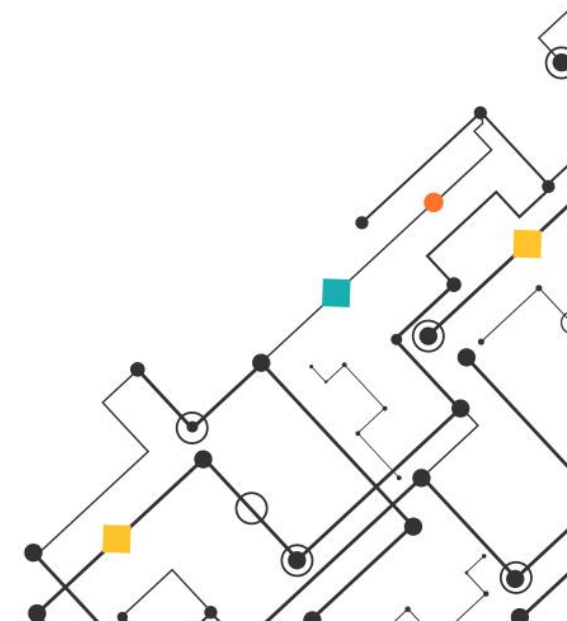




# The Tech Project

Arduino en el aula



2.4

# IDE: el entorno de programación



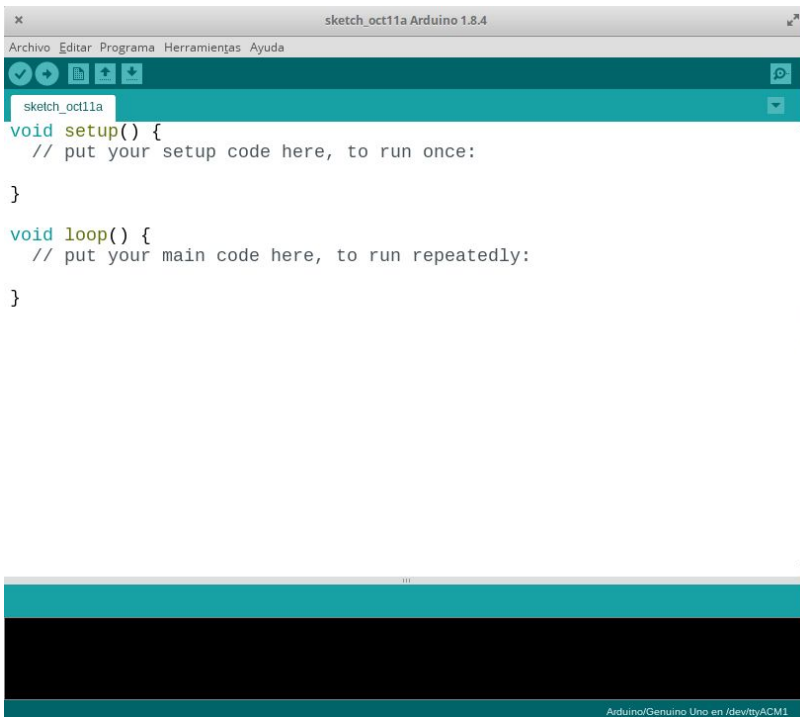
# IDE de Arduino

- ¿Qué es un IDE?
- ¿Cómo se instala?
- Características
- Configuración
- Compilación y errores
- Subida del programa a la placa y sus errores



# IDE de Arduino

Entorno de Desarrollo Integrado o conjunto de programas que nos permite trabajar con código



The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_oct11a Arduino 1.8.4". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar contains icons for opening, saving, and running. The main text area shows the following code:

```
sketch_oct11a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

The status bar at the bottom indicates "Arduino/Genuino Uno en /dev/ttyACM1".

## Contiene:

- Editor
- Compilador
- Depurador
- Instalador Librerías
- Documentación

## Multiplataforma:

- Linux
- Windows
- OsX
- Es Open Source

Usaremos lenguaje C++

IDE de Arduino es Software Libre 100%



# IDE de Arduino: Instalación

Podemos descargarlo de este enlace y elegir versión según nuestro sistema operativo:

<https://www.arduino.cc/en/Main/Software>

## Download the Arduino IDE



### ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer

**Windows** ZIP file for non admin install

**Windows app** 

**Mac OS X** 10.7 Lion or newer

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

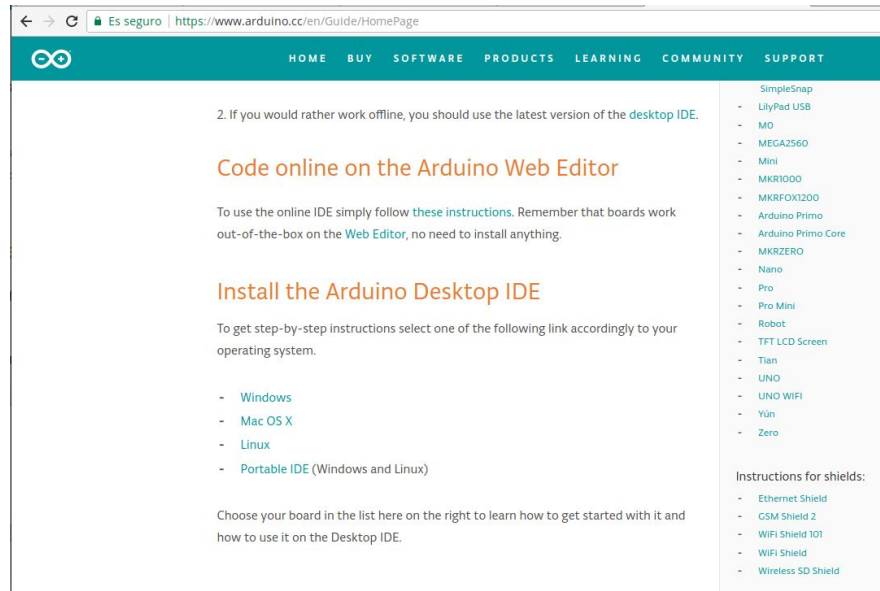


# IDE de Arduino: Instalación

Existen guías de instalación para las diferentes plataformas:

<https://www.arduino.cc/en/Guide/HomePage>

<http://arduino.cc/en/Guide/troubleshooting>



A screenshot of the Arduino website's 'Guide/HomePage'. The browser address bar shows 'https://www.arduino.cc/en/Guide/HomePage'. The navigation menu includes HOME, BUY, SOFTWARE, PRODUCTS, LEARNING, COMMUNITY, and SUPPORT. The main content area features a heading 'Code online on the Arduino Web Editor' and a sub-heading 'Install the Arduino Desktop IDE'. A list of boards is provided on the right side, including SimpleSnap, LilyPad USB, MO, MEGA2560, Mini, MKR1000, MKRFOX1200, Arduino Primo, Arduino Primo Core, MKRZERO, Nano, Pro, Pro Mini, Robot, TFT LCD Screen, Tian, UNO, UNO WIFI, Yún, and Zero. Below this list, there is a section for 'Instructions for shields:' with links to Ethernet Shield, GSM Shield 2, WIFI Shield 101, WIFI Shield, and Wireless SD Shield.

2. If you would rather work offline, you should use the latest version of the desktop IDE.

### Code online on the Arduino Web Editor

To use the online IDE simply follow these instructions. Remember that boards work out-of-the-box on the Web Editor, no need to install anything.

### Install the Arduino Desktop IDE

To get step-by-step instructions select one of the following link accordingly to your operating system.

- Windows
- Mac OS X
- Linux
- Portable IDE (Windows and Linux)

Choose your board in the list here on the right to learn how to get started with it and how to use it on the Desktop IDE.

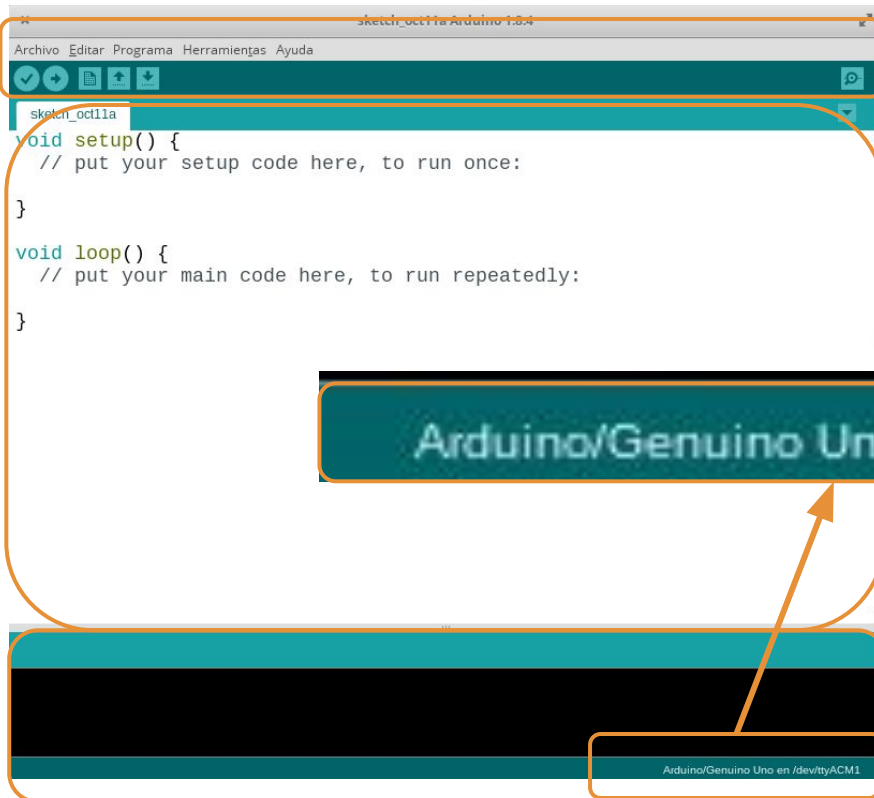
- SimpleSnap
- LilyPad USB
- MO
- MEGA2560
- Mini
- MKR1000
- MKRFOX1200
- Arduino Primo
- Arduino Primo Core
- MKRZERO
- Nano
- Pro
- Pro Mini
- Robot
- TFT LCD Screen
- Tian
- UNO
- UNO WIFI
- Yún
- Zero

Instructions for shields:

- Ethernet Shield
- GSM Shield 2
- WIFI Shield 101
- WIFI Shield
- Wireless SD Shield



# IDE de Arduino: Características



**Menú y Botonera**, para acceder a las principales herramientas

**Editor de código**, donde trabajaremos la mayor parte del tiempo

Arduino/Genuino Uno en /dev/ttyACM1

**Conexión y placa**, con información sobre la placa, el modelo de Arduino y dónde está conectado

**Depuración**, errores



## Resumen Botonera

**Subir el programa** a la placa que tenemos conectada

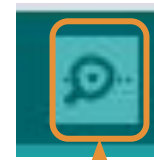
**Abrir un proyecto,** para proyectos guardados previamente



**Verificar y compilar el código,** primer paso para comprobar que nuestro código es correcto y generar el ejecutable

**Nuevo proyecto,** nos permite generar un proyecto desde cero

**Guardar el proyecto** con el que estamos trabajando



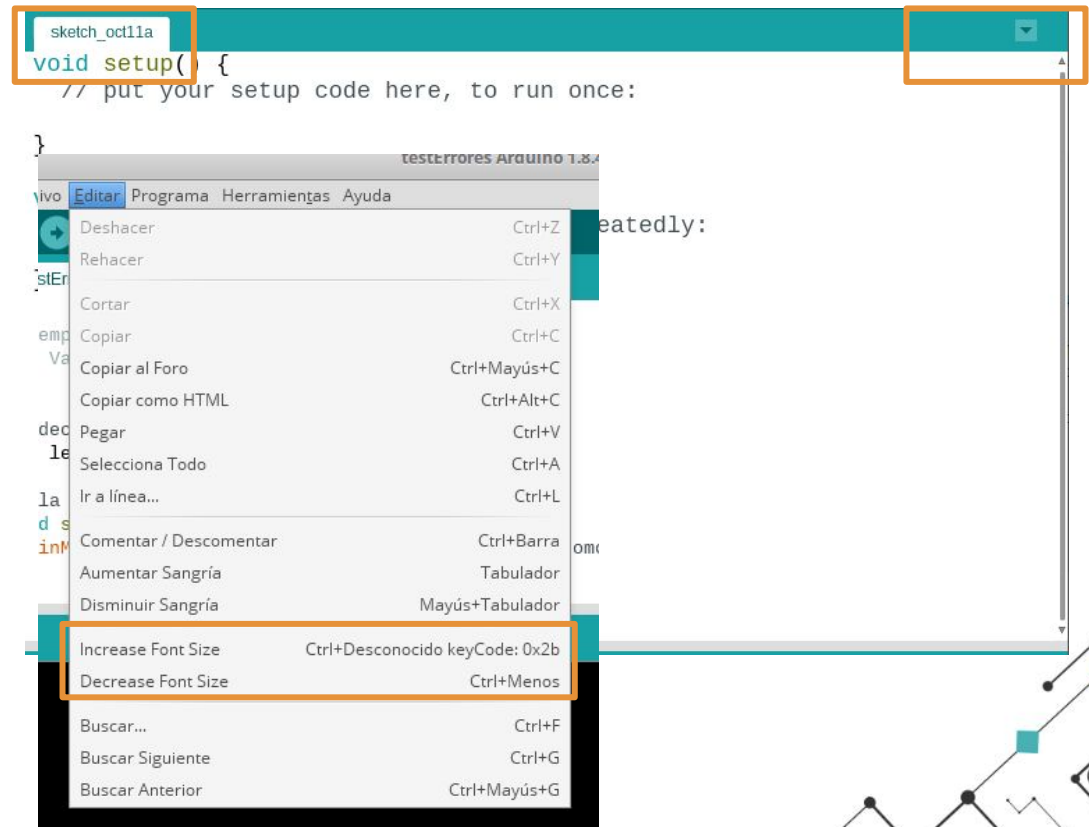
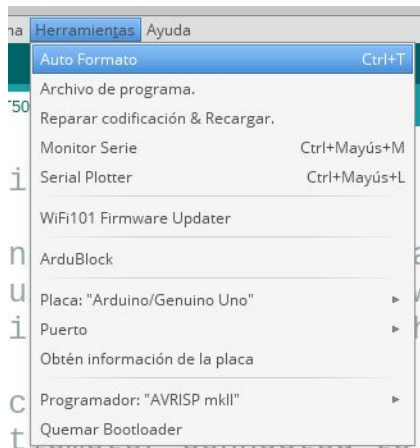
**Monitor Serie,** que nos permite ver la comunicación entre la placa y el PC





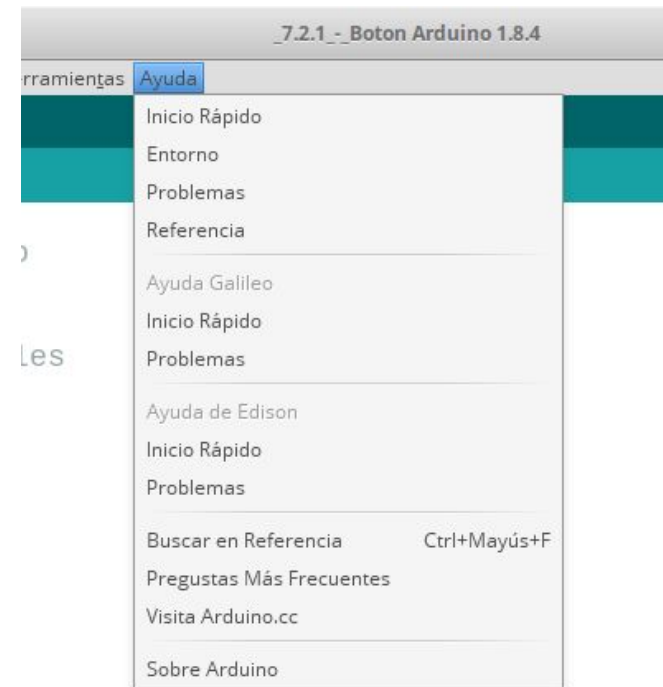
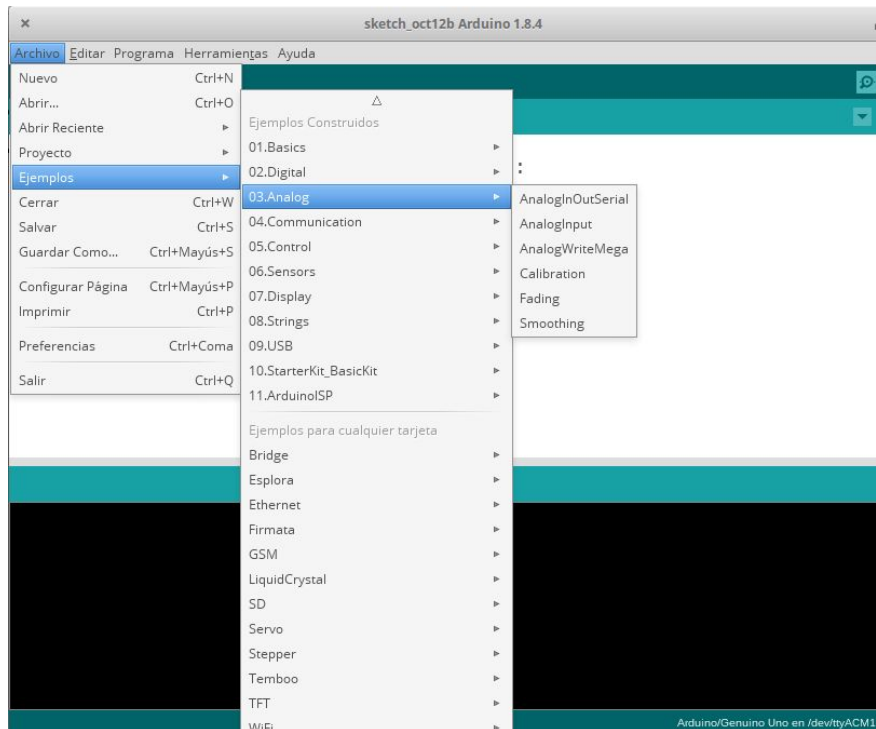
# Funcionalidades: Editor de código

- Resalta la sintaxis y el formato
- Proyectos (Sketch)
- Varios archivos
- Cambiar el tamaño de la letra
- AutoFormato



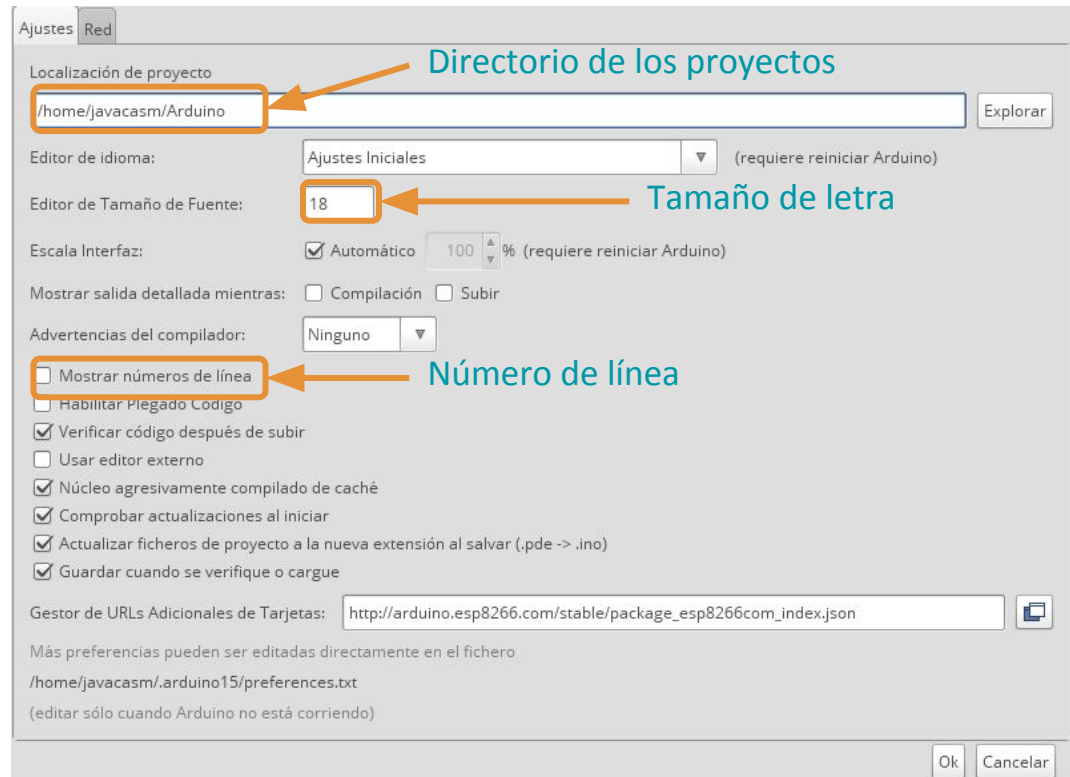
# Funcionalidades: ejemplos y documentación

Existen gran cantidad de ejemplos y documentación, previamente instalada en nuestro equipo



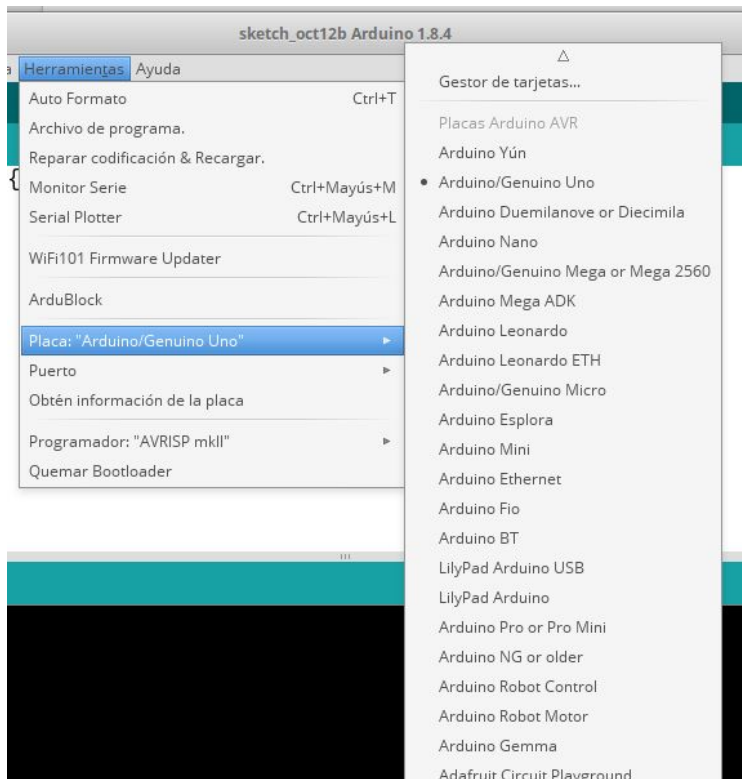
# Funcionalidades: Preferencias

En esta sección podemos modificar características como que se muestren o no números de línea, el tamaño de letra o la carpeta donde se guardan los directorios por defecto



# Configuración y carga de programas

¿Cómo configuramos nuestro entorno para una placa concreta?

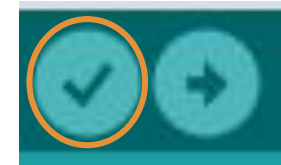


1

Seleccionar el modelo de Arduino

2

Verificar



3

**Compilar:** proceso en que se mezcla nuestro código con el código de las librerías

- Se añade el código de las librerías que usamos
- Se verifica que no hay errores
- Se convierte todo en instrucciones de código máquina para el procesador



# Lectura de errores

En este punto podemos encontrar que se producen errores. Nos mostrará dónde están y cómo resolverlos.

```
Archivo Editar Programa Herramientas Ayuda
testErrores
// declaracion de variables
int ledPin=13; // LED rojo asociado al pin 13

// la funcion setup se ejecuta una sola vez
void setup() {
  pinMode(ledPin, OUTPUT) // inicializa el pin 13 como una salida
}

// la funcion loop se ejecuta repetidamente de forma infinita
void loop() {
  expected ';' before '}' token
}
exit status 1
expected ';' before '}' token
```

Línea con el error

Error encontrado

Posición del error

Descripción del error

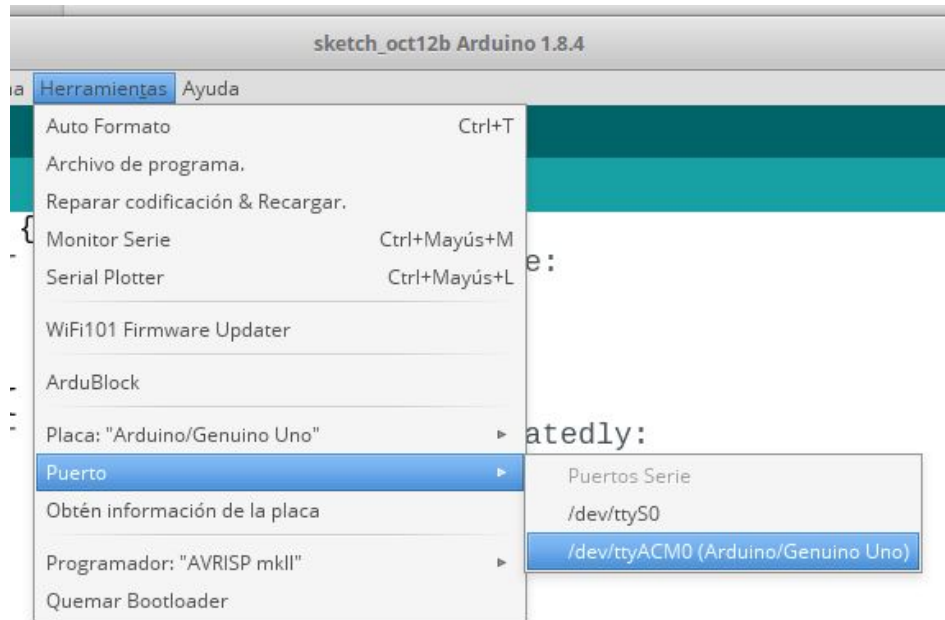


# Errores de código más frecuentes

- Falta un ; al final de la línea
- Falta o sobra una llave { ó }
- Falta o sobra un paréntesis
- Error en la definición del comentario
- No respetar las mayúsculas/minúsculas



# Configuración y carga de programas

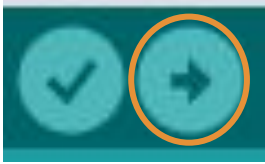


- Seleccionamos el puerto de conexión: Herramientas >> Puerto
- La placa debe estar conectada
- El nombre del puerto varía según SO y modelo de placa



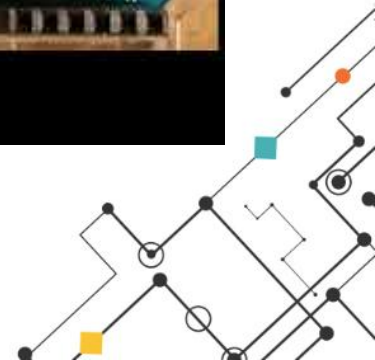


# Configuración y carga de programas



## Subir el programa a la placa

Se produce un parpadeo en las luces Tx y Rx. Nos indica que se ha producido comunicación con la placa. El programa se queda en la placa aunque la apaguemos o reseteemos.





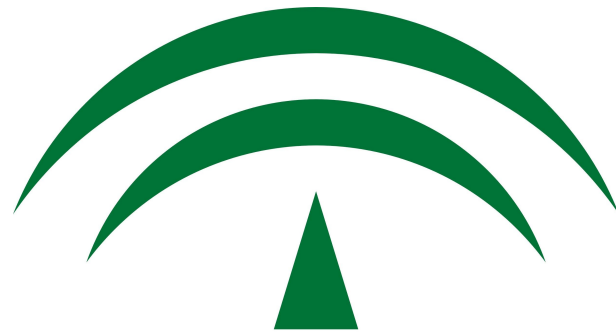
# Problemas más frecuentes al subir el programa

- Necesitamos un driver
- Problemas de permisos
- Problemas de conexión:
  - Cable mal conectado
  - Problema del cable USB

En esta guía podemos encontrar el modo de resolverlos:

<http://arduino.cc/en/Guide/troubleshooting>





**JUNTA DE ANDALUCIA**  
**CONSEJERÍA DE EDUCACIÓN**

Impulsa:



Colaboran:



**Autores:** José Pujol Pérez y José Antonio Vacas Martínez

**Dirección y realización:** [AVANTE Formación](#)