



Instituto Tecnológico de San Juan del Río



Web Api concurrente con dispositivo de IoT

ALMACENAMIENTO EN PARALELO PARA IOT

P R E S E N T A N:

**Gomar Salvador Juan Manuel
Mejía Chávez María Fernanda**

Ingeniería en Sistemas Computacionales

San Juan del Río, Qro., Noviembre de 2020



1. Web Api concurrente con dispositivo de IoT.

1.1 Objetivo

Desarrollar un prueba de concepto de un servicio web que recibe información en paralelo desde diferentes dispositivos de IOT.

1.2 Antecedentes

Con base en los conocimientos adquiridos a lo largo de la carrera se destacan los más importantes para la realización de este pequeño proyecto el cual involucra aspectos tales como base de datos, estructura de programación, base de datos noSQL, y rubros importantes para el desarrollo de dispositivos de IoT. Destacando las materias más importantes requeridas para este proyecto son: IoT, todos los tópicos de Base de datos SQL y NoSQL, Programación web y tópicos de programación en general así como las materias troncales de Electrónica.

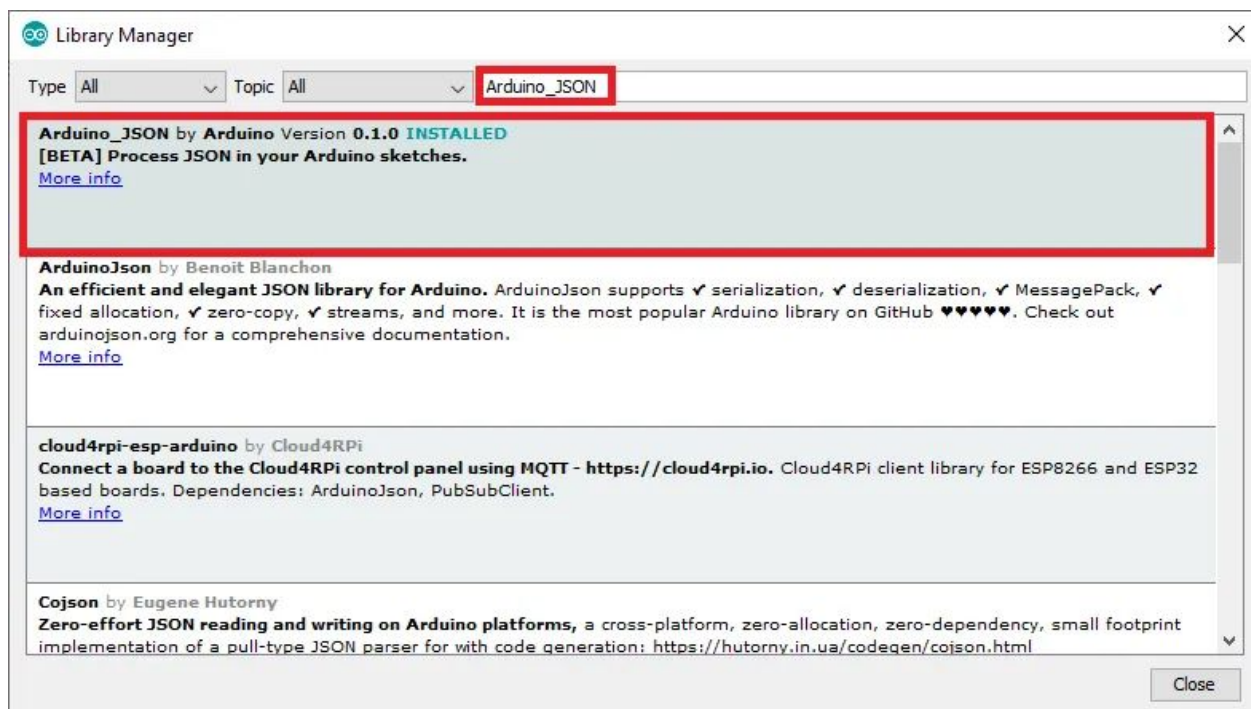
1.3 Materiales



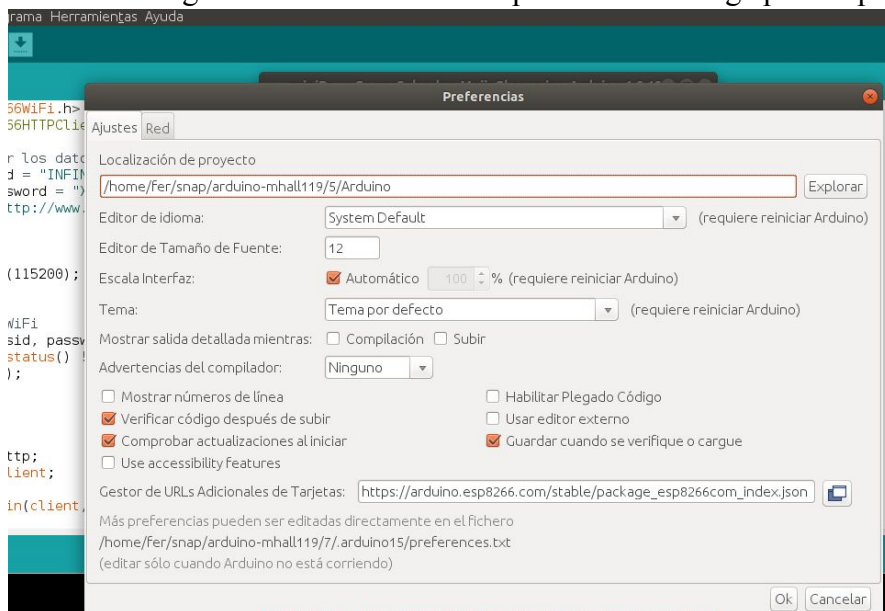
- **Nodemcu ESP8266**
- Protoboard
- Cables para proto
- 2 interruptores on/off
- Diodo Led
- Resistencias
- Conexión estable WiFi
- Cable de datos

1.4 Software

Se destaca la importancia de las librerías para el software de Arduino IDE necesarias para los controladores de Json. El entorno está escrito en Java y basado en Processing. El cliente de API de escritorio para REST.

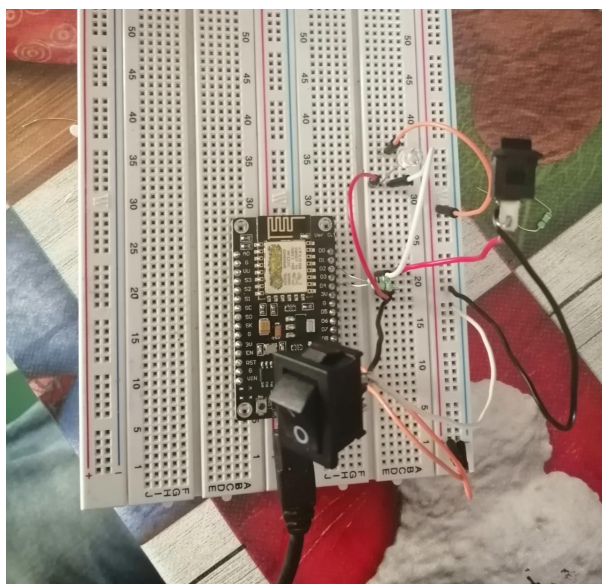


Así como integrar correctamente en las preferencias la liga para la placa [ESP8266](https://arduino.esp8266.com/stable/package_esp8266com_index.json).



1.5 Descripción del proyecto.

Se divide en dos componentes el dispositivo de IoT y Servidor de web API, cada uno juega un rol específico son sus respectivas tareas, sin embargo como Servidor y el Dispositivo de IoT, serán escritos en lenguajes de programación distintos, deberán usar como formato de transporte de información JSON. Obtener información proporcionada por un servidor que facilita al hacer la llamada a un endpoint/URL, por ejemplo, en formato CSV, JSON o XML. En nuestro caso un archivo Json.



1.5.1 Construcción del circuito

Para la construcción del circuito nos apoyamos de las resistencias para que no tenga un exceso de paso de corriente que pueda afectar el circuito o dañar algún componente y ubicamos las salidas D2 y D4.

1.5.2 Sistema

Método POST para almacenar la información en la base de datos.

Método GET para extraer todos los registros de la base de datos.



1.5.2.1 Código Arduino IDE

```
miniProy_GomarSalvador_MejiaChavez.ino Arduino 1.8.12
Archivo Editar Programa Herramientas Ayuda
miniProy_GomarSalvador_MejiaChavez.ino $
//#####
//#          Gomar Salvador Juan Manuel          #
//#          Mejía Chávez María Fernanda          #
//#####
//#include <ArduinoJson.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
//#include <ArduinoJson.h>
#include <ESP8266WiFi.h>
String server = "";

const char *ssid      = "INFINITUM6708_2.4";
const char *password  = "XY8zTaaN77";

const int btnP12 = 4; //D2
const int btnP1  = 2; //D4

int p1 = 0;
int act1 = 0;
int p12 = 0;
int act12 = 0;

void setup() {
```

```
miniProy_GomarSalvador_MejiaChavez.ino $
void loop() {

  if (WiFi.status() == WL_CONNECTED) {
    if (digitalRead(btnP1) == HIGH) {
      act1 = 1;
      if (p1 != act1) {
        Serial.println("P1 ON Right");
        post("P1", "ON Right");
        delay(1000);
        p1 = 1;
      }
    }
    else {
      p1 = 0;
      if (p1 != act1) {
        Serial.println("P1 OFF Left");
        post("P1", "OFF Left");
        delay(1000);
        act1=0;
      }
    }
  }
  if (digitalRead(btnP12) == HIGH) {
```

```
miniProy_GomarSalvador_MejiaChavez.ino $
    Serial.println("P1 OFF Left");
    post("P1", "OFF Left");
    delay(1000);
    act1=0;
  }
}
  if (digitalRead(btnP12) == HIGH) {
    act12 = 1;
    if (p12 != act12) {
      Serial.println("P2 ON Right");
      post("P2", "ON Right");
      delay(1000);
      p12 = 1;
    }
  }
  else {
    p12 = 0;
    if (p12!=act12) {
      Serial.println("P2 OFF Left");
      post("P2", "OFF Left");
      delay(1000);
      act12=0;
    }
  }
}
```

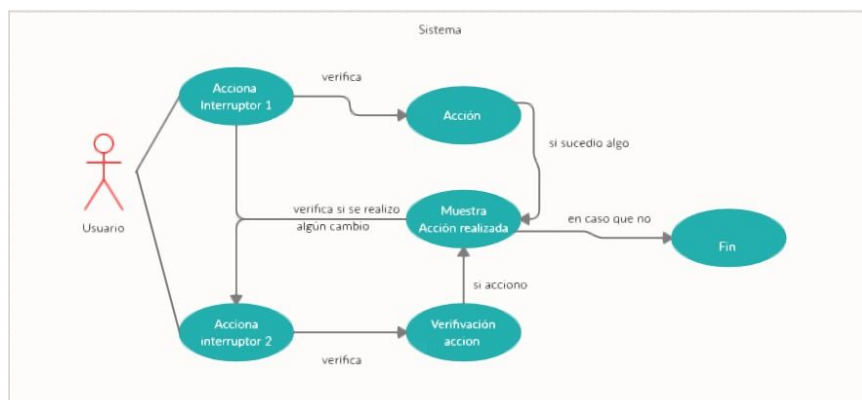


1.5.2.2 Código Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
index.js x
home > fer > Documentos > ITEC > 9 SEMESTRE > Almacenamiento en paralelo > index.js > app.get('/api/miniP') callba
1 //#####
2 //# Gomar Salvador Juan Manuel #
3 //# Mejia Chávez Maria Fernanda #
4 //#####
5 'use strict'
6 const spress = require('express')
7 const bodyPaser='body-parser'
8 const mongoose = require('mongoose')
9 const MiniP =require('./models/MiniP.js')
10 const app=express()
11 const port = process.env.PORT ||
12 app.use(bodyPaser.urlencoded({ extended: false}))
13 app.use(bodyPaser.json())
14
15
16 app.get('/api/miniP',(req, res) =>{
17   MiniP.find({}, (est,MiniP) =>{
18     if(est) return res.status(500).send({message: 'Request Error'})
19     if(MiniP) return res.status(404).send({message: 'No hay valores'})
20     res.send(200, {MiniP})
21   })
22 })
23
24 > app.get('/api/Min/:btnID',(req,res) => {--
31
32 app.post('/api/btnID',(req,res)=>{
33   console.log('POST/api/btn')
34   console.log(req,body)
35   let MiniP = new MiniP()
36
37   MiniP_btn = req.body.btn
38   MiniP_pos = req.body.post
  
```

1.5.2.2 Diagrama UML



1.5 Conclusión

Una de las principales funciones de las API es poder facilitarle el trabajo, ya que no necesitarás crear desde cero un sistema sino podemos facilitar el uso de sus características optimizando los recursos y utilizando las herramientas que programamos para ser llamadas una y otra vez siempre



y cuando sea necesario. En este proyecto se sectorizan principalmente utilizando un determinado servicio el que contendrá los datos que deseamos conocer involucrando un dispositivo físico en este caso el dispositivo IoT. El mayor problema al que nos enfrentamos era al momento de que el dispositivo escuchara las peticiones pero logramos solucionarlo. Otro problema al que nos enfrentamos fue el hecho de que nuestros equipos presentan muchas fallas para la instalación de paquetes y algunos software. Por lo que nos tomó más tiempo el desarrollo del trabajo aquí presentado.

Bibliography

GitHub. (n.d.). esp8266 / Arduino. esp8266. <https://github.com/esp8266/arduino>

