
Máster en Tecnologías de Análisis de Datos Masivos: BIG DATA

Internet de las Cosas en el Contexto de Big Data

PRÁCTICA 1: INTRODUCCIÓN A ARDUINO

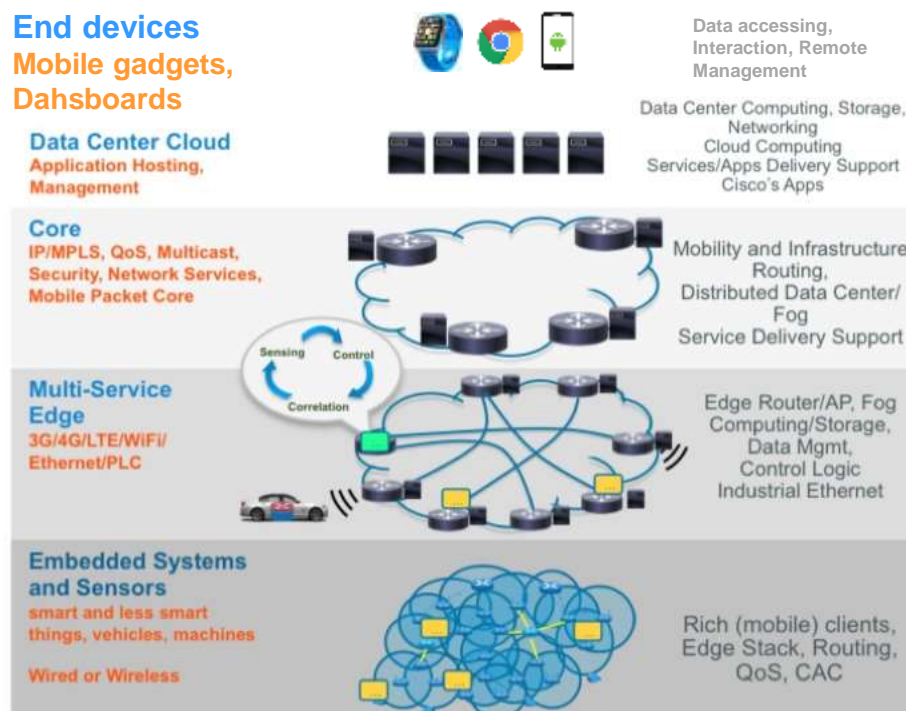
Enrique Mármol Campos enrique.marmol@um.es

Índice

- ▶ **Introducción**
- ▶ **Arduino**
- ▶ **Montaje del circuito**
- ▶ **Programación de Arduino**

Introducción

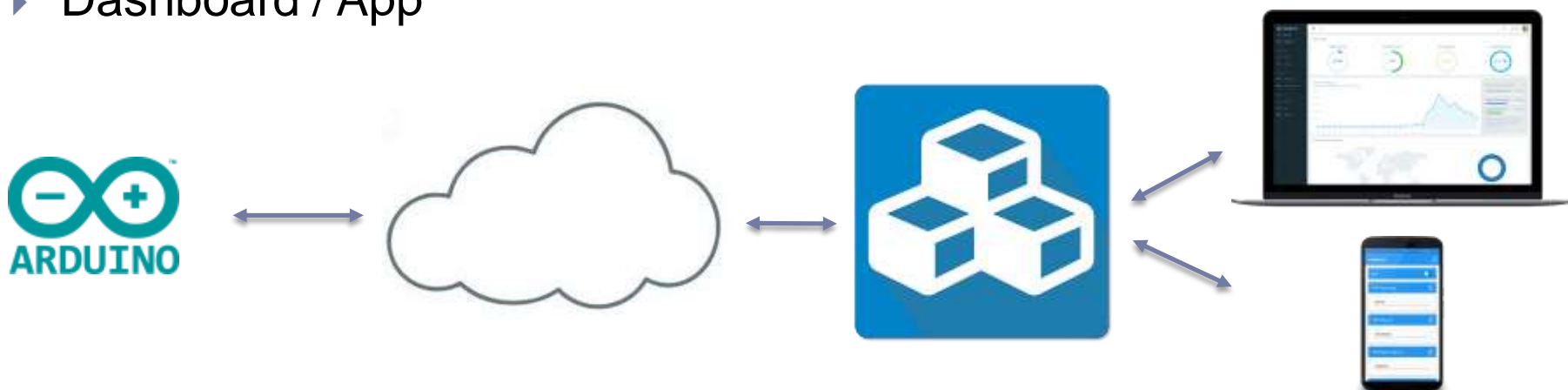
- ▶ En esta práctica vamos a explorar el proceso completo de despliegue de un servicio IoT
 - ▶ Montaje de dispositivo y sensor/actuador
 - ▶ Programación del dispositivo
 - ▶ Envío de datos a la nube
 - ▶ Visualización de datos en diversas plataformas: web y móvil



Introducción

Tecnologías

- ▶ Arduino + sensor (temperatura y humedad) + actuador (relé)
- ▶ Ethernet – Internet
- ▶ Plataforma Thinger.io
- ▶ Dashboard / App



Índice

- ▶ **Introducción**
- ▶ **Arduino**
- ▶ **Montaje del circuito**
- ▶ **Programación de Arduino**

r
q
u
i
t
e
c
t
u
r
a
d
e
R
e

Arduino

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en hardware flexible y fácil de usar



Arduino

Está basado en una tarjeta con un microcontrolador que permite conectar sensores, actuadores y otros elementos mediante sus entradas y salidas, analógicas y digitales

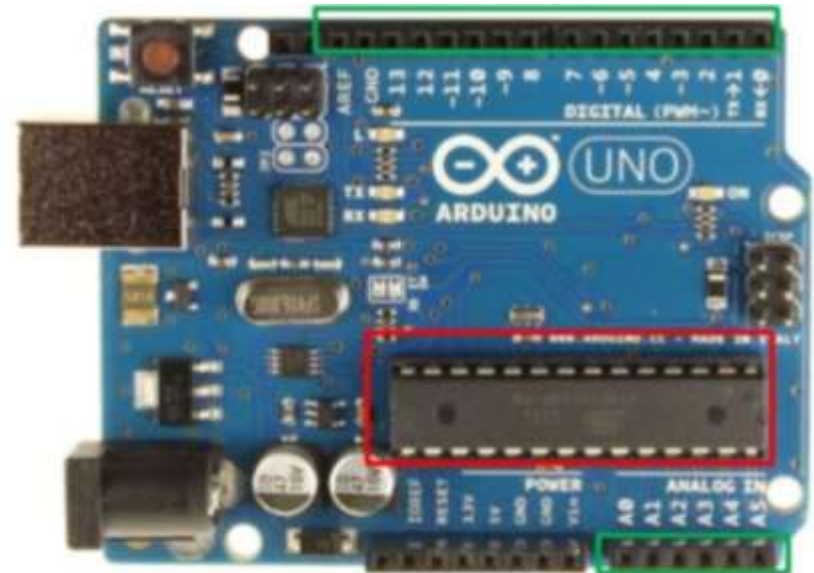
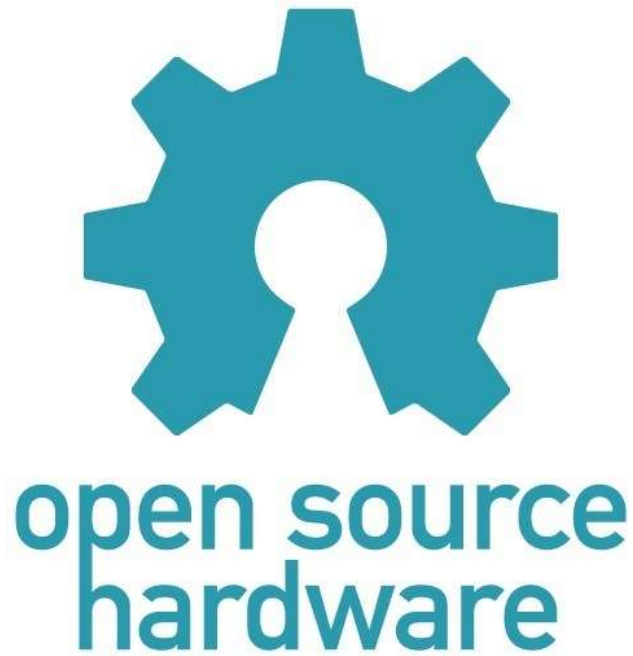


Imagen tomada de: <http://arduino.co/es/>

Arduino Uno R3

Arduino



Al ser **Open-Hardware**, tanto su diseño, como su distribución es libre. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin necesidad de adquirir ninguna licencia

Arduino



Duemilanove



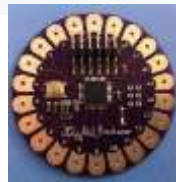
Mega



Uno



Nano



Lilypad



Mini

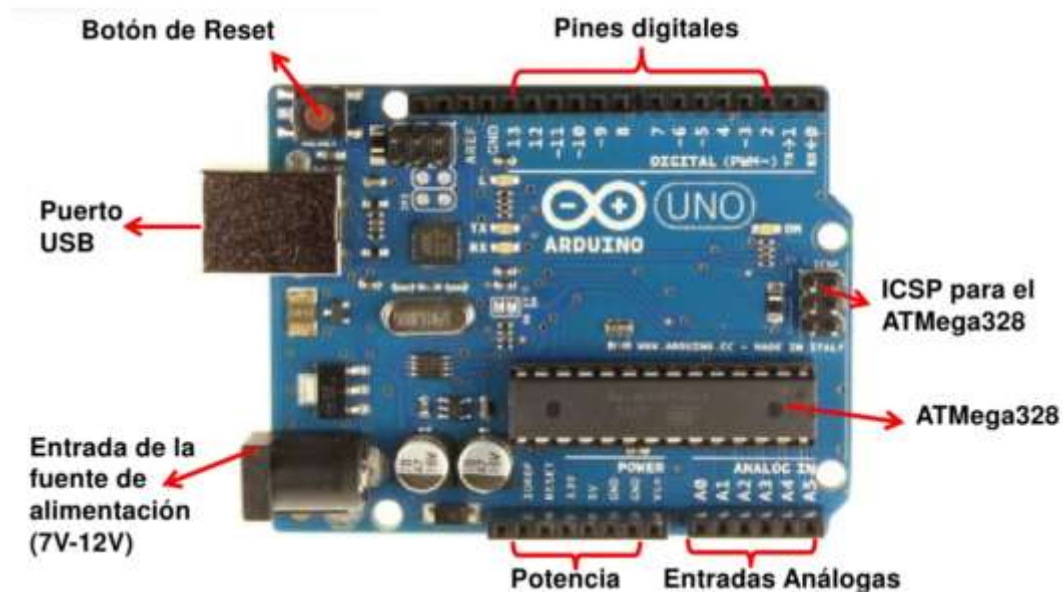


Leonardo

Arduino

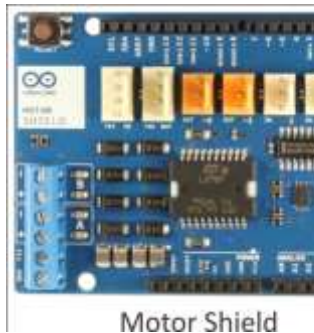
▶ Arduino Uno: características

- ▶ Microcontrolador: ATmega328
- ▶ Voltaje de operación: 5V DC
- ▶ Voltaje de alimentación: 7 – 12V DC
- ▶ Pines digitales I/O: 14
- ▶ Pines entrada analógica: 6
- ▶ Interfaz de programación: USB
- ▶ Frecuencia del Reloj: 16 Mhz



Arduino

- ▶ Arduino Shields: expansión de funcionalidad
- ▶ Permiten añadir a la placa básica de Arduino nuevas características: comunicación, sensorización, E/S en memorias externas, etc.



Motor Shield



GPS - GSM Shield



Ethernet shield

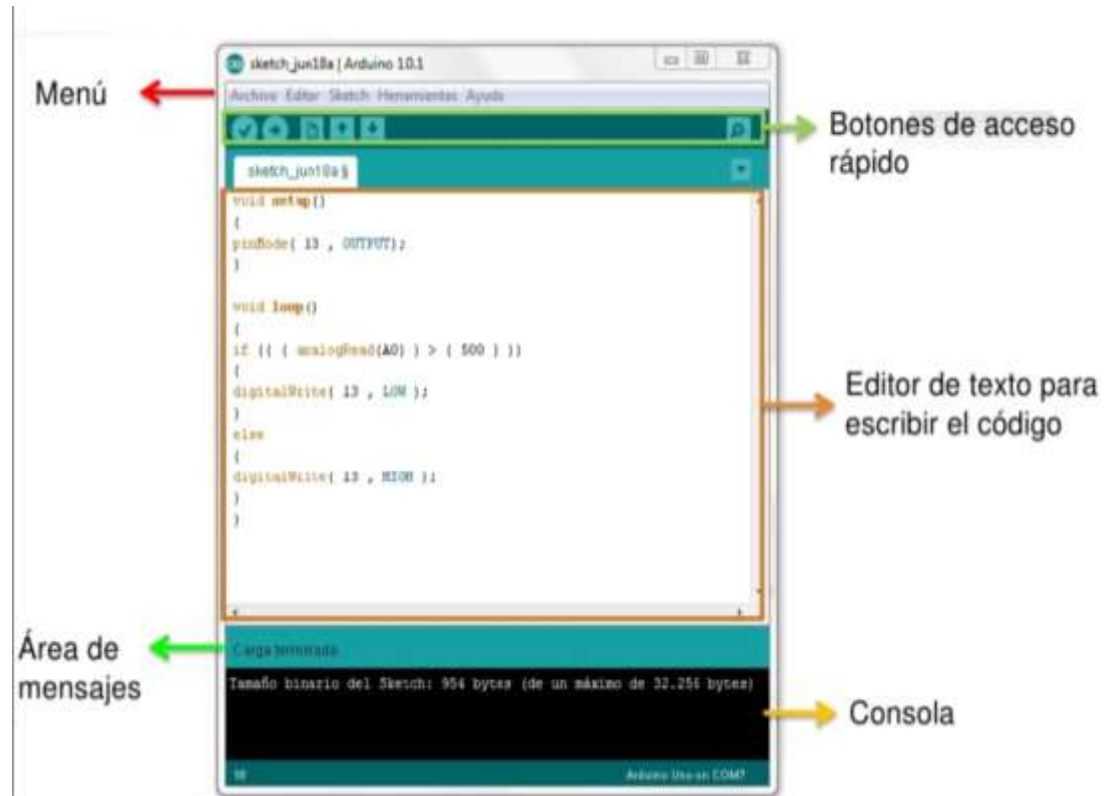


Celular Shield

Arduino

Programación

Arduino cuenta con un entorno de desarrollo nativo creado en Java, por lo que es multiplataforma



El lenguaje que utiliza es propio de Arduino y está basado en **C**

Arduino

▶ Programas Arduino

- ▶ Uso de librerías para facilitar la programación: todos los sensores/actuadores proporcionan su librería y documentación
- ▶ Tienen tres partes bien diferenciadas:
 - ▶ Importar librerías y declaración de variables globales
 - ▶ Función setup(): configuración inicial al arrancar el dispositivo
 - ▶ Función loop(): se repite constantemente: funcionalidad del programa

Arduino

```
digitemp_arduino code
/* ProjectsDunia
   http://projectsdunia.blogspot.in
   Arduino Digital Thermometer */
#include<LiquidCrystal.h>
int A=0;
int B;
int sensorpin=A0;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  lcd.begin(16, 2);
  lcd.print("Thermometer");
}
void loop() {
  A= analogRead(sensorpin);
  B= (A/2);
  lcd.setCursor(0,0);
  lcd.print("Temperature:");
  lcd.setCursor(0,1);
  lcd.print(B);
  lcd.print("C");
}
```


Arduino

► Hola mundo con Arduino: Blink

1. Conectar Arduino por USB al PC. Abrir el IDE de Arduino y comprobar que el puerto es correcto: Herramientas → puerto → seleccionar la placa
2. Abrir el *sketch* Blink:

Archivo → Ejemplos → 01. Basics → Blink

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

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

6 Arduino

- Hola mundo con Arduino: Blink
- 3. Verificar el código y cargarlo en la placa
- 4. Comprobar como el led que monta la placa se enciende y apaga a intervalos de 1 segundo
- 5. Modificar esos intervalos y volver a cargar el programa para observar el cambio de funcionalidad

Arduino

► Ejercicio LED Arduino:

1. Hacer señal S.O.S con el LED (tres destellos cortos, tres largos, y tres cortos).
2. Hacer código que escribiendo una letra por el puerto serial y active/desactive la señal S.O.S. y muestre por pantalla el estado de este.