

MÓDULO 2: ECOSISTEMA “MAKER”

CURSO PROGRAMACIÓN DE PLACAS ROBÓTICAS



HARDWARE

Cultura Maker: cultura que representa una extensión basada en la tecnología de la cultura DIY (hágalo-usted-mismo).

Los proyectos *makers* se basan en el uso de algún tipo de electrónica para llevarlos a cabo.

Existen diferentes alternativas para los *makers*, siendo las principales:

- Arduino: es una plataforma de creación de electrónica de código abierto basada en hardware y software libre.
- Raspberry Pi: es un ordenador de placa simple y bajo coste, contando con potencia suficiente como para ejecutar programas.

HARDWARE: ARDUINO VS RASPBERRY

Son conceptos diferentes, aunque con puntos en común.

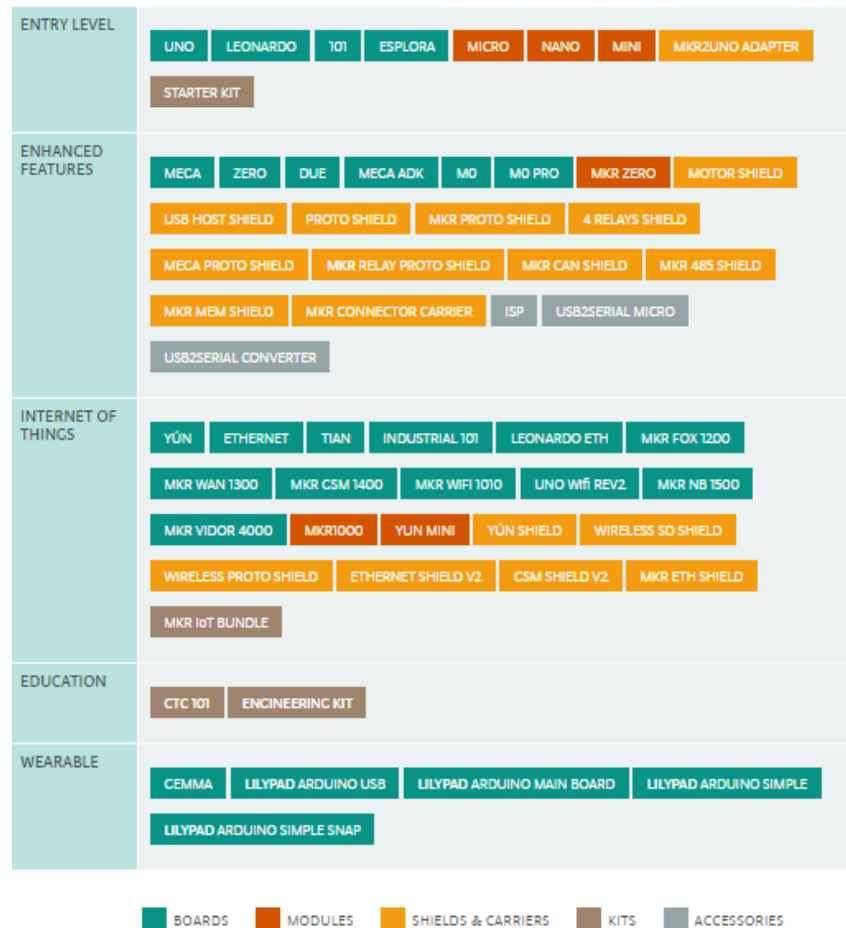
ARDUINO

- Hardware y Software libre
- Diseñada para aplicaciones de electrónica con enfoque en la facilidad
- Conectividad inmediata con el mundo
- Ejecuta directamente las aplicaciones programadas

RASPBERRY PI

- Software libre, Hardware propietario
- Diseñada como ordenador, posee una mayor capacidad de proceso
- Necesidad de uso de librerías para conexión con el mundo
- Necesita sistema operativo

HARDWARE: ARDUINO



Arduino dispone de una gran cantidad de placas y shields para diferentes necesidades:

- Placa: permiten construir dispositivo digitales que puedan sensor y controlar objetos del mundo real.
- Shield: amplia las capacidades de la placa conectándose a ésta.

Web oficial Arduino:

- <https://www.arduino.cc/en/Main/Products>

HARDWARE: PLACAS ARDUINO

Características principales

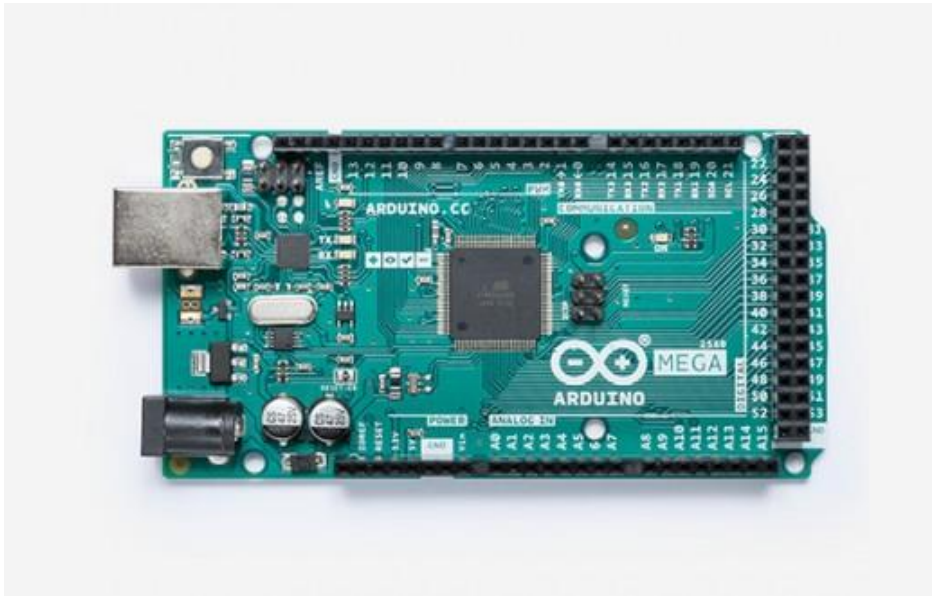


Arduino UNO rev3

- Placa más utilizada de la familia Arduino: amplia documentación, proyectos y apoyo de la comunidad.
- 14 entradas/salidas digitales
- 6 salida/entradas analógicas
- Procesador a 16 Mhz
- Memoria Flash: 32 Kb
- USB
- Jack de alimentación

HARDWARE: PLACAS ARDUINO

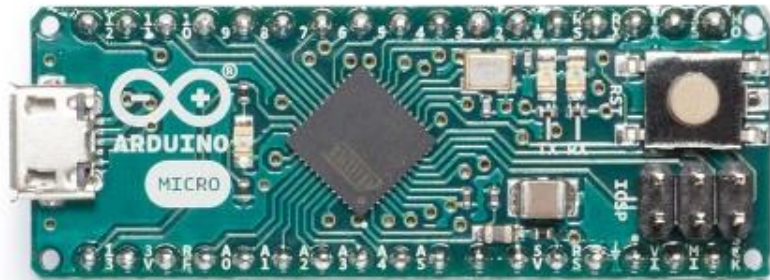
Características principales



Arduino MEGA 2560 rev3

- Diseñadas para proyectos más complejos: amplía las capacidades de UNO
- 54 entradas/salidas digitales
- 16 salida/entradas analógicas
- Procesador a 16 Mhz
- Memoria Flash: 256 Kb
- USB
- Jack de alimentación

HARDWARE: PLACAS ARDUINO



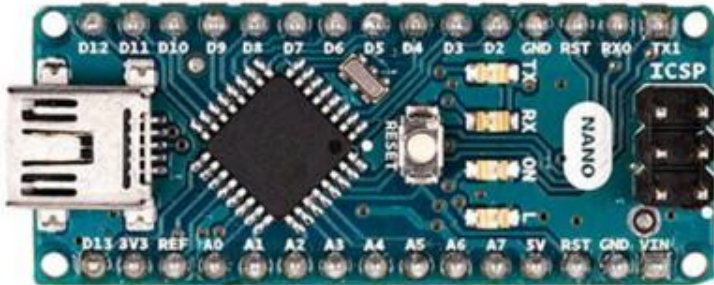
Arduino Micro

Características principales

- La más compacta, diseñada para integrarse en objetos cotidianos.
- 20 entradas/salidas
 - 12 pueden usarse como analógicas
- Procesador a 16 Mhz
- Memoria Flash: 32 Kb
- USB

HARDWARE: PLACAS ARDUINO

Características principales



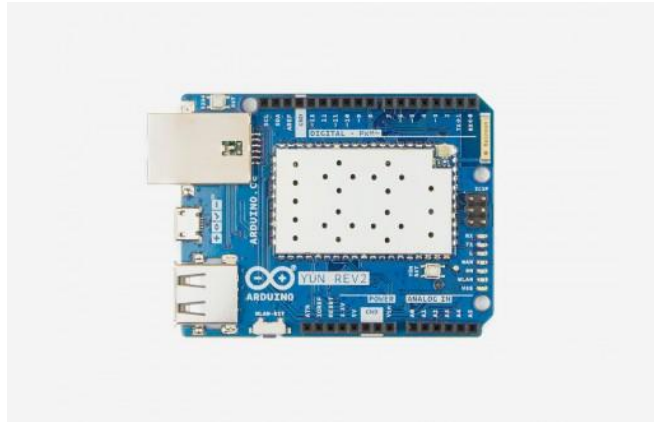
Arduino Nano

- Características similares a la UNO, pero construida con otra distribución más compacta.
- 14 entradas/salidas digitales
- 6 salida/entradas analógicas
- Procesador a 16 Mhz
- Memoria Flash: 32 Kb
- USB

HARDWARE: PLACAS ARDUINO



Genuino ZERO



Arduino YUN rev2



Arduino TIAN

HARDWARE: RASPBERRY PI

Es un ordenador de bajo coste y tamaño reducido.

- XXXX



Raspberry Pi 3

PROGRAMACIÓN

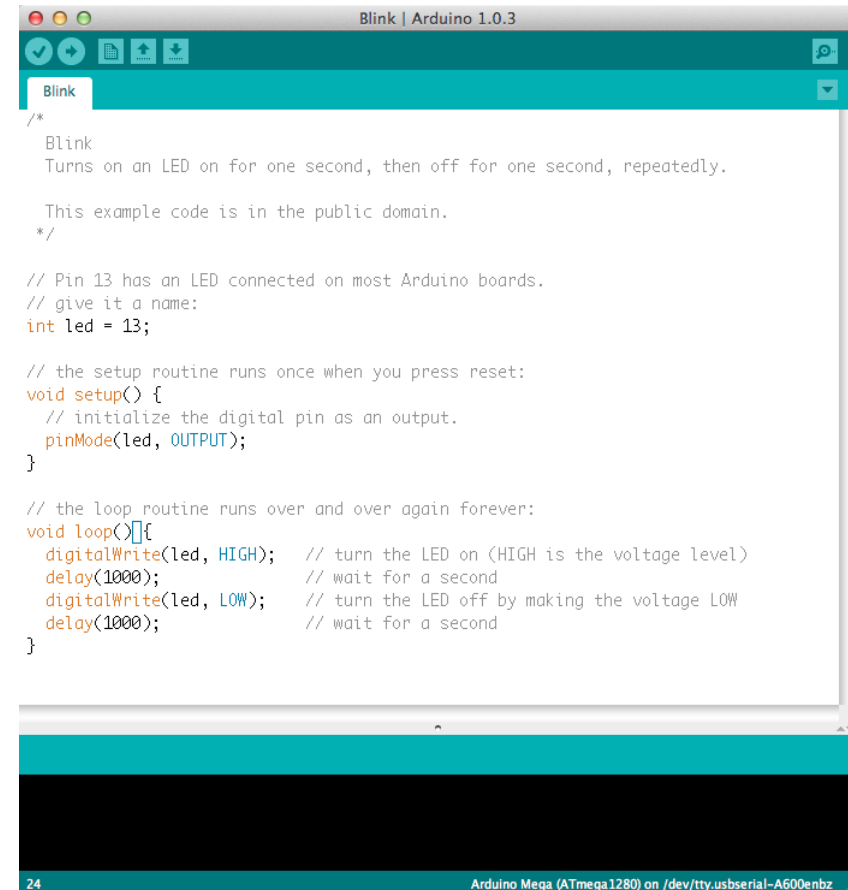
Es el proceso por medio del cual se diseña, codifica, limpia y protege el código fuente de programas de ordenador.

- El objetivo de la programación es la de crear software, que después será ejecutado por el hardware.
- Se guía por una serie de reglas y un conjunto pequeño de órdenes.
- En el mundo *maker* hay dos tipos de familias:
 - **Convencionales:** lenguajes basados en texto
 - **Por bloques:** lenguajes basados programación visual

PROGRAMACIÓN: CONVENCIONAL

La programación se realiza por estructuras de texto escritas bajo unas reglas establecidas para cada lenguaje.

- Muy versátil
- Permite desplegar toda la potencia de un lenguaje de programación
- Uso profesional
- Curva de aprendizaje más lenta

A screenshot of the Arduino IDE interface. The title bar at the top reads "Blink | Arduino 1.0.3". Below the title bar is a toolbar with icons for file operations and a "Verify" button. The main text area contains the C++ code for the "Blink" example. The code includes comments explaining the function, pin definitions, and the setup and loop functions. The status bar at the bottom indicates "24" and "Arduino Mega (ATmega1280) on /dev/tty.usbserial-A600enbz".

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop(){
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

PROGRAMACIÓN: POR BLOQUES

La programación se realiza mediante bloques de instrucciones que se agrupan de forma visual.

- Más fáciles de comprender e intuitivos
- Uso didáctico
- Menos versátiles
- Dificultad para crear programas complejos

