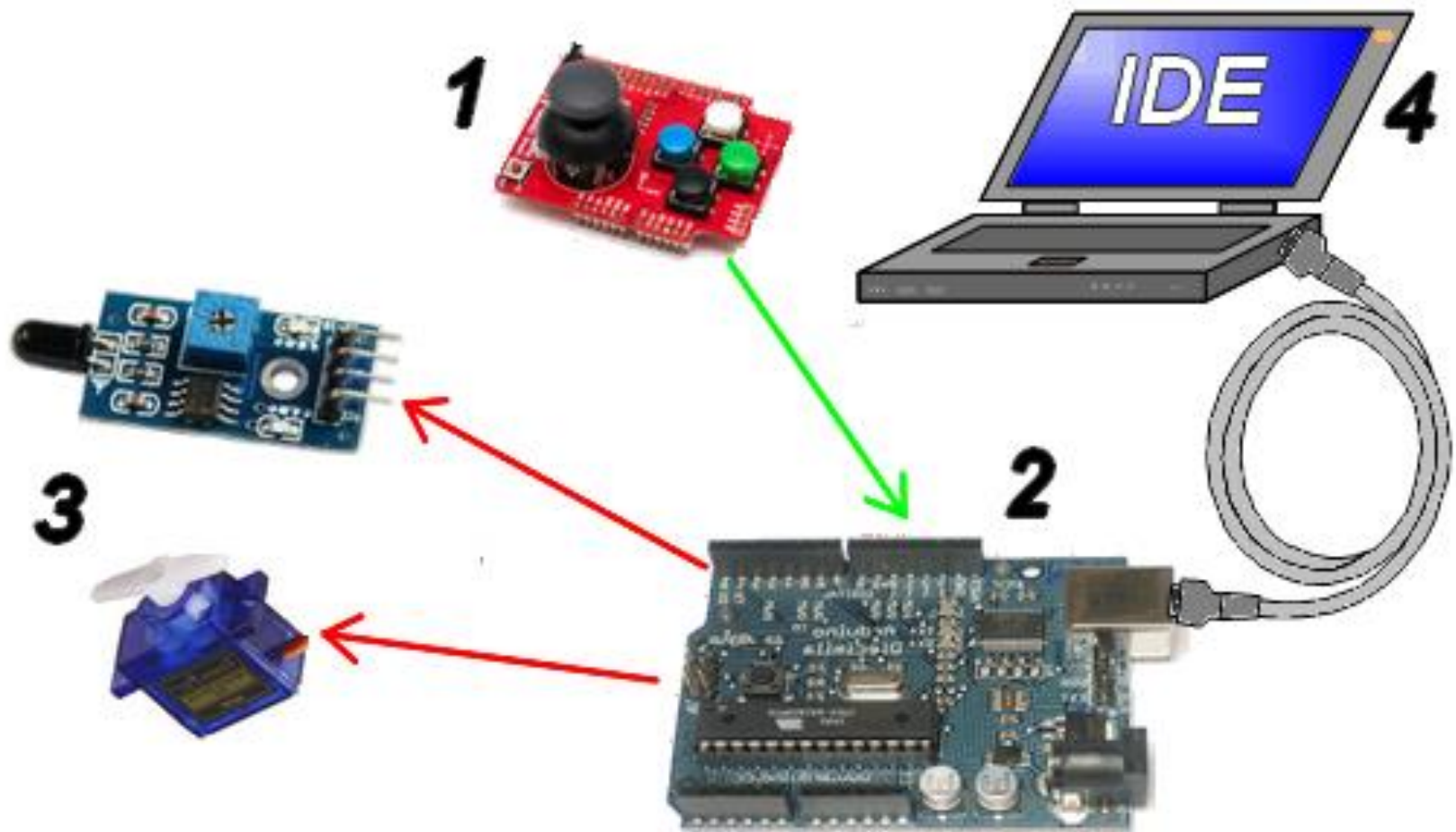
4. Infomación.

Documentación en la web a nivel mundial

¿Cómo Funciona?

- Como pasa con la mayoría de las placas microcontroladoras las funciones de Arduino pueden resumirse en tres. En primera instancia, tenemos una **interfaz de entrada**, seguido del **microprocesador** (Atmel en su caso) y los **periféricos de salida**. Todo ello con su lenguaje de programación basado en C++ compilado desde el IDE propio de Arduino.

Esquema



Versiões de Arduinos



Arduino UNO



Arduino Yún



Arduino Due



Arduino Esplora



Arduino Lilypad
USB



Arduino Micro



Arduino Ethernet



Arduino Mini



Arduino Nano



Arduino Robot



Arduino
Leonardo



Arduino Mega
2560



Arduino LilyPad
SimpleSnap



Arduino Mega
ADK



Arduino Fio



Arduino UNO
SMD



Arduino Zero

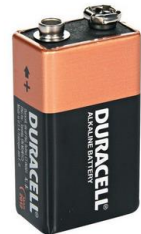
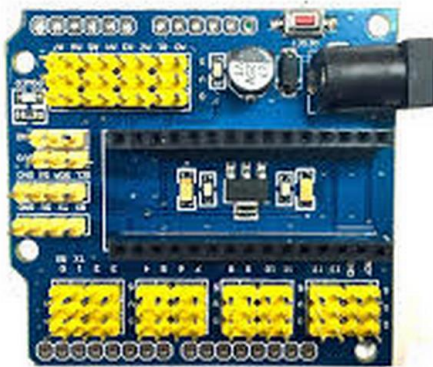
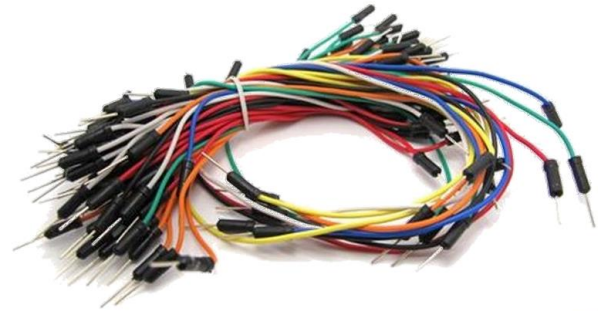
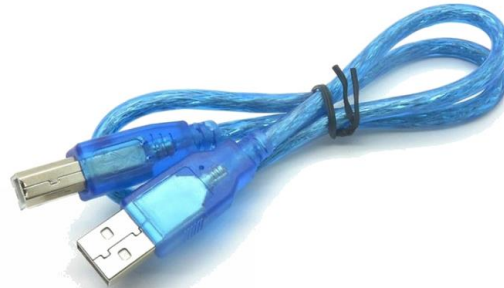
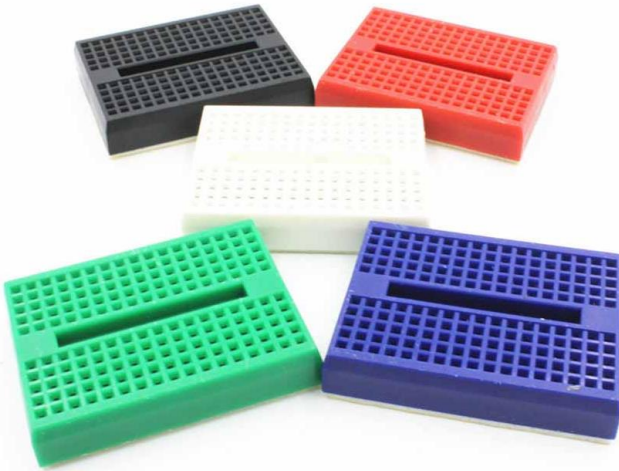


Arduino Yún Mini

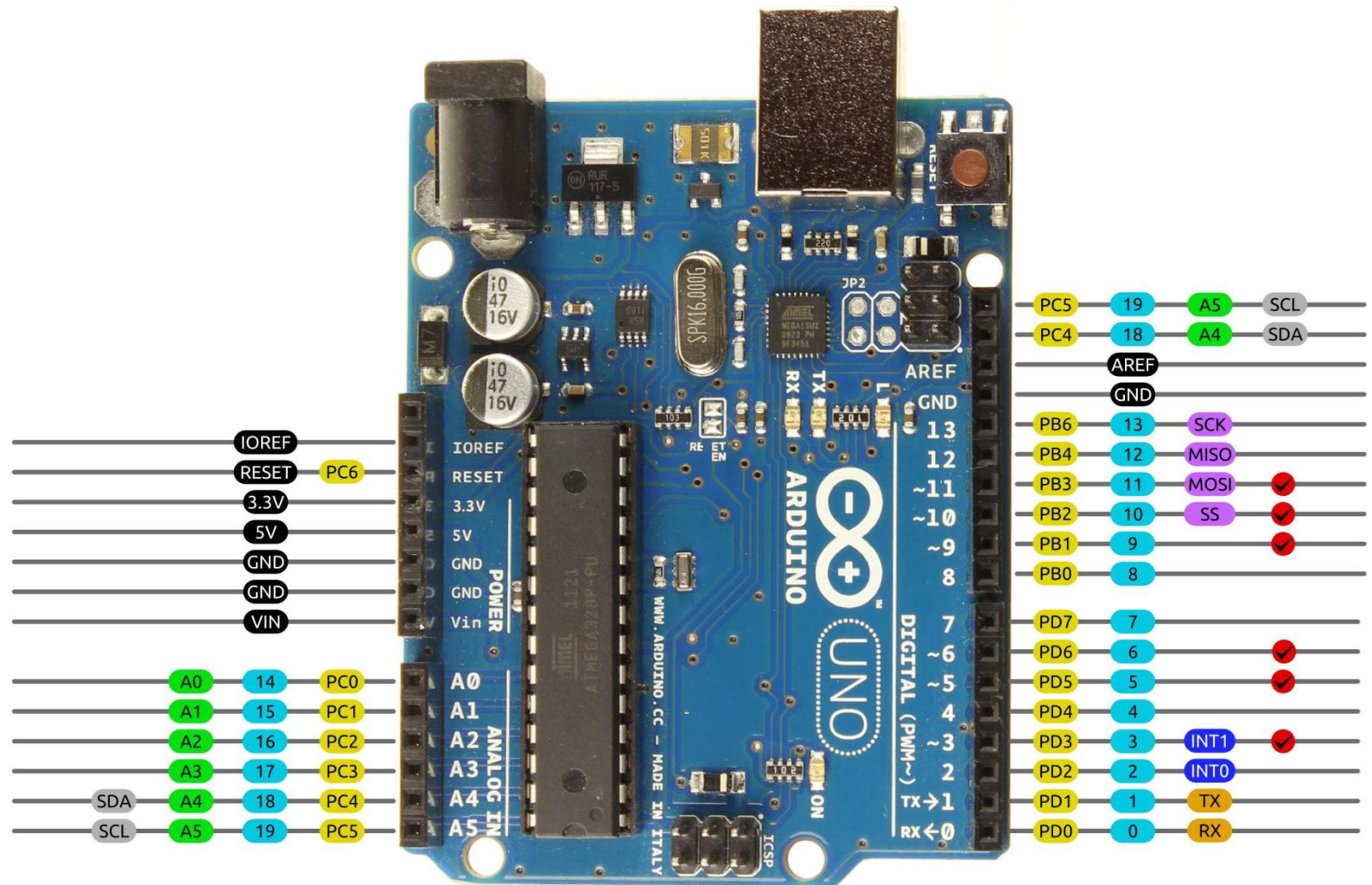
Periféricos



Accesorios

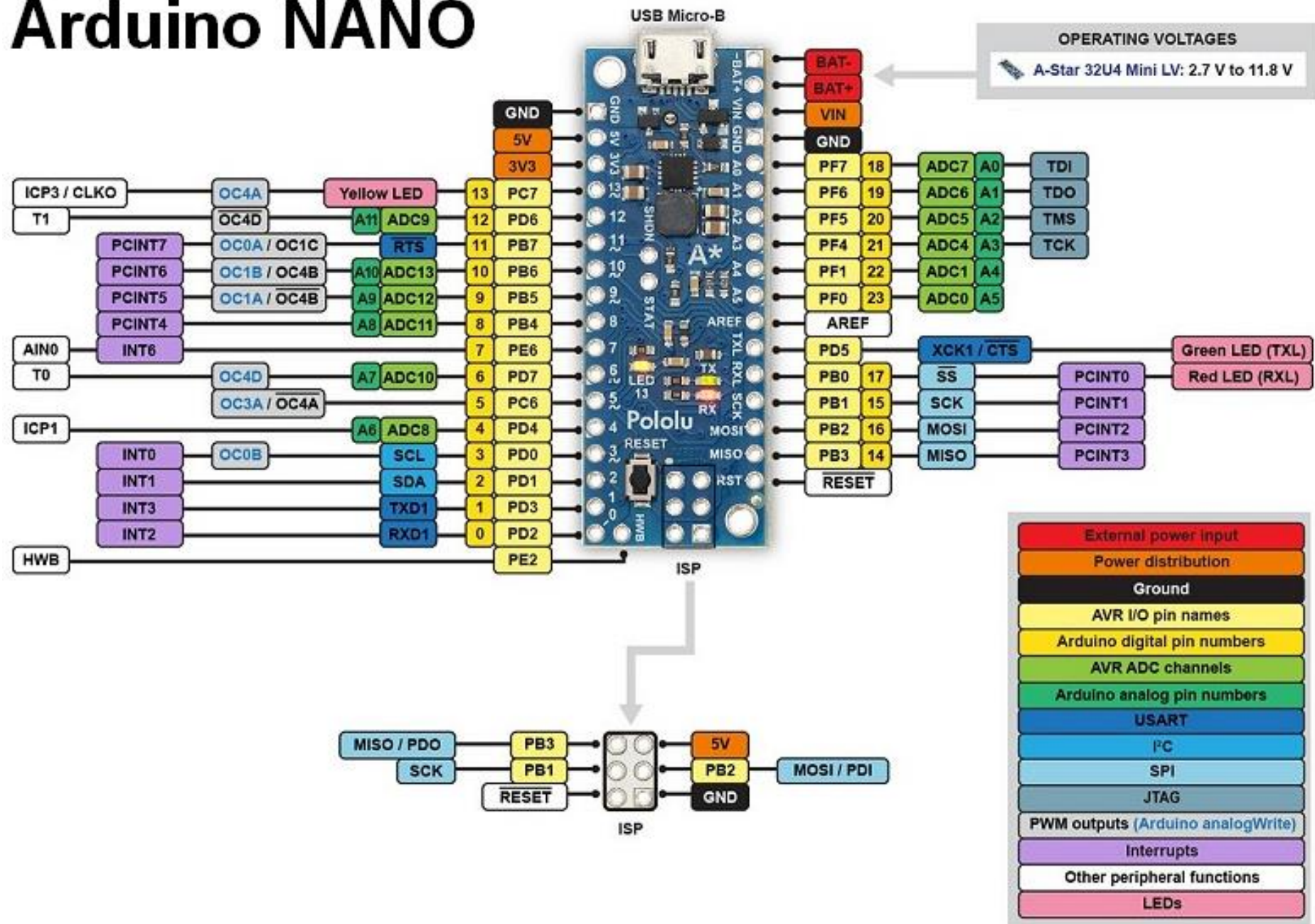


Arduino Uno R3 Pinout



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

Arduino NANO



Entorno de programación

IDE Arduino

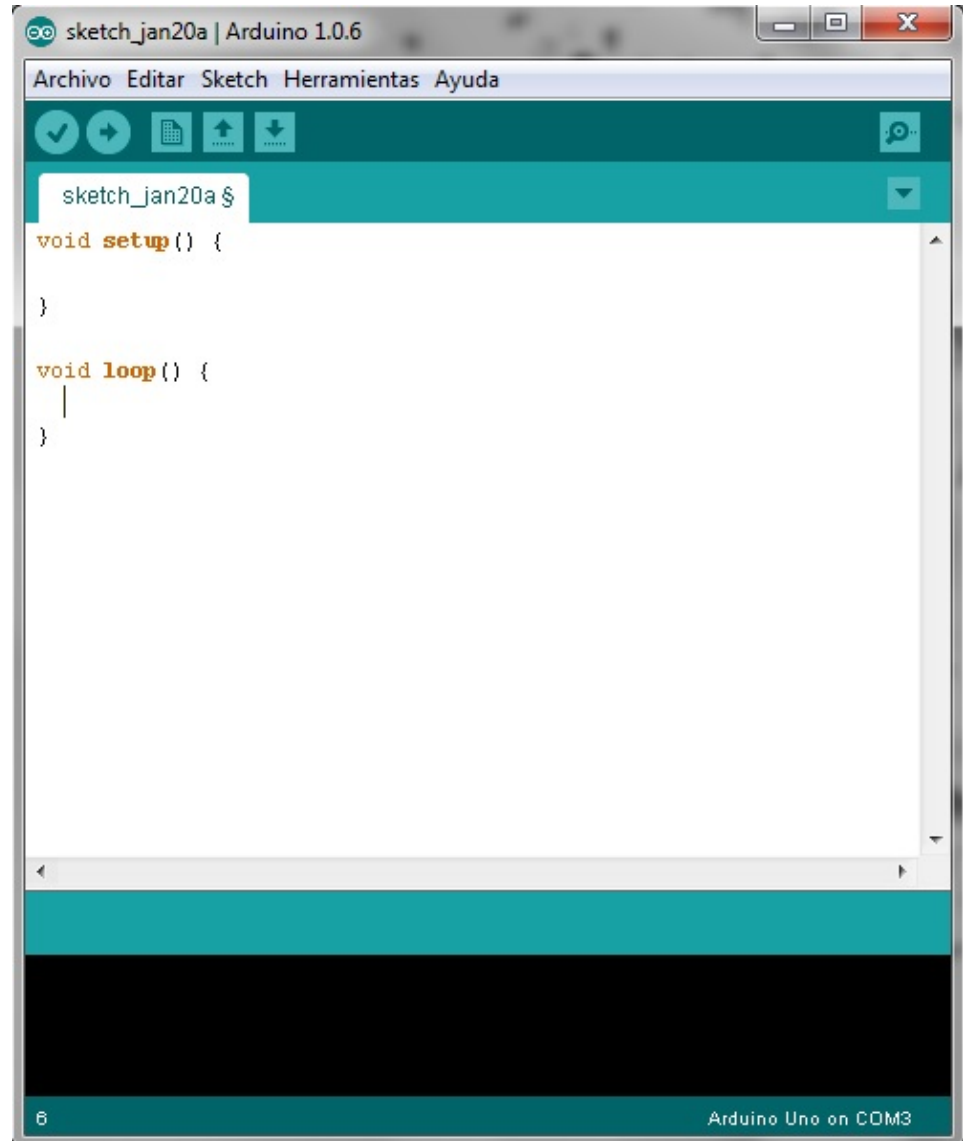
sketch = programa

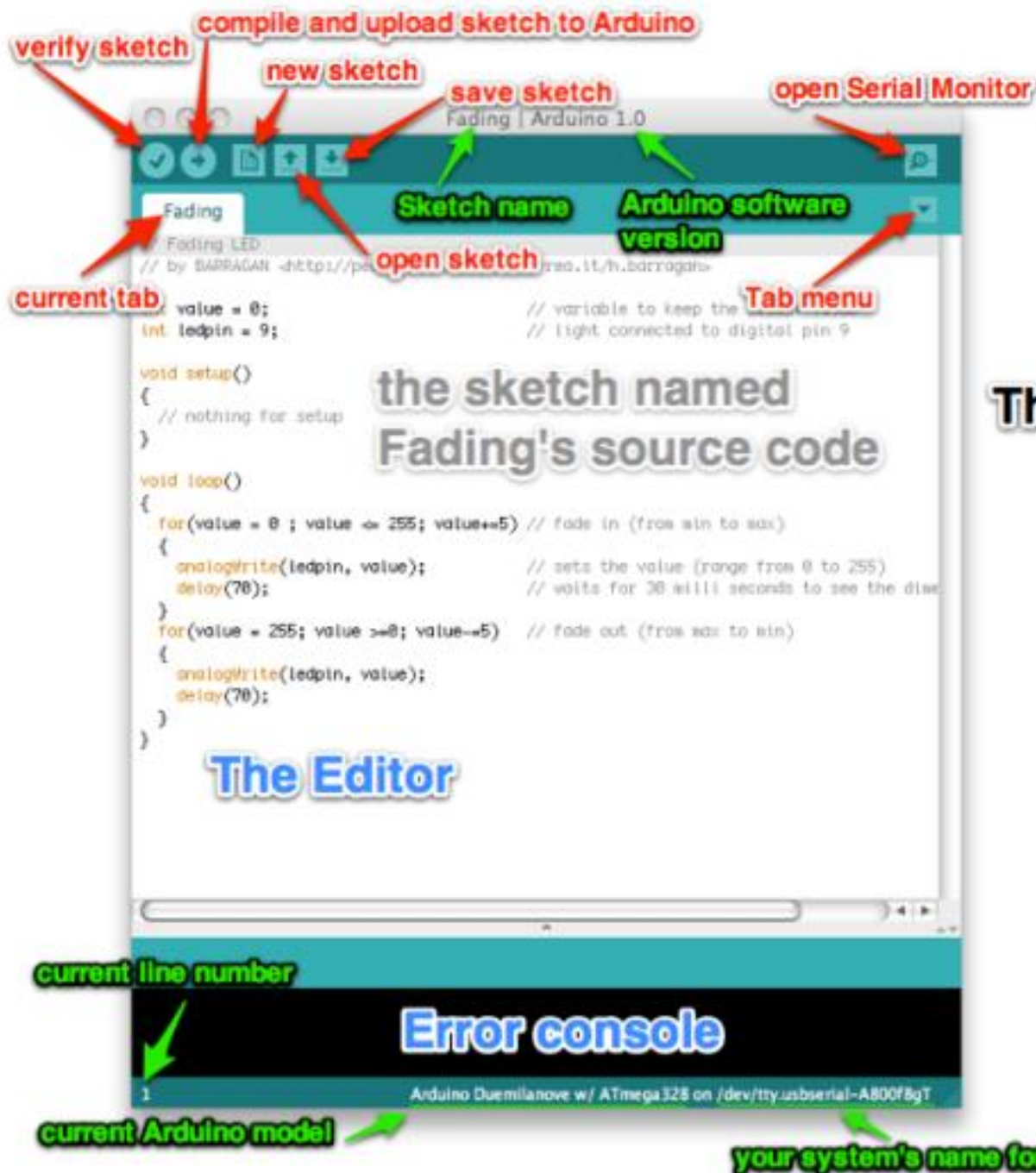
compile = verificar

upload = cargar programa
en la placa

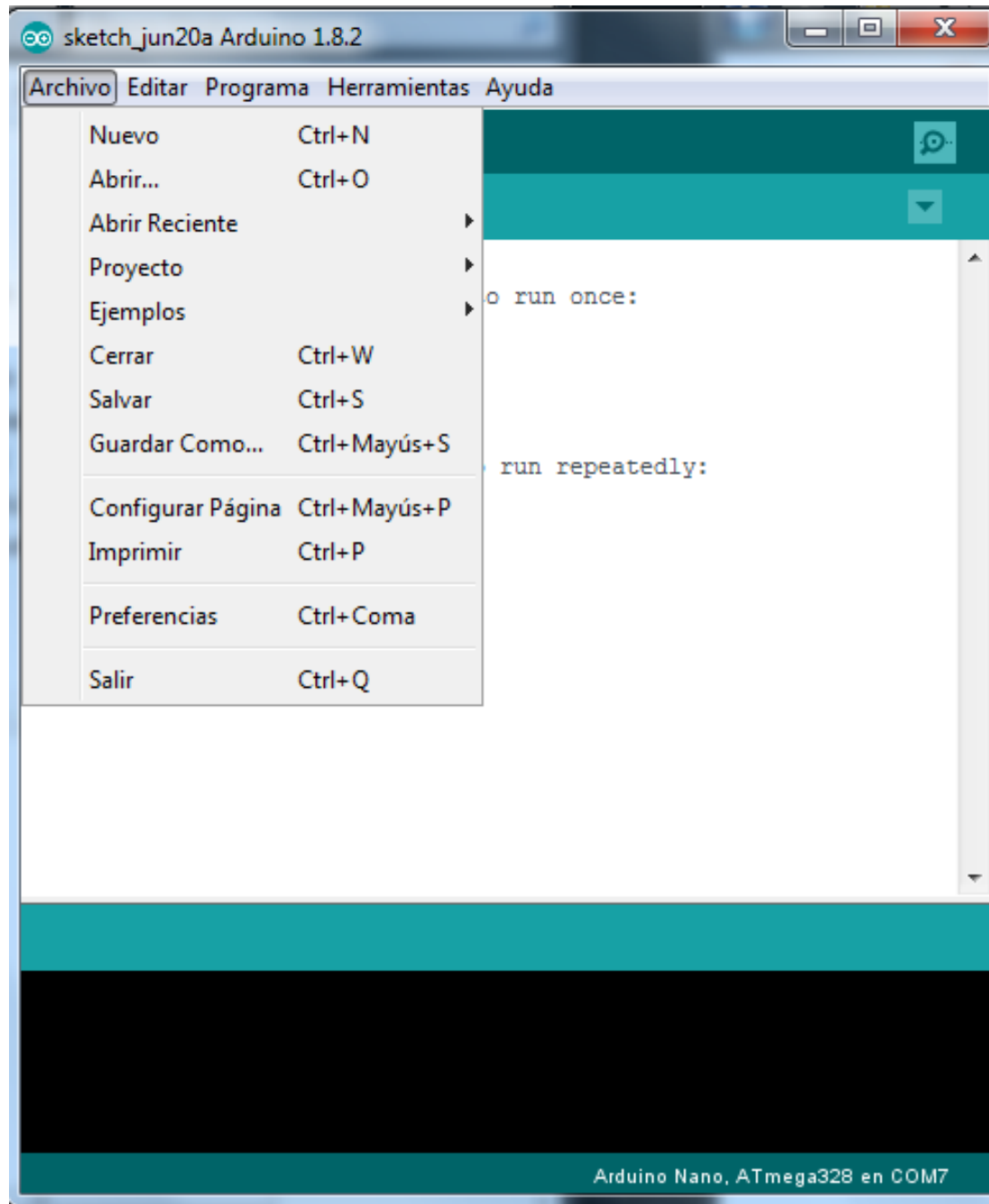
pin = zócalo conexión

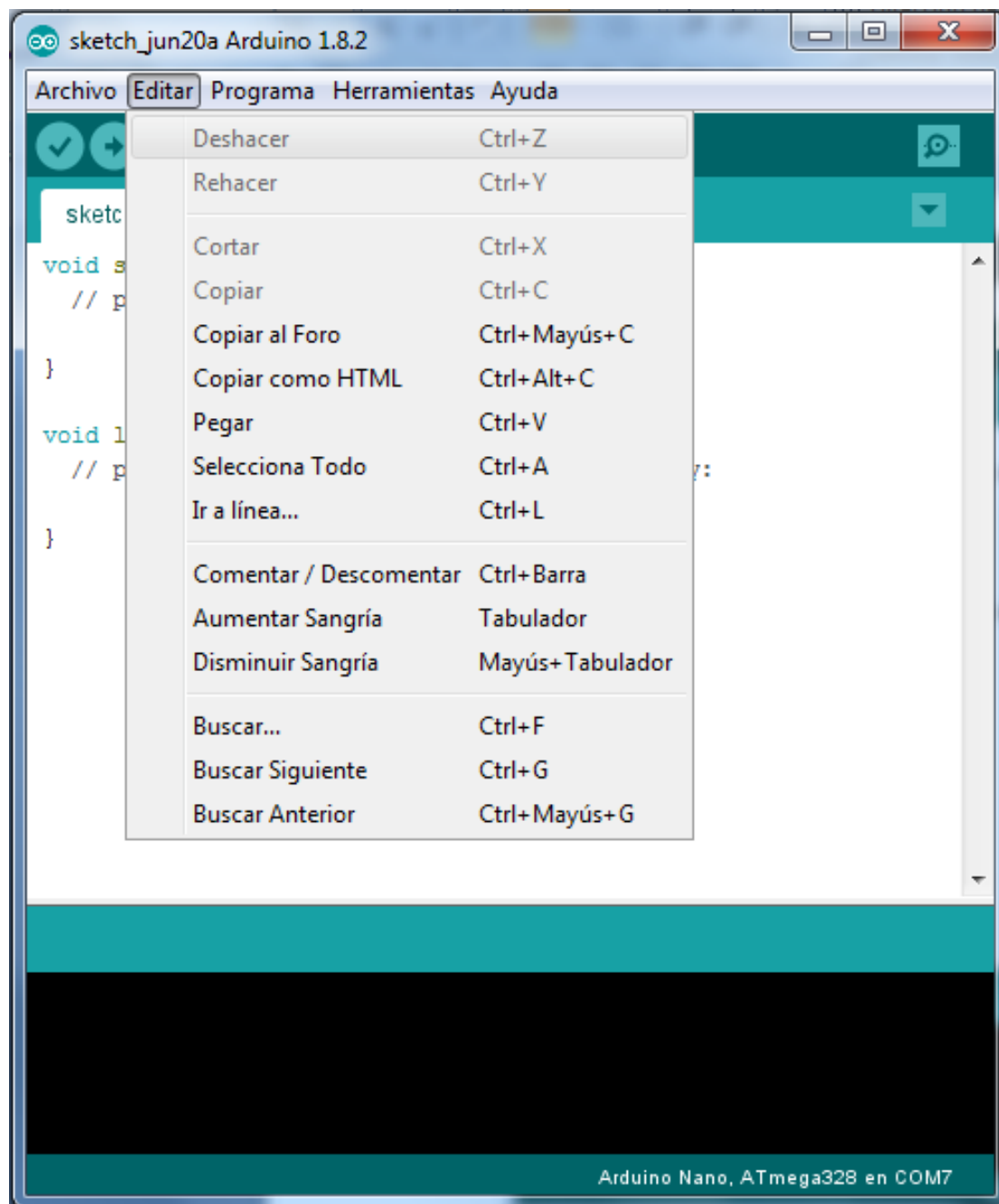
serial = visualizar en PC

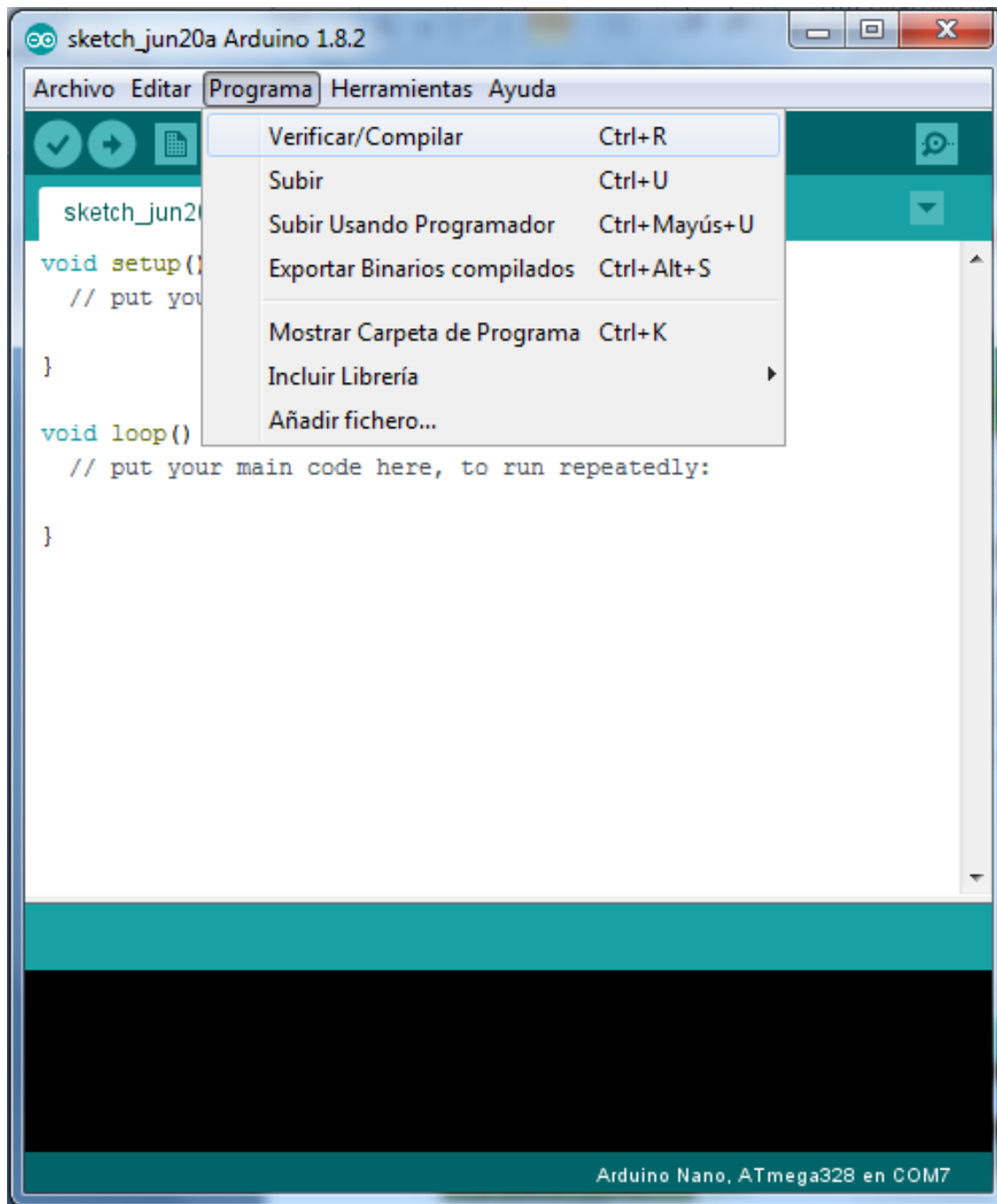


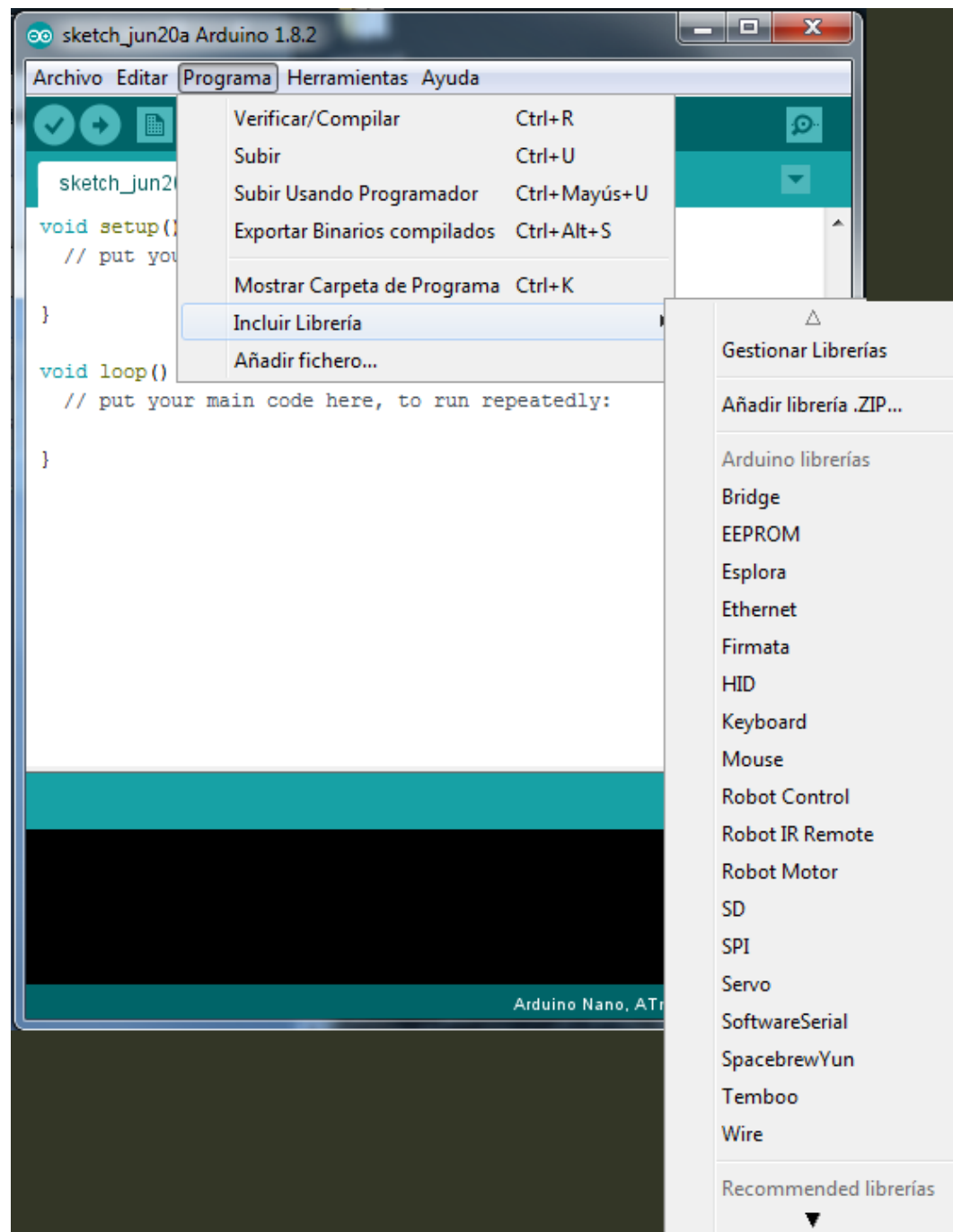


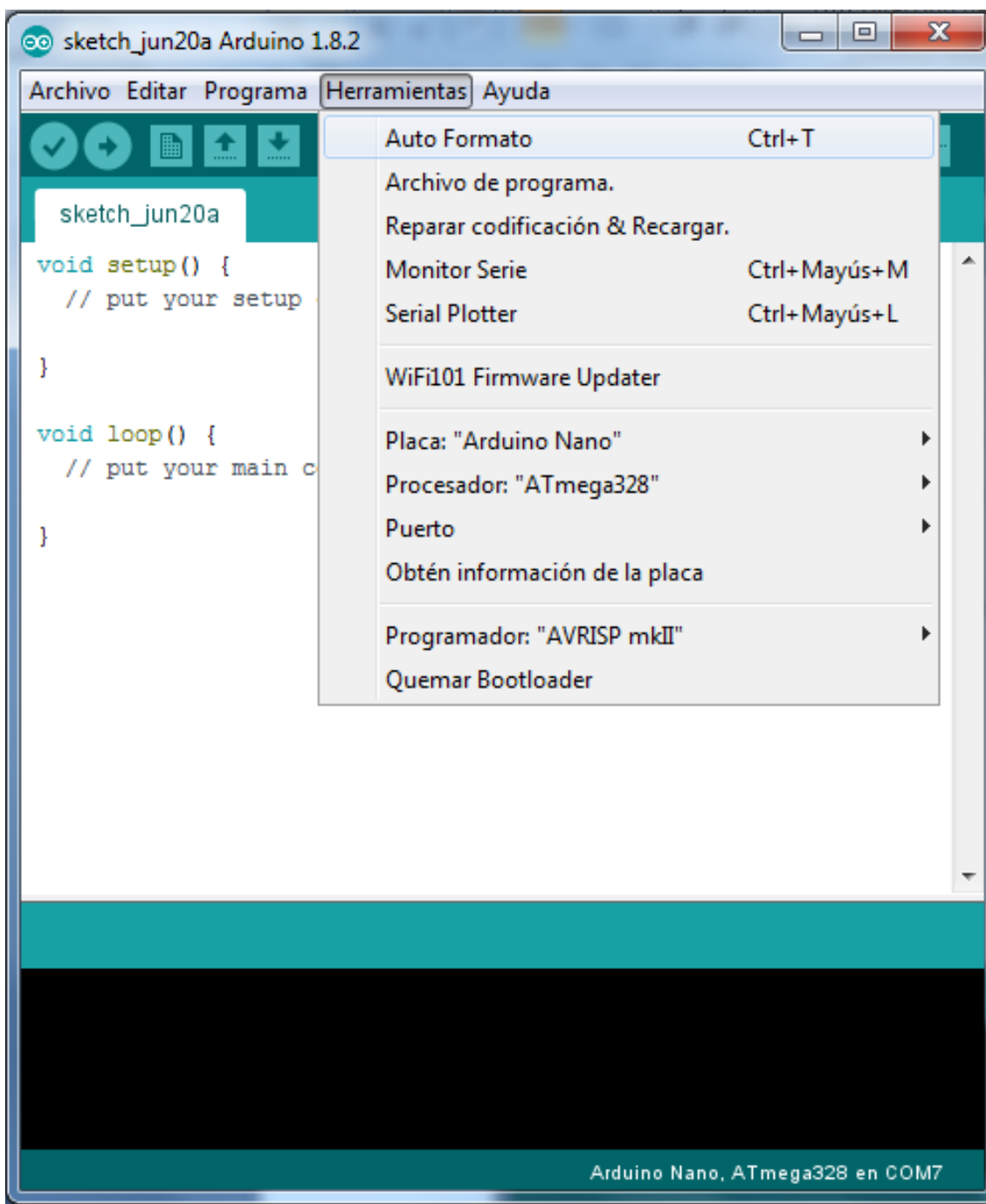
The Arduino IDE

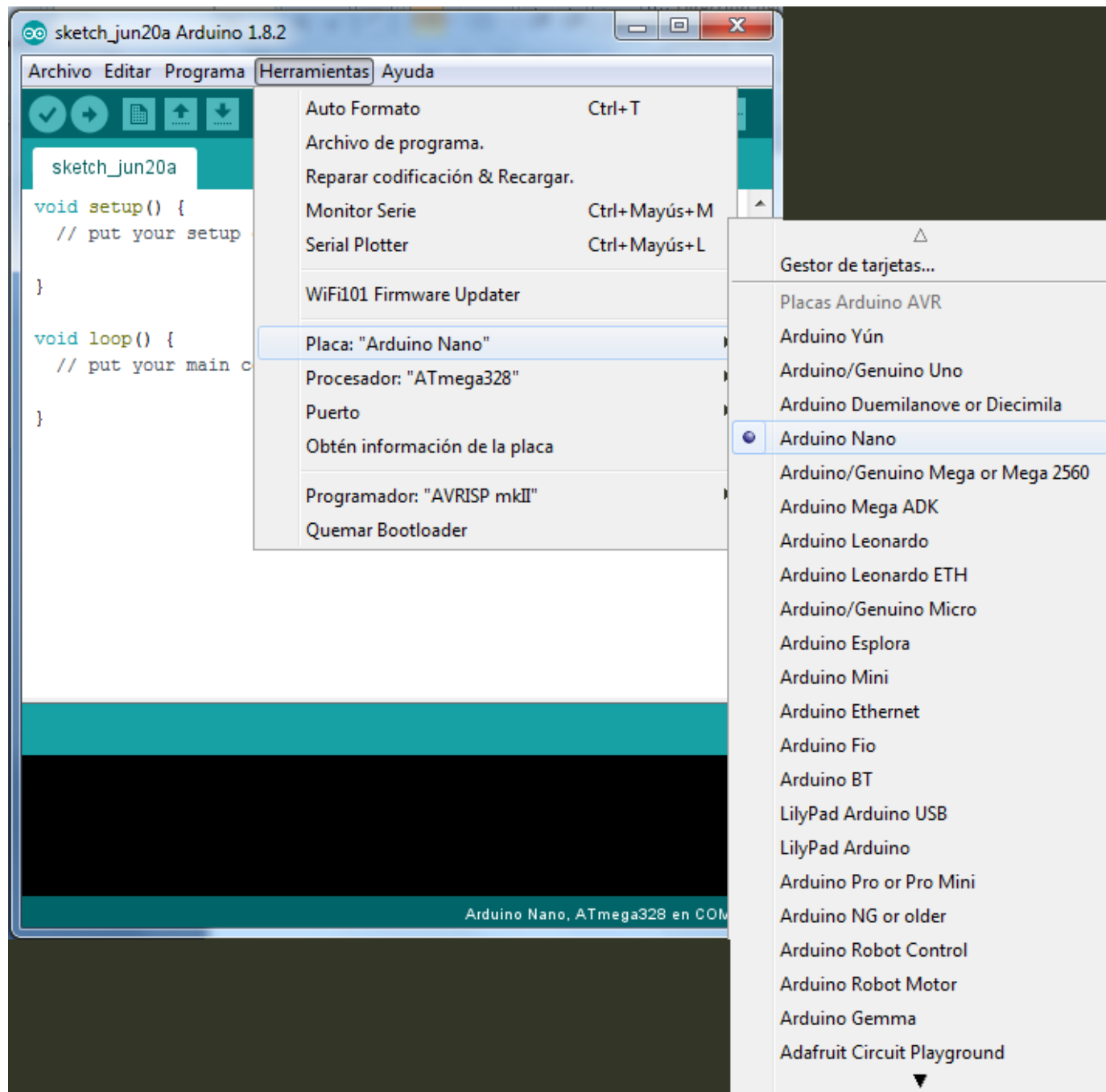


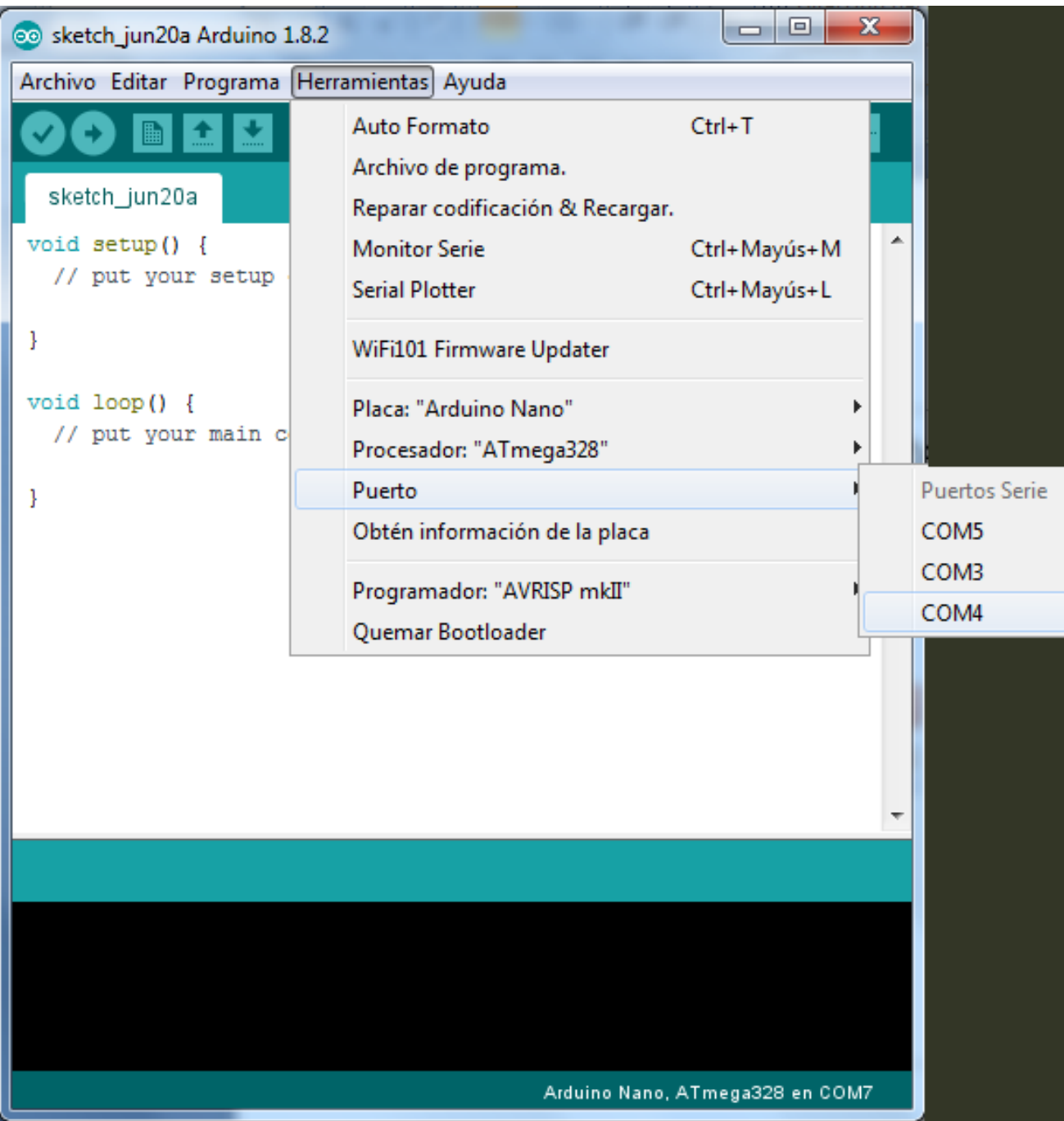


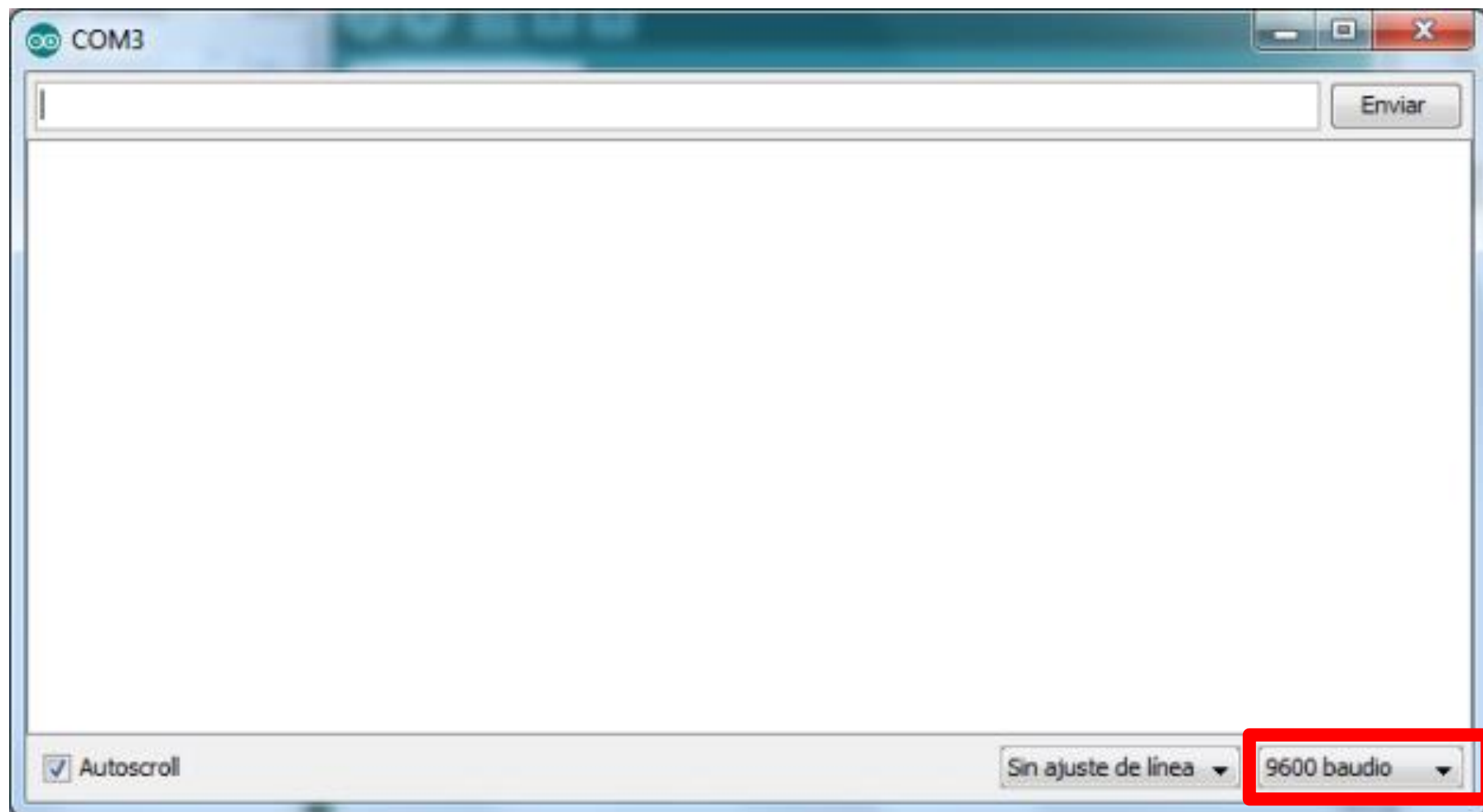


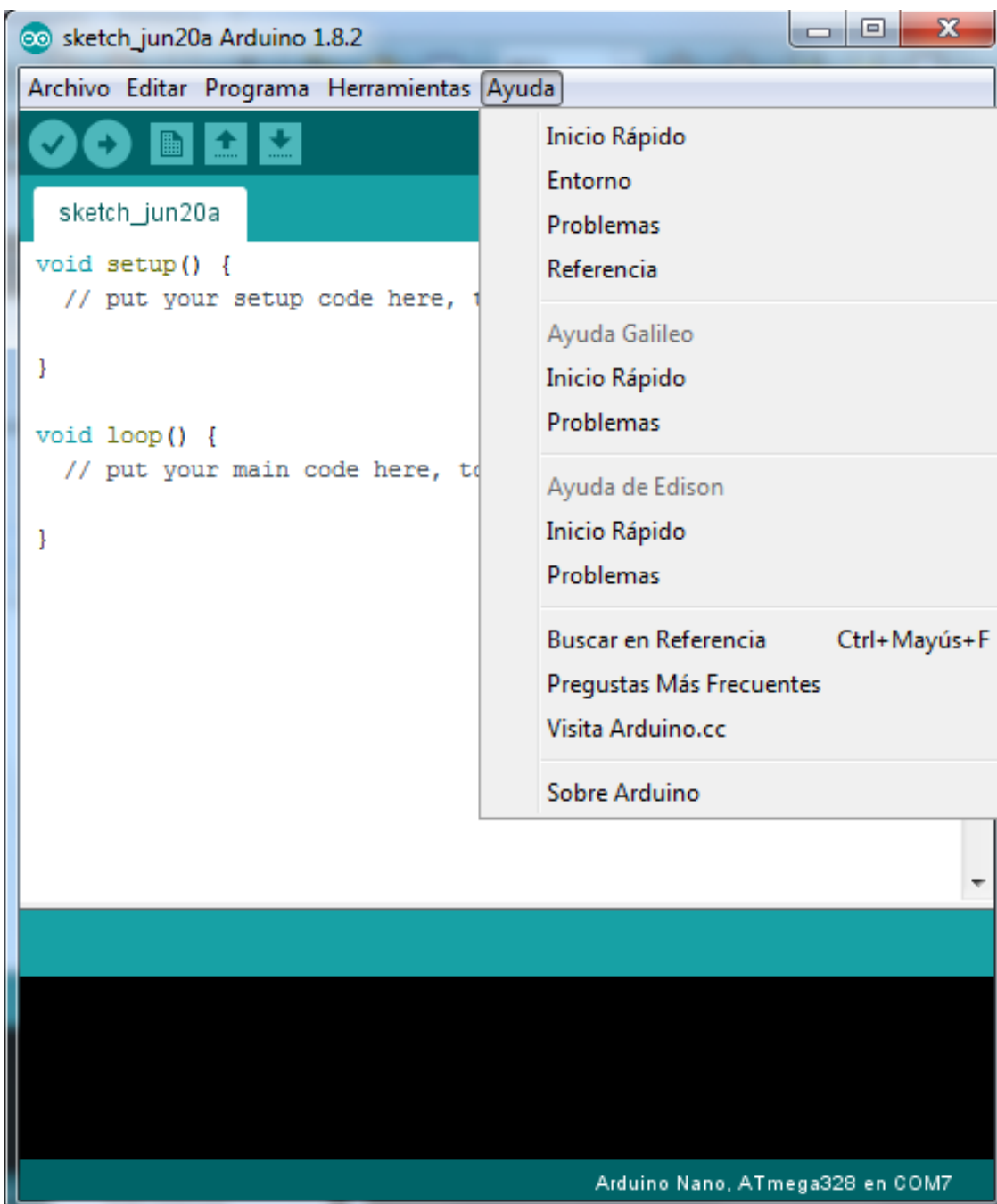




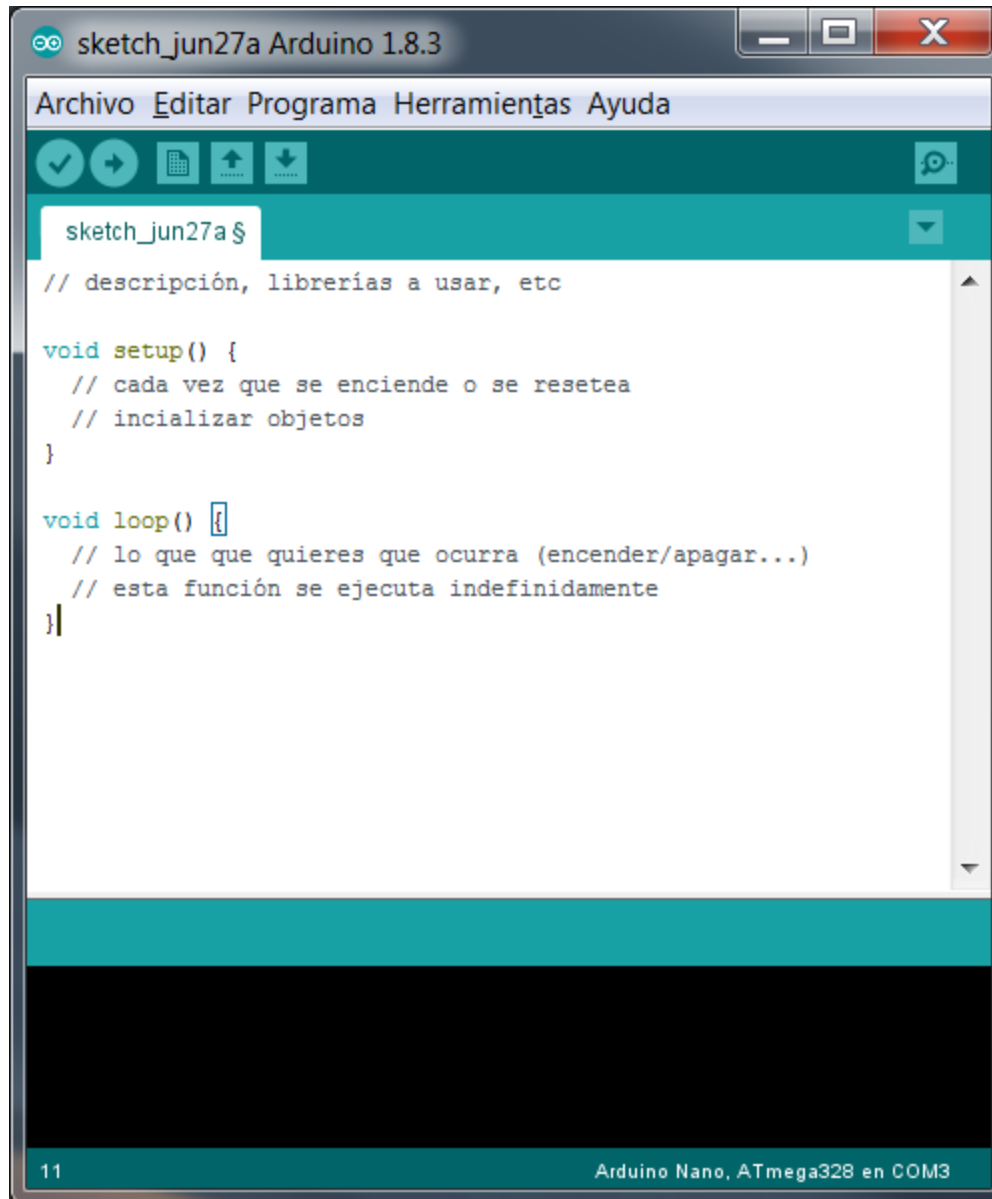








Estructura



The image shows the Arduino IDE interface for a sketch named 'sketch_jun27a'. The window title is 'sketch_jun27a Arduino 1.8.3'. The menu bar includes 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. The toolbar contains icons for checking, running, uploading, downloading, and a help icon. The sketch name 'sketch_jun27a\$' is displayed in the top bar. The main text area contains the following code:

```
// descripción, librerías a usar, etc

void setup() {
  // cada vez que se enciende o se resetea
  // inicializar objetos
}

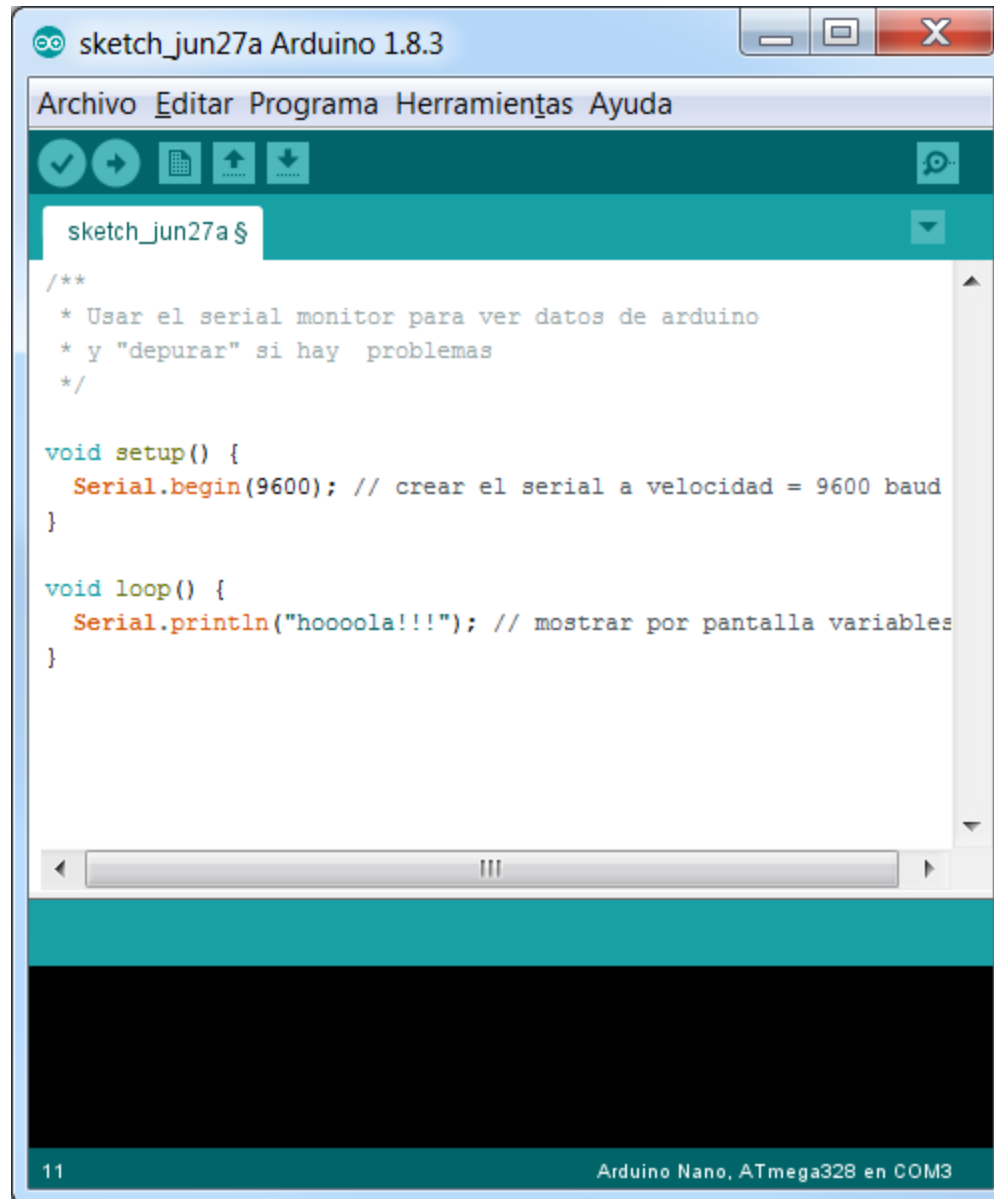
void loop() {
  // lo que que quieres que ocurra (encender/apagar...)
  // esta función se ejecuta indefinidamente
}
```

The status bar at the bottom shows '11' on the left and 'Arduino Nano, ATmega328 en COM3' on the right.

Uso básico



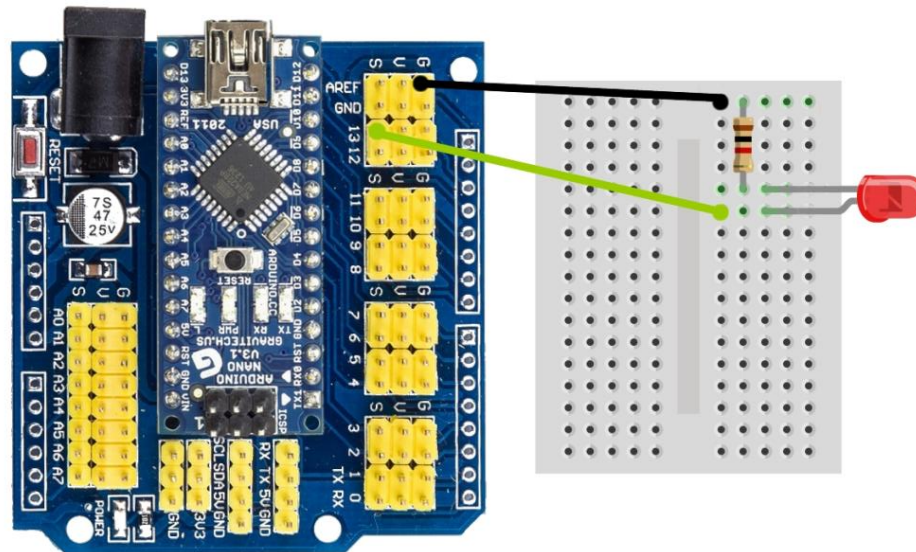
Serial monitor



Blink

Nuestro primer programa será hacer parpadear el led (bombillita) integrado que lleva Arduino u otro led conectado a un pin (13) digital a través de una resistencia.

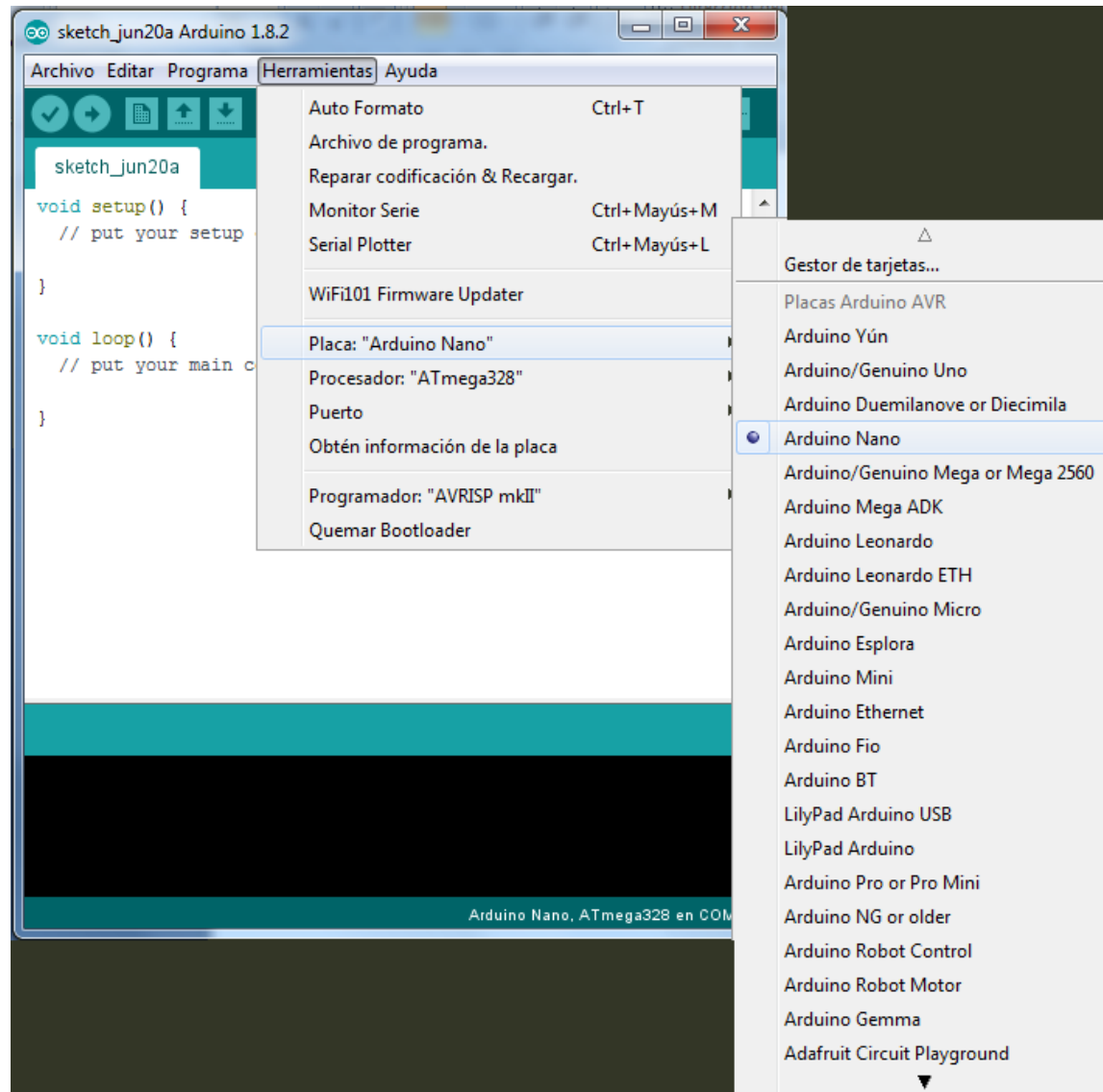
Este es el esquema a usar:



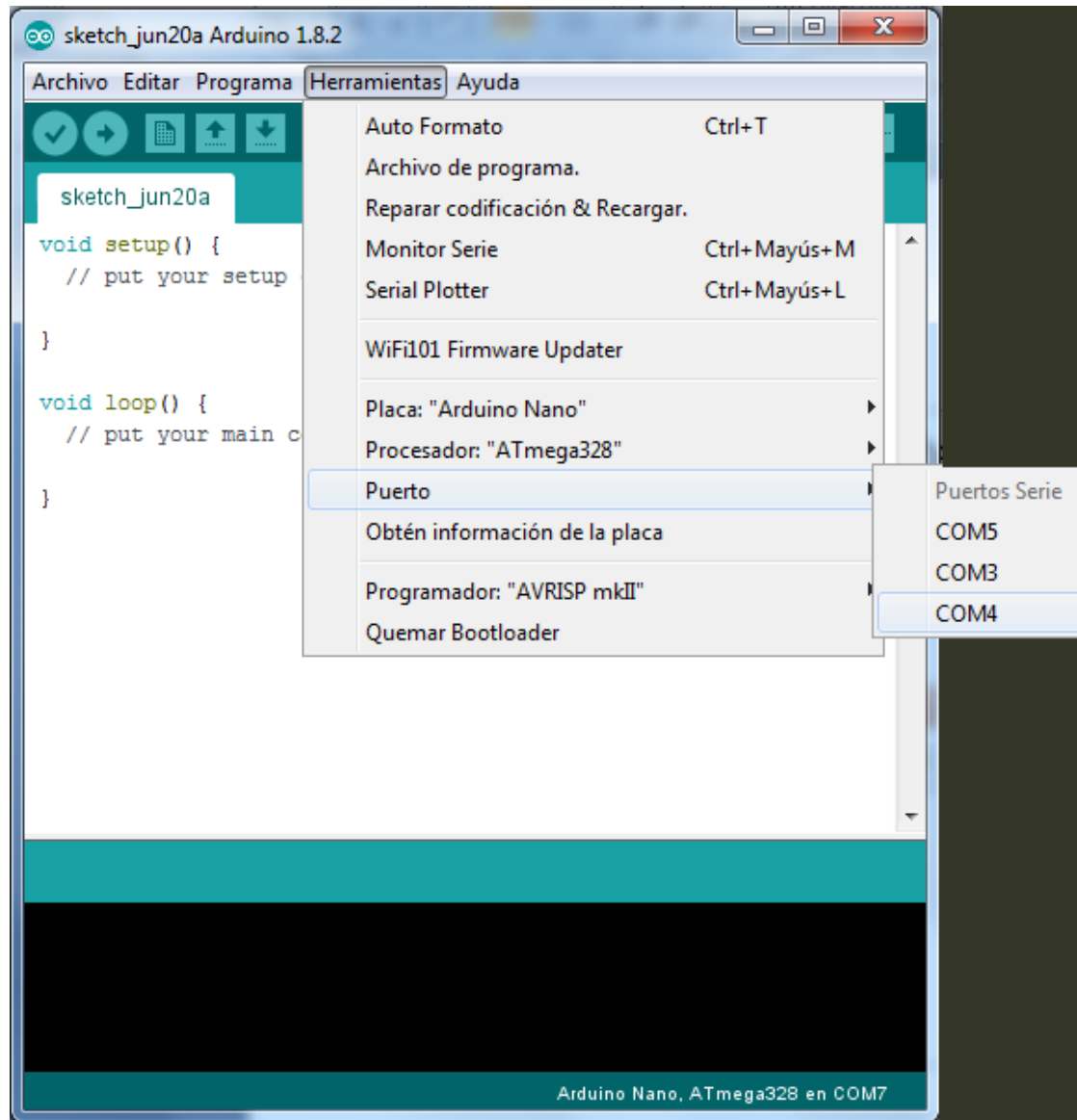
Pasos a seguir:

- Abrir la aplicación Arduino
- Seleccionar Placa a utilizar
- Seleccionar Puerto usado por la placa
- Escribir programa (Código)
- Subir programa

Seleccionar la placa



Seleccionar Puerto



El Código

Estructura de un programa:

void setup() //Primera Parte

{

// inicializar objetos

}

void loop() //Segunda Parte

{

// ejecutar programa indefinidamente

}

setup()

La función **setup()** se invoca una sola vez cuando el programa empieza. Se utiliza para inicializar los modos de trabajo de los pins, o el puerto serie. Debe ser incluido en un programa aunque no haya declaración que ejecutar. Así mismo se puede utilizar para establecer el estado inicial de las salidas de la placa.

```
void setup() {  
    pinMode(13, OUTPUT); // configura el 'pin' como salida  
    digitalWrite(13, HIGH); // pone el 'pin' en estado HIGH  
}
```

loop()

Después de llamar a **setup()**, la función **loop()** hace precisamente lo que sugiere su nombre, se ejecuta de forma cíclica, lo que posibilita que el programa esté respondiendo continuamente ante los eventos que se produzcan en la placa.

```
void loop() {  
    digitalWrite(13, HIGH); // pone en uno (on, 5v) el 'pin'  
    delay(1000); // espera un segundo (1000 ms)  
    digitalWrite(13, LOW); // pone en cero (off, 0v.) el 'pin'  
    delay(1000); // espera un segundo (1000 ms)  
}
```

Subir el programa

- Cargar el programa pulsando el botón “subir”. El programa se compilará y luego se verá parpadeando los leds Tx y Rx de Arduino, indicando que se está cargando el fichero binario (.hex) en Arduino. Cuando aparezca el mensaje “subido” habremos acabado.
- Unos segundos después veremos el LED parpadeando.

Ejercicios

- **Blink:** encender y apagar leds
- **Pulsador digital:** enviar ordenes a arduino
- **Puls dig serial:** depurar arduino en el PC
- **LDR:** señales analógicas (luminosidad)
- **Servo:** mover objetos
- **Servo Pot:** regular el movimiento de objetos
- **Relé:** ON/OFF, separar electrónicamente (ej: 220 Volts)

Ejercicios (cont)

- **Relé LDR:** relé pero ON/OFF automático
- **Bluetooth:** controlar arduino con móvil
- **Infrarrojo:** mandos a distancia tele
- **RF_433:** comunicación inalámbrica económica
- **WI-FI server:** conectarse a una WI-FI existente
- **WI-FI server:** crear red WI-FI con arduino
- **WI-FI ADC:** Transformar datos analógicos a un mensaje digital y enviar por WI-FI

A partir de aquí.....



.....Imaginación