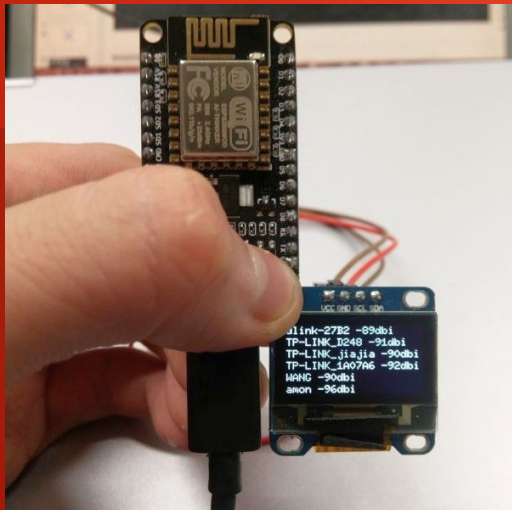




# IoT con NODEMCU



INTERNET  
OF THINGS



# INTERNET

Se podría definir como una red global de redes de ordenadores cuya finalidad es permitir el intercambio de información entre todos sus usuarios.



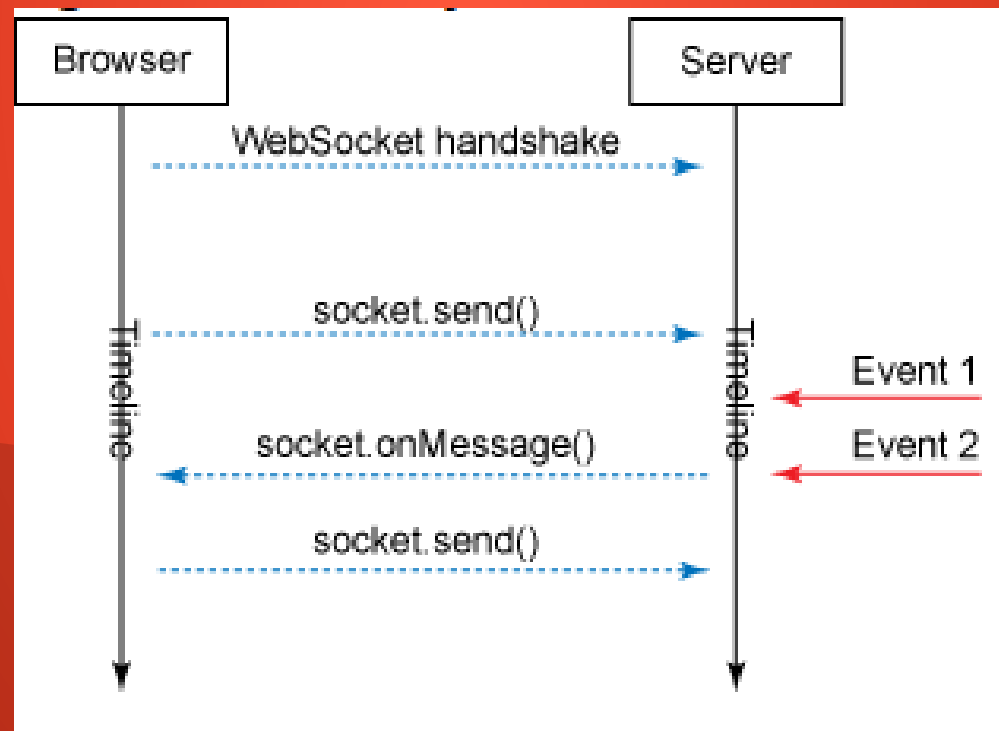
# HTTP

Es el protocolo de comunicación que permite las transferencias de información en la World Wide Web.



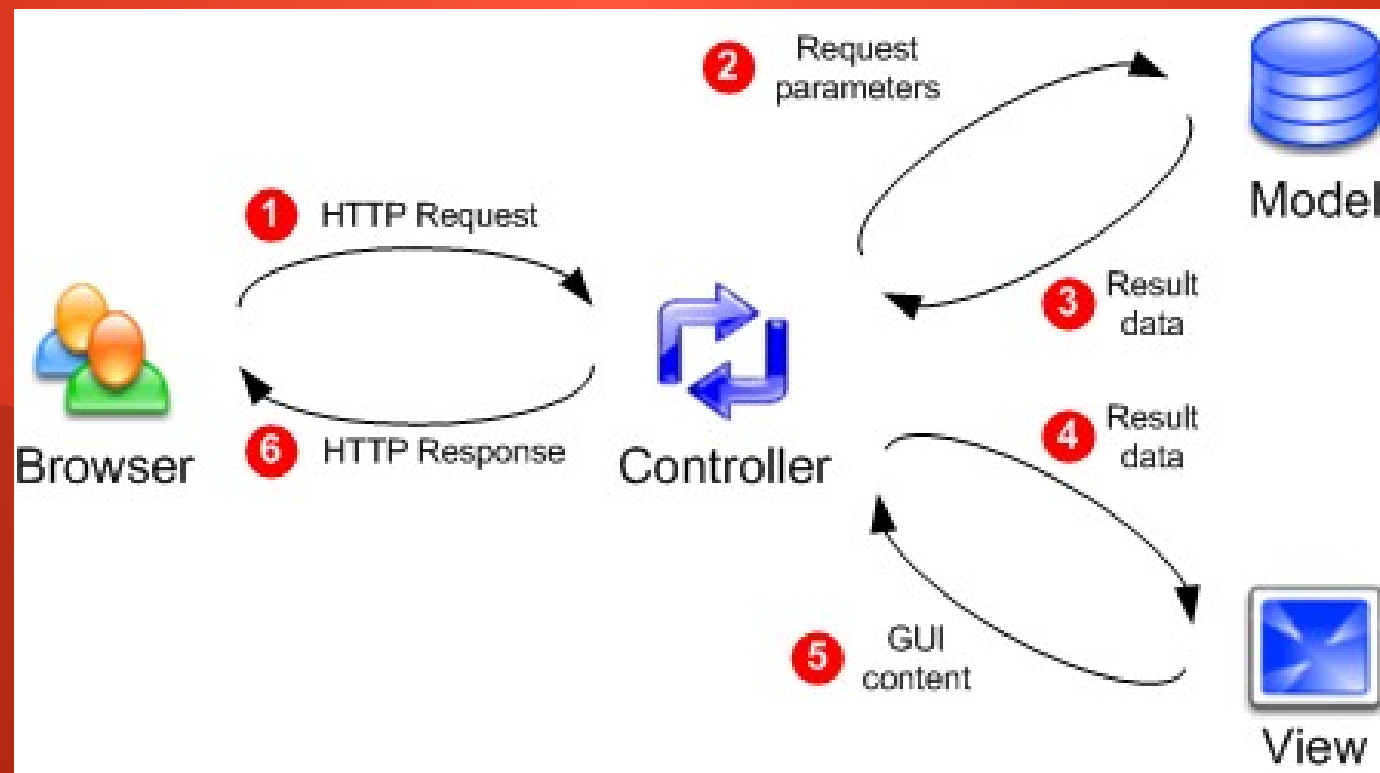
# WebSocket

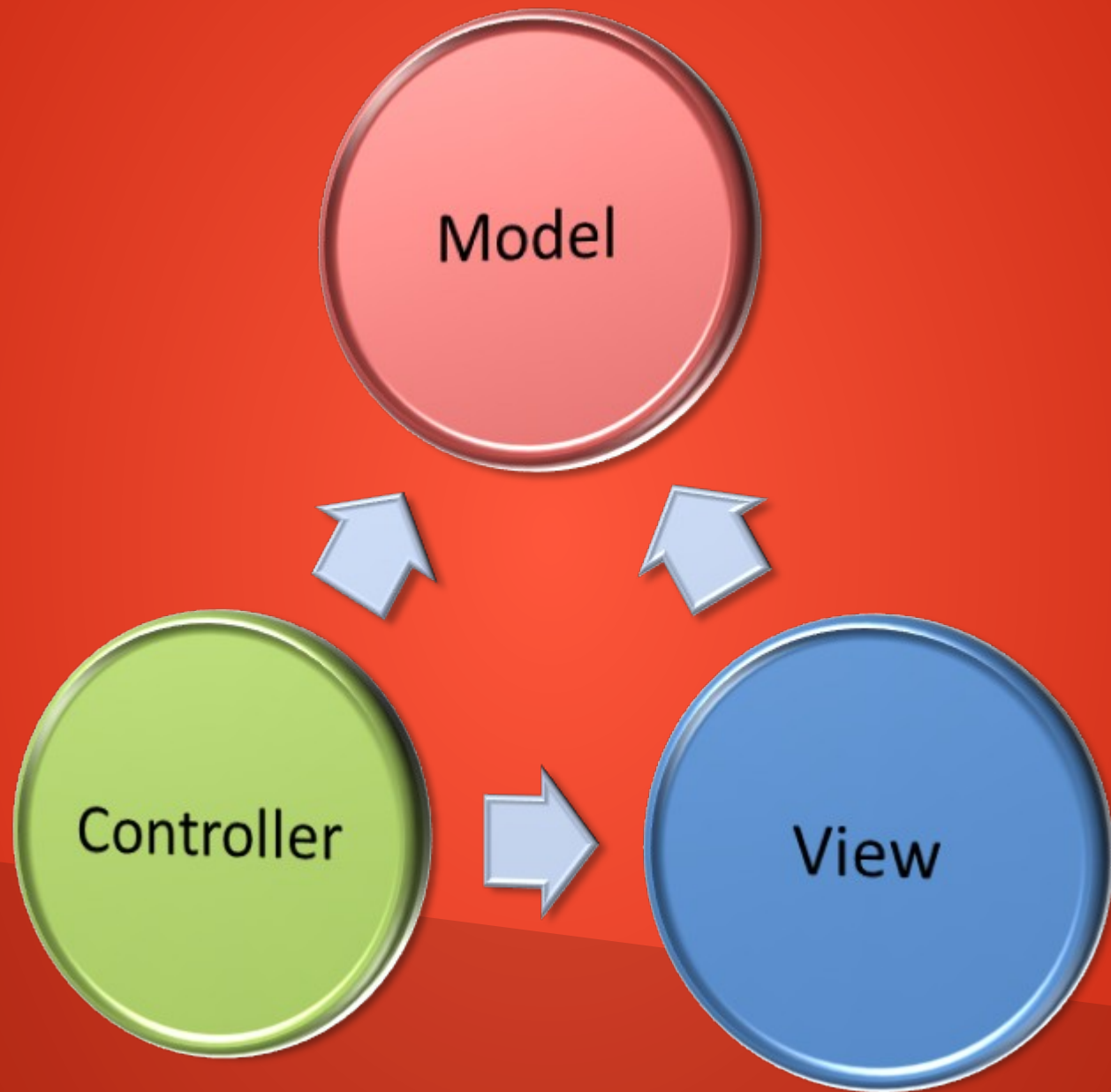
Es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP.

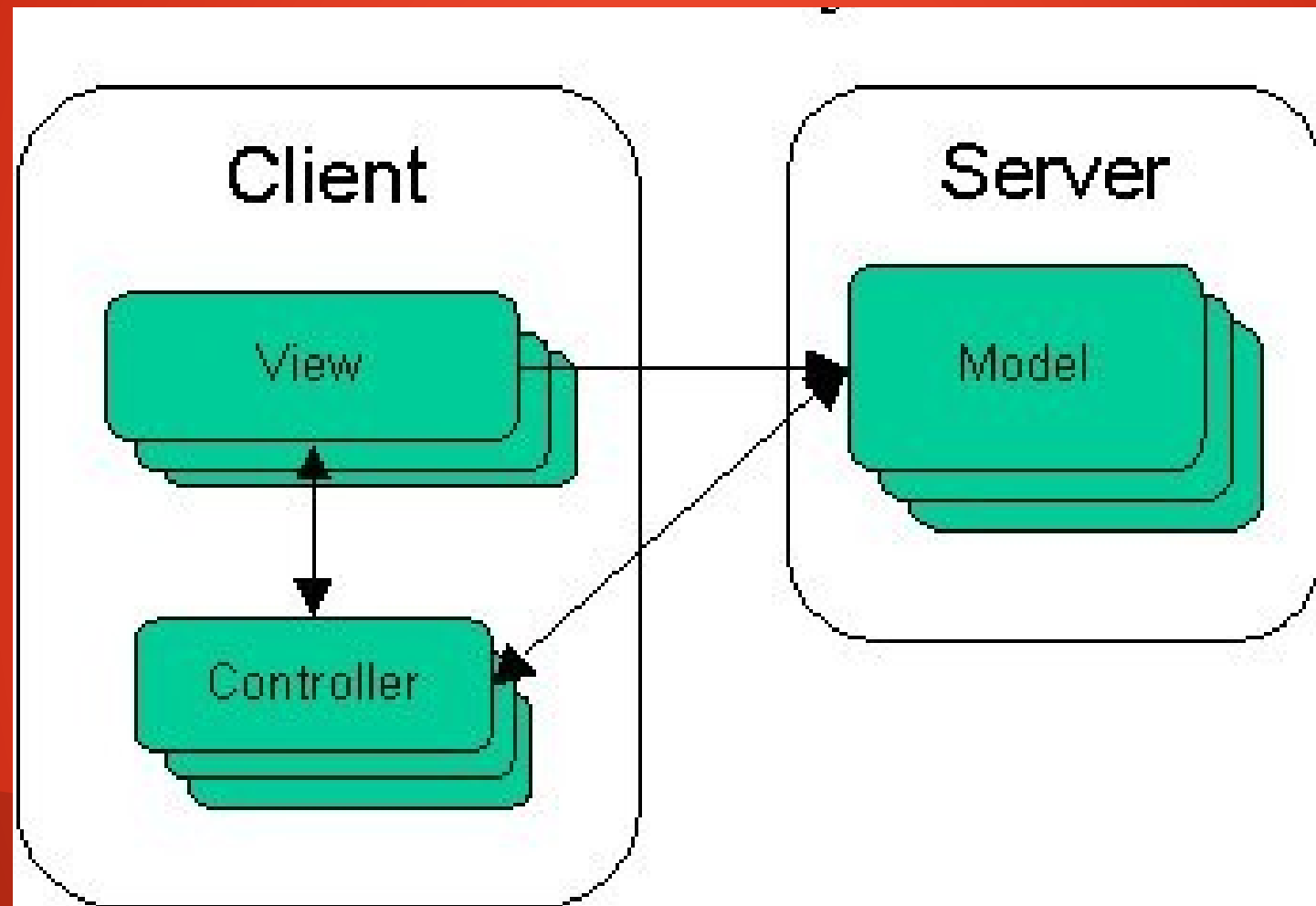


# Modelo MVC

Es un patrón de arquitectura de software por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.







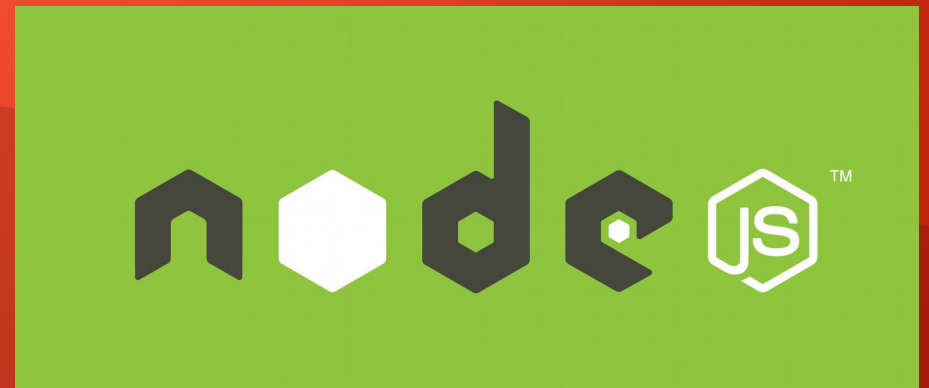


# Lenguajes de internet



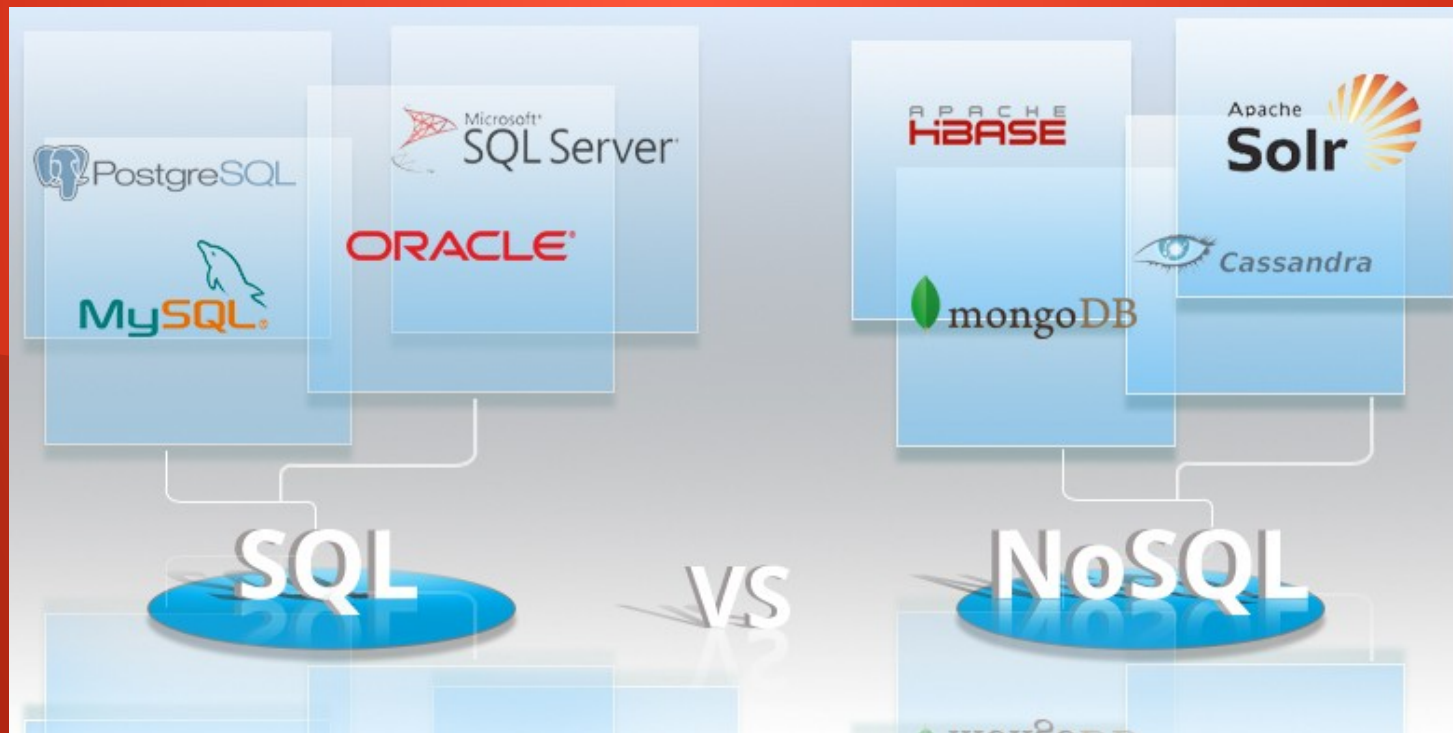


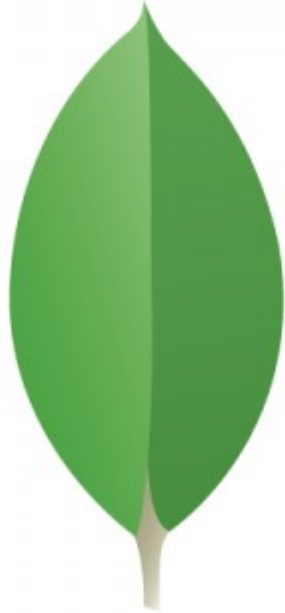
# Framework



# Base de Datos

- Almacenamiento de información.
- Tenemos las bases de datos relacionales y las no relacionales





mongoDB

- Mayor rapidez de búsqueda.
- No es necesario definir la estructura.
- Fácil integración con python
- Usa formatos JSON.

# Internet de las cosas

- Es un concepto que se refiere a la interconexión digital de objetos cotidianos con internet.



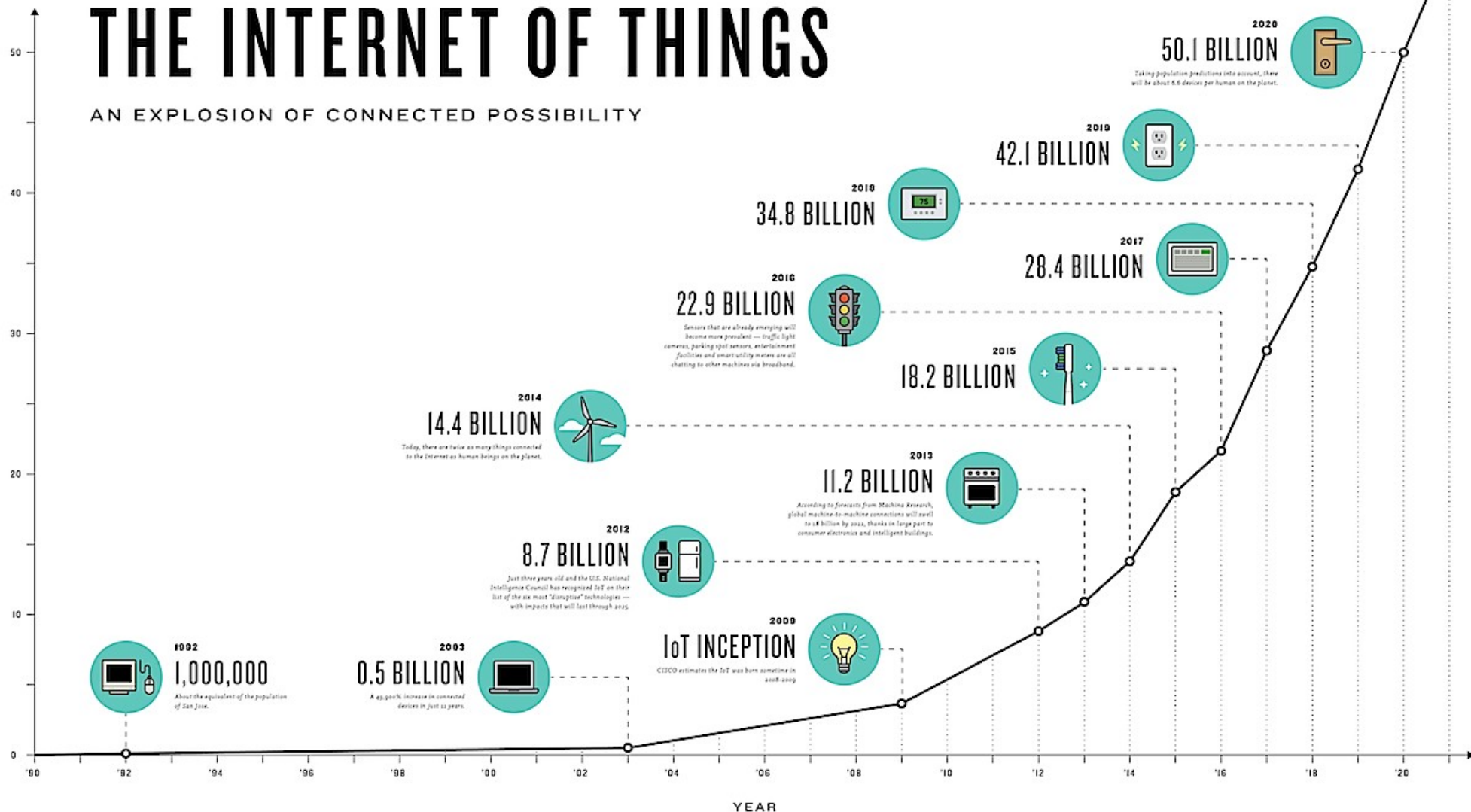




# THE INTERNET OF THINGS

AN EXPLOSION OF CONNECTED POSSIBILITY

BILLIONS OF DEVICES





# Protocolos de internet de las cosas

Generales

HTTP

WebSocket

Dedicados

MQTT

CoAP

# HTTP: API-REST

- REST, REpresentational State Transfer, es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP.
- Las APIs son como bibliotecas para poder ordenarnos.
- Para manipular los recursos, HTTP nos dota de los siguientes métodos con los cuales debemos operar:
  - GET: Para consultar y leer recursos
  - POST: Para crear recursos
  - PUT: Para editar recursos
  - DELETE: Para eliminar recursos.
  - PATCH: Para editar partes concretas de un recurso.

<http://192.168.2.8:80/led13/on>

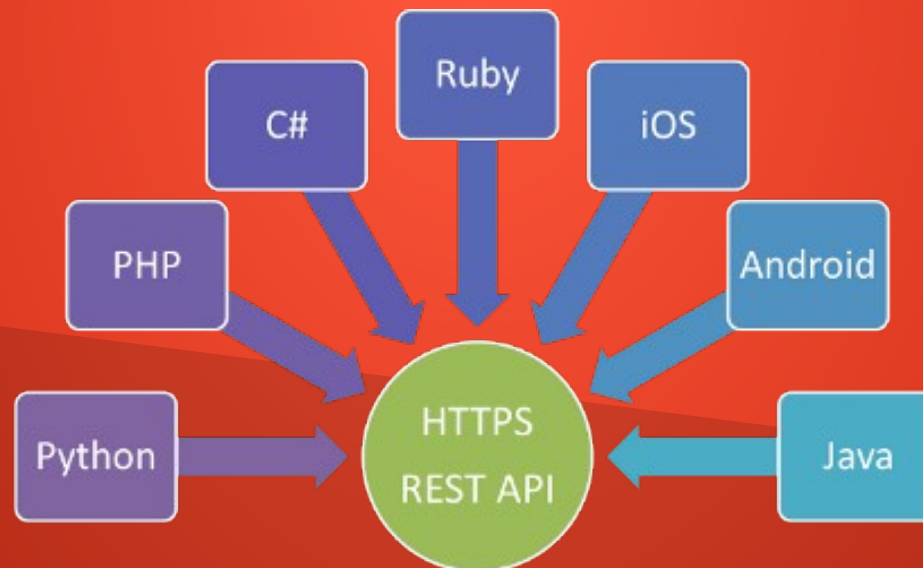
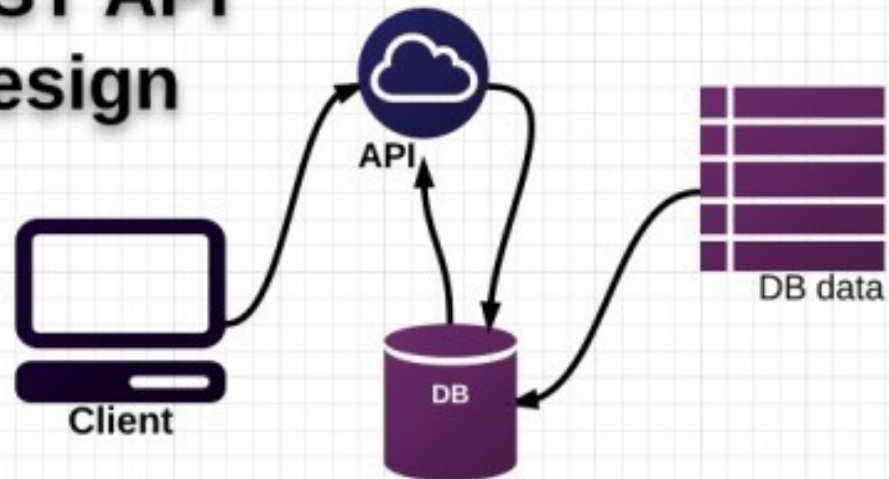
(c) Srinivas Tamada Production



[www.9lessons.info](http://www.9lessons.info)

# REST API Design

GET /tasks - display all tasks  
POST /tasks - create a new task  
GET /tasks/{id} - display a task by ID  
PUT /tasks/{id} - update a task by ID  
DELETE /tasks/{id} - delete a task by ID



# CURL

- Es un proyecto de software consistente en una biblioteca (libcurl) y un intérprete de comandos (curl) orientado a la transferencia de archivos. Soporta los protocolos FTP, FTPS, HTTP, HTTPS, TFTP, SCP, SFTP, Telnet, DICT, FILE y LDAP, entre otros.

```
admin — bash — 80x24
mac:~ admin$ curl -X POST 'https://ablanks.cloudant.com/testdb' -H 'Content-Type: application/json' -d '{"foo":"bar"}' | jq .
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total  Spent  Left   Speed
100   108    100    95    100    13     216     29  --:--:-- --:--:-- --:--:--    217
{
  "ok": true,
  "id": "faba67946873099b36e630343ee832eb",
  "rev": "1-4c6114c65e295552ab1019e2b046b10e"
}
mac:~ admin$
```

# MQTT

- MQTT es un protocolo abierto, sencillo, ligero y fácil de implantar.
- Es ideal para responder a las siguientes necesidades:
  - Esta orientado exclusivamente a hardware de recursos limitados como microcontroladores.
  - Está especialmente adaptado para utilizar un ancho de banda mínimo
  - Es ideal para utilizar redes inalámbricas.
  - Consume muy poca energía.
  - Es muy rápido y posibilita un tiempo de respuesta superior al resto de protocolos web.
  - Permite una gran fiabilidad si es necesario
  - Requiere pocos recursos procesadores y memorias
- El MQTT no es el único protocolo que intenta imponerse: otros como XMPP, REST API y CoAp también tienen ciertas ventajas.





temperature  
sensor

publish: "21°C"



MQTT-Broker

subscribe  
publish: "21°C"



laptop

subscribe  
publish: "21°C"



mobile device

1

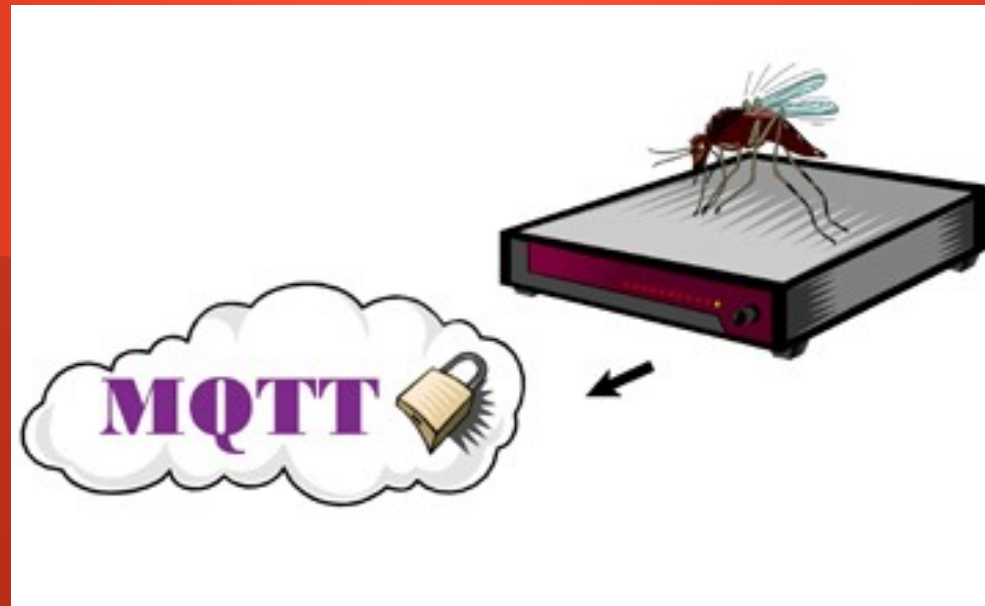
subscribe to  
topic: "temperature"

2

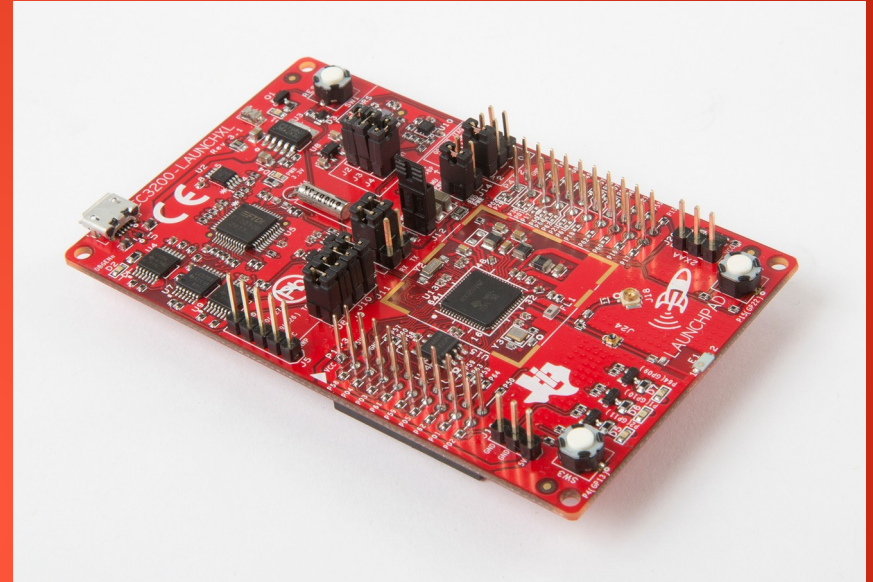
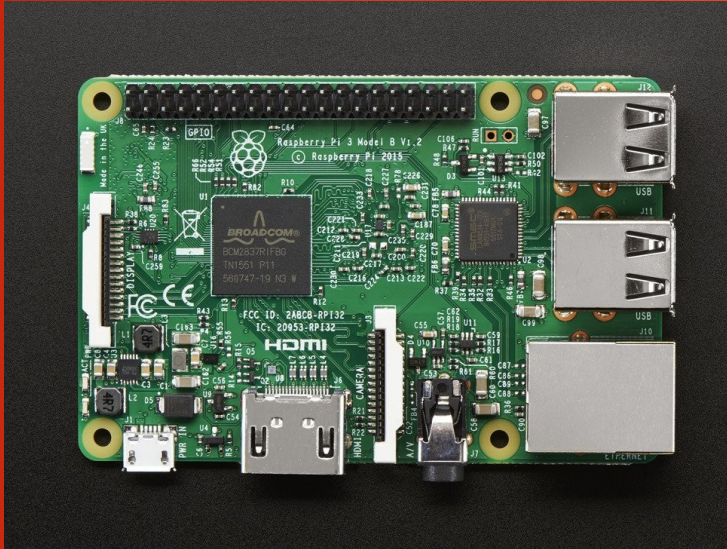
publish to  
topic: "temperature"

# Mosquitto

- Mosquitto es un broker OpenSource ampliamente utilizado debido a su ligereza lo que nos permite, fácilmente, emplearlo en gran número de ambientes, incluso si éstos son de pocos recursos.



# HARDWARE



# NODEMCU

- Procesador interno de 32 bits a 80 MHz y se le puede subir hasta 160Mhz.
- 80K de DRAM.
- 35K de IRAM , memoria rápida para el procesador.
- 1 Mb de memoria Flash para nuestros programas.
- Stack TCPIP WIFI a 2.4 Ghz.
- 30 Metros de alcance teórico.
- 16 pines GPIO programables disponibles.



- Full TCPIP stack incluido.
- Gestión completa del WIFI con amplificador incluido.
- 802.11 protocol y Wifi Direct (P2P) Soft-AP.
- Regulador y unidad de alimentación incluidos.
- Consumo en reposo <10 mW.
- Soporta antena externa para mayor alcance.
- Soporta el bus SPI.
- Soporta I2C
- 1 entrada analógica (Muy mal documentada)

# Versiones DEL NODEMCU

ESP8266



ESP-02



ESP-03



ESP-04



ESP-05



ESP-06



ESP-07



ESP-08



ESP-09



ESP-10



ESP-11



ESP-12





# NODEMCU 1.0



The official NodeMCU boards



devboard v0.9



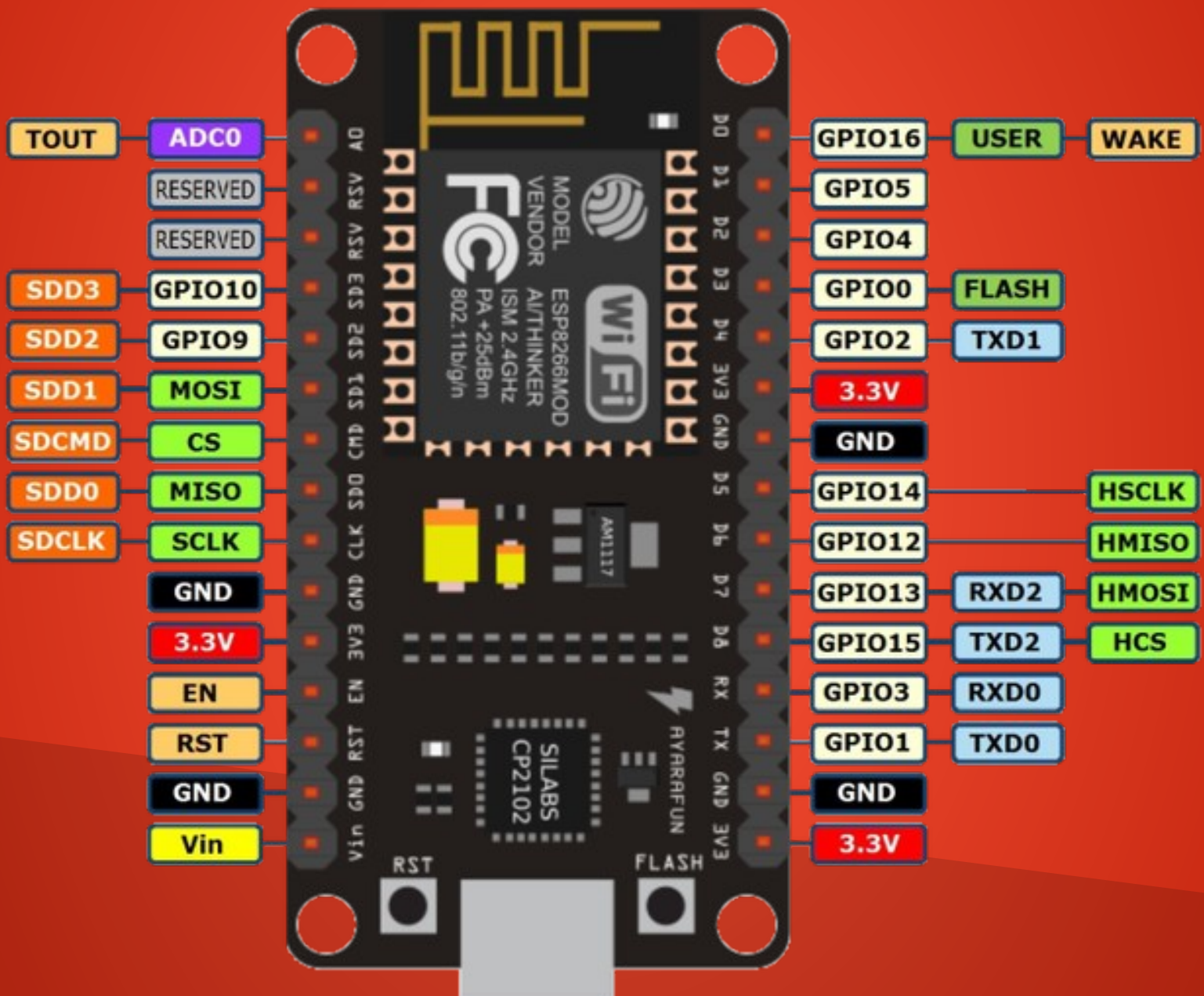
devboard v1.0

Support NodeMCU team, buy an official board!

Unofficial boards

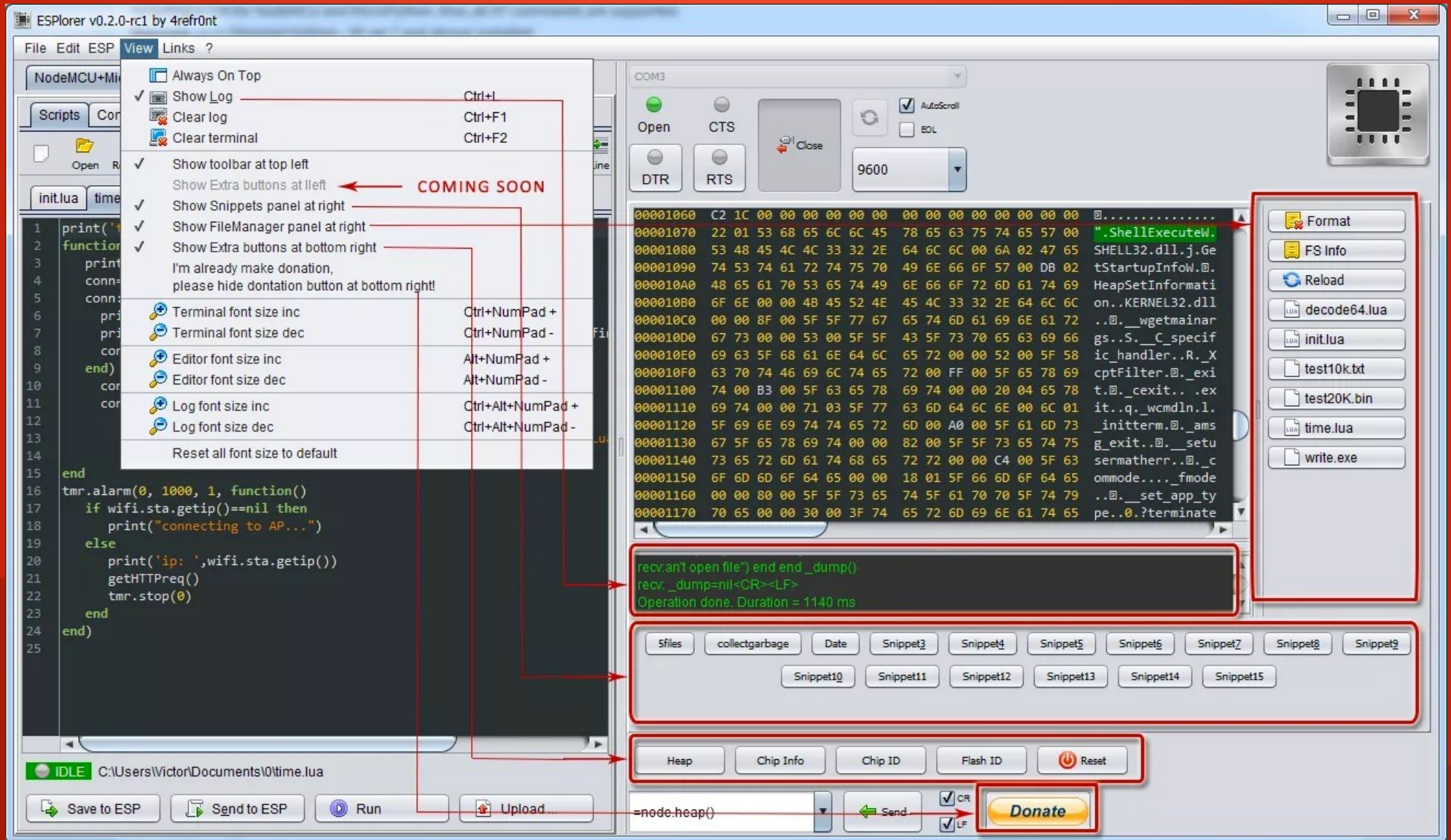


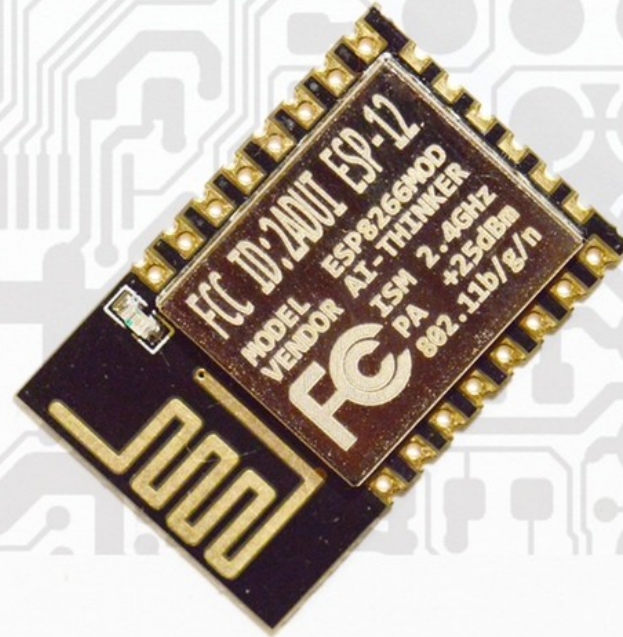
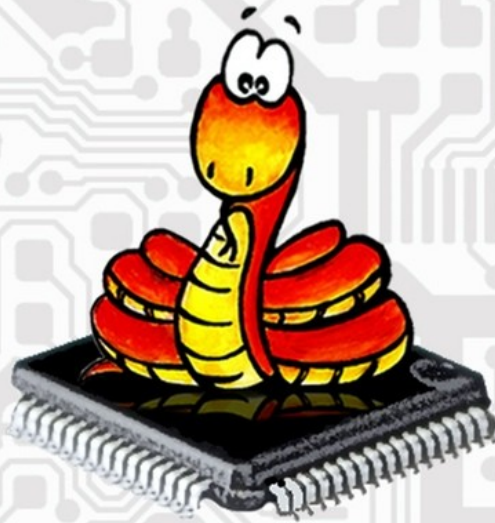
Unstable and not compatible boards





# Programación de NODEMCU





## MicroPython on the ESP8266

test\_blink12 Arduino 1.6.5

Archivo Editar Programa Herramientas Ayuda

test\_blink12

```
1 #include <Wire.h>
2 #include "OLED.h"
3
4 OLED display(4,5);|
5
6 void setup() {
7   pinMode(16,OUTPUT);
8   display.begin();
9   display.print("Hello World");
10 }
11
12 void loop() {
13   digitalWrite(16,1);
14   delay(500);
15   digitalWrite(16,0);
16   delay(500);
17 }
18 }
```

Compilado

4

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Serial, 115200, 4M (3M SPIFFS) on /dev/ttyUSB0

