



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA



TECNOLOGÍAS DE SOFTWARE PARA ELECTRÓNICA

COMUNICACIÓN CON SERVICIOS WEB EN EL ESP8266
NODEMCU Y MySQL

DOCENTE: ING. DARWIN ALULEMA

ESTUDIANTES:

SALAZAR ISABEL

MORALES STEVE

REINOSO SANTIAGO

NRC: 4463

SANGOLQUÍ, 12 DE JUNIO DEL 2019

Índice general

Índice general	2
0.1. PLANTEAMIENTO DEL PROBLEMA	3
0.2. OBJETIVOS	4
0.2.1. OBJETIVO GENERAL	4
0.2.2. OBJETIVOS ESPECIFICOS	4
0.3. ESTADO DEL ARTE	5
0.4. MARCO TEÓRICO	6
0.4.1. INTERNET DE LAS COSAS (IOT)	6
0.4.2. ESP8266 NODEMCU	7
0.4.3. SERVICIOS WEB	9
0.4.4. SPRING BOOT	9
0.4.5. MySQL	9
0.5. DIAGRAMAS	10
0.6. LISTA DE COMPONENTES	11
0.7. MAPA DE VARIABLES	12
0.8. EXPLICACIÓN DE CÓDIGO FUENTE	13
0.9. DESCRIPCIÓN DE PRERREQUISITOS Y CONFIGURACIÓN	16
0.10. CONCLUSIONES	19
0.11. RECOMENDACIONES	20
0.12. CRONOGRAMA	21
0.13. LINKS DE OVERLEAF Y GITHUB	21
0.14. BIBLIOGRAFIA	22
0.15. MANUAL DE USUARIO	23
0.16. HOJAS TÉCNICAS	27

0.1. PLANTEAMIENTO DEL PROBLEMA

El desarrollo de sistemas de tecnología basadas en el empleo de servicios web como herramienta principal sigue desarrollándose de manera acelerada, es así que en este tiempo la mayoría de aplicaciones de control o cualquier otro tipo de estas llevan consigo una implementación de un control software y este a su vez lleva un servicio web como apoyo para el desarrollo de su funcionalidad, en conjunto con ESP8266 NODEMCU el cual es un módulo wifi que permite enlazarse con el internet para así establecer una conexión con los posibles servicios web a consumir.

0.2. OBJETIVOS

0.2.1. OBJETIVO GENERAL

Establecer una conexión en tiempo real entre el módulo ESP8266 NODEMCU, los servicios web y MySQL mediante la interfaz de arduino, para poder visualizar los datos e interactuar con los actuadores.

0.2.2. OBJETIVOS ESPECIFICOS

- Permitir la interacción entre clientes mediante el uso de servicios web.
- Interactuar con MySQL para lograr observar datos almacenados de los sensores y ser retransmitidos a actuadores.

0.3. ESTADO DEL ARTE

En el campo de la comunicación con los servicios web empleando hardware adicional como lo son módulos wi-fi, Arduino, etc., se a dispuesto diferentes aplicaciones que motivan la innovación permitiendo al usuario obtener accesos directo a su objetivo de búsqueda como es mencionado en el proyecto de grado “Diseño e implementación de un sistema de vigilancia remota para una residencia usando ESP8266 NODEMCU”, desarrollado por A. Soledad Zapata Y. y Andrés R. Vallejo P. en el año 2017 en el cual se plantea implementar un sistema que monitoree una vivienda a través del censo de 6 eventos intercambiando las muestras obtenidas mediante Wi-Fi usando el módulo ESP8266 NODEMCU hacia un servidor de base de datos. Por otra parte, podemos apreciar una aplicación más exterior en el trabajo de grado “Sistema de Medición Acústica usando NODEMCU ESP8266 para Determinar el Nivel de Ruido y guardarlos en una base de datos My SQL en Av. Víctor Larco cuadra 14 Trujillo 2018”, desarrollado por Br. Otiniano López, Mercedes Francisco en el año 2018 que describe una implementación más electrónica del módulo ESP8266 NODEMCU para la medición acústica del nivel de ruido con lo cual también se busca prever enfermedades en los habitantes del sector, El cual los datos almacenados se encuentran en la base de datos para de ese modo proporcionar en la hora de la medición. En comparación con nuestro objetivo los trabajos anteriormente mencionados presentan una misma vía de comunicación la cual es el módulo ESP8266 NODEMCU y a su vez el intercambio de información, sin embargo, podemos apreciar un enfoque más específico el cual es la vigilancia de una vivienda o la medición acústica de niveles de ruido, en cambio, en el presente trabajo buscamos obtener un intercambio de información más básico.

Basados en la idea de comunicarse con servicios web se debe tener conocimiento de que un servidor Web es un software diseñado para la transferencia de hipertextos, o páginas en lenguaje HTML, este servidor se encuentra todo el tiempo en espera de algún tipo de petición por parte de un cliente, el servidor envía ante la petición el código y el cliente es el encargado de interpretarlo como lo afirma el trabajo de grado “Implantación de los servicios web 2.0 para la página del Departamento de Eléctrica y Electrónica de la Escuela Politécnica del Ejercito”, desarrollado por Sr. Juan Felipe Calle Zhañay en el año 2011 en el cual se plante la configuración y el funcionamiento de un servidor web con aplicaciones Web 2.0 para de este modo potenciar la interacción existente entre los estudiantes del departamento y sus docentes. En comparación con nuestro proyecto podemos destacar los servicios web que se pretende ofrecer y hacer uso, aunque se mantiene una gran diferencia en el hardware de los mismos

0.4. MARCO TEÓRICO

0.4.1. INTERNET DE LAS COSAS (IOT)

El internet de las cosas (IOT) consiste en que diferentes cosas u objetos tengan la capacidad de conectarse a internet en cualquier momento y en cualquier lugar. En un sentido más técnico, consiste en la integración de sensores y dispositivos en objetos cotidianos que estén conectados a internet a través de redes fijas e inalámbricas. De esta manera, cualquier objeto es susceptible de ser conectado y manifestarse en la red. Además, el IOT implica que todo objeto puede ser una fuente de información. (Tojeiro, 2014). El internet de las cosas está presente en nuestro día a día. En nuestro entorno laboral, en nuestra casa, en la escuela, en el supermercado, etc. Un ejemplo de ello sería una máquina expendedora de refrescos que funciona con la publicación de un tweet, o con un hashtag en la página de la propia empresa o con un like en Facebook.

En definitiva, el objetivo ideal del IOT sería lograr que cualquier objeto tenga vida propia a través de internet y con ello una identidad. (Tojeiro, 2014)

Figura 1: INTERNET DE LAS COSAS



0.4.2. ESP8266 NODEMCU

Según (Naylampmechatronics, 2017) en su teoría indicó es un nuevo dispositivo orientado al internet de las cosas (IT) tiene un chip integrado llamado ESP8266 es muy pequeño siendo su ventaja principal y permite conectarme a internet vía Wi-Fi, teniendo un costo muy cómodo al alcance de todos, es parecido o similar al arduino que integra muchos componentes librerías, código, sensores, etc.

Figura 2: NODEMCU-ESP8266



Funcionamiento de Nodemcu Esp8266

Según (Naylampmechatronics, 2017) en su teoría indicó usando esta nueva tecnología que cuenta un chip integrado llamado ESP 8266 muy pequeño capaz de conectarme a internet vía Wi-Fi, usando datos del internet a través del celular, esta nueva tecnología permite registrar y guardar información en tiempo real siendo a una base de datos Mysql que está alojado en la nube conectándose con la página web (PHP) para ver los reportes de dicha medida, la unidad de medida se encuentra en decibeles.

Variantes

Como ya lo hemos dicho el ESP8266 es solo un procesador, pero su versión varía a la hora de construirlo sobre una placa impresa ya que sus características de construcción difieren en diferentes aspectos. Existen diferentes marcas fabricantes de estas excelentes variantes basadas en ESP8266. AI-Thinker la empresa China es una de las más importantes, con una extensa variedad de módulos de una excelente calidad a nivel global. Wemos (Compañía China) y Olimex (Europa) también aportan sus propias versiones. Las compañías Norte Americanas Adafruit y SparkFun no se pueden quedar atrás, fabricando dos modelos más de estas poderosas tarjetas Wi-fi. A continuación, revisaremos cada una de estas variantes de la global AI-Thinker.

Figura 3: VARIACIONES DE ESP

ESP-01:

- Dimensiones: 14,30 mm x 24,80 mm
- Conexiones: 8 patillas entre alimentación y GPIO
- Antena impresa en la PCB sin apantallar
- Alimentación: 3,3 V Para ser precisos, las versiones más nuevas incluyen el ESP8266EX y las primitivas el modelo inicial del ESP8266 (sin EX).

ESP-02

- Dimensiones: 14,20 mm x 14,20 mm
- Conexiones: 8 conexiones de superficie (es viable soldar patillas de 0,1 ")
- Sin antena en la placa, pero con un conector para antena externa sin apantallar
- Alimentación: 3,3 V

ESP-03

- Dimensiones: 17,30 mm x 12,10 mm
- Conexiones: 14 conexiones de superficie en los dos lados mayores
- Antena de tipo cerámico sin apantallar
- Alimentación: 3,3 V

ESP-04

- Dimensiones: 14,70 mm x 12,10 mm
- Conexiones: 14 conexiones de superficie en los dos lados mayores
- Sin antena Apantallado
- Alimentación: 3,3 V

ESP-05

- Dimensiones: 14,20 mm x 14,20 mm
- Conexiones: 8 patillas separadas una décima de pulgada en una única tira
- Sin antena en placa, con un conector para antena externa
- Apantallado
- Alimentación: 3,3 V

ESP-06

- Dimensiones: 14,20 mm x 14,70 mm
- Conexiones: 12 conexiones bajo la placa Sin antena
- Apantallado
- Alimentación: 3,3 V

NodeMCU

- Basado en ESP-12 Dimensiones: 30,85 mm x 47,35 mm
- Conexiones: 30 patillas separadas una décima de pulgada y USB
- Antena impresa en la PCB
- Apantallado
- Alimentación: 3,3 V y 5 V
- Pulsadores user y programación (flash)

ESP

- De ESP 07 a ESP 4
- Wroom
- 201

0.4.3. SERVICIOS WEB

Los servicios web son aplicaciones autónomas modulares que se pueden describir, publicar, localizar e invocar a través de una red.

El servidor de aplicaciones da soporte a los servicios web que se desarrollan e implementan de acuerdo con la especificación de servicios web para Java™ EE (Java Platform, Enterprise Edition). El servidor de aplicaciones da soporte a los modelos de programación JAX-WS (Java API for XML Web Services) y JAX-RPC (Java API for XML-based RPC). JAX-WS es un modelo de programación estratégico que simplifica el desarrollo de aplicaciones mediante el soporte de un modelo basado en anotaciones estándar para desarrollar clientes y aplicaciones de servicios web.

0.4.4. SPRING BOOT

Spring Boot es una de las tecnologías dentro del mundo de Spring de las que más se está hablando últimamente. ¿Qué es y cómo funciona Spring Boot? . Para entender el concepto primero debemos reflexionar sobre cómo construimos aplicaciones con Spring Framework Maven/Gradle y descargar las dependencias necesarias. En segundo lugar desarrollamos la aplicación y en tercer lugar la desplegamos en un servidor. Si nos ponemos a pensar un poco a detalle en el tema , únicamente el paso dos es una tarea de desarrollo. Los otros pasos están más orientados a infraestructura.

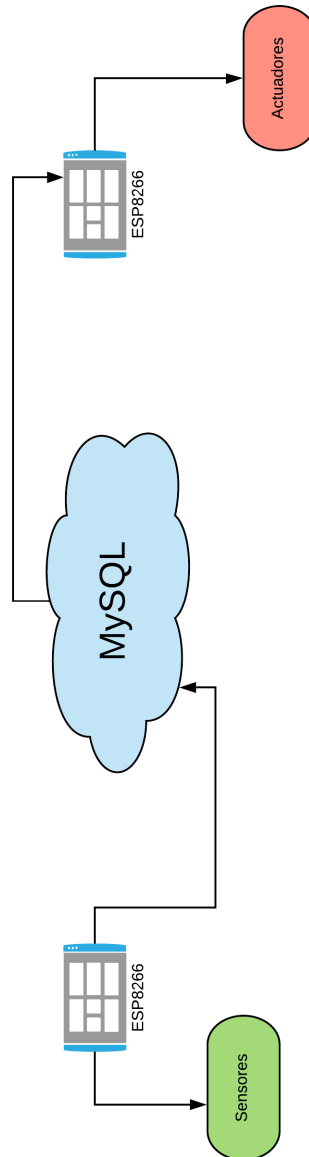
0.4.5. MySQL

MySQL es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle. Actualmente, es la base de datos de código abierto más famosa y utilizada en el mundo entero.

MySQL sirve para almacenar toda la información que se desee en bases de datos relacionales, como también para administrar todos estos datos sin apenas complicaciones gracias a su interfaz visual y a todas las opciones y herramientas de las que dispone. Es algo esencial, sobre todo en webs que cuentan con la opción de registrar usuarios para que inicien sesión.

0.5. DIAGRAMAS

Figura 4: Diagrama



en blanco.png

0.6. LISTA DE COMPONENTES

■ PROGRAMA ARDUINO IDE

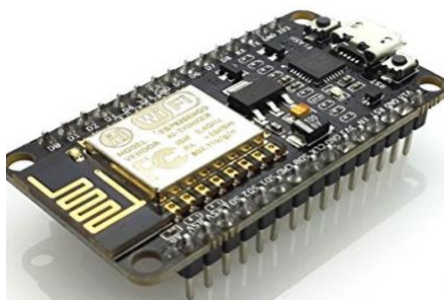
Entorno de desarrollo integrado, llamado IDE (sigla en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

Figura 5: PROGRAMA ARDUINO IDE



■ ESP8266 NODEMCU

Figura 6: NODEMCU-ESP8266



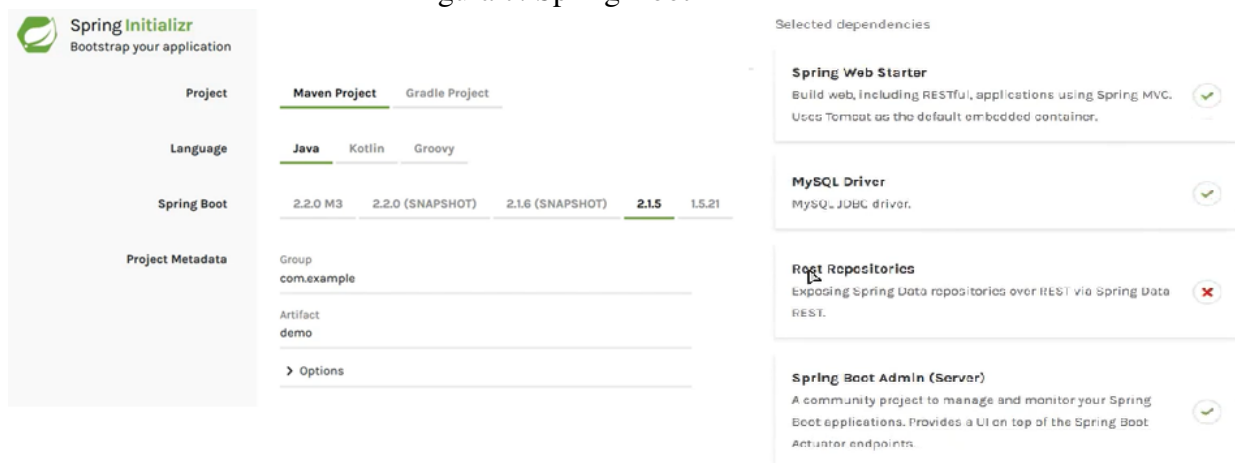
0.7. MAPA DE VARIABLES

Variables Globales			Variable Temporal	
	ssid	password	host	line
	ESPE	****	www.vermiiip.es	Monitor Serial
1	ESPE	****	www.vermiiip.es connected
2	ESPE	****	www.vermiiip.es	[Conectando a www.vermiiip.es ... conectado]
3	ESPE	****	www.vermiiip.es	[Enviando petición]
4	ESPE	****	www.vermiiip.es	[Respuesta:]
5-717	ESPE	****	www.vermiiip.es	
718	ESPE	****	www.vermiiip.es	HTTP/1.1 200 OK
719	ESPE	****	www.vermiiip.es	Location: https://ufasipontise.espe.edu.ec:8443/portal/gateway?sessionId=faf00010a001e7b5462f1c15c&portal=af43af030-5a11-11e8-ac7e-42dbec302e0d&action=cwa&token=02f4b44a2d40434339dec8643a8b61d0f8redirect=www.vermiiip.es/sf
720	ESPE	****	www.vermiiip.es	Content-Type: text/html
721	ESPE	****	www.vermiiip.es	Content-Length: 459
722	ESPE	****	www.vermiiip.es	
723	ESPE	****	www.vermiiip.es	<HTML><HEAD><TITLE> Web Authentication Redirect</TITLE><META http-equiv="Cache-control" content="no-cache"><META http-equiv="Pragma" content="no-cache"><META http-equiv="Expires" content="1"><META http-equiv="refresh" content="1"; URL=https://ufasipontise.espe.edu.ec:8443/portal/gateway?sessionId=faf00010a001e7b5462f1c15c&portal=af43af030-5a11-11e8-ac7e-42dbec302e0d&action=cwa&token=02f4b44a2d40434339dec8643a8b61d0f8redirect=www.vermiiip.es/sf></HEAD></HTML>
724	ESPE	****	www.vermiiip.es	
725	ESPE	****	www.vermiiip.es	[Desconectado]

0.8. EXPLICACIÓN DE CÓDIGO FUENTE

Iniciamos con crear el archivo en Spring Initializr con sus respectivas dependencias mostradas en la Figura 7.

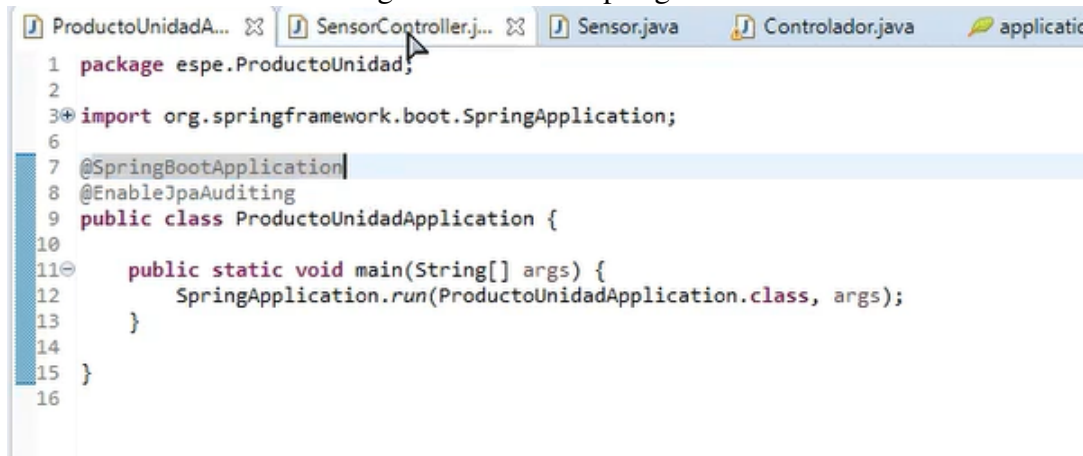
Figura 7: Spring Boot



boot.png

Descargamos el archivo lo agregamos al workSpace de Eclipse para poder trabajar en este, procedemos a abrirlo y agregamos la libreria que se encuentre mostrada en la Figura 8

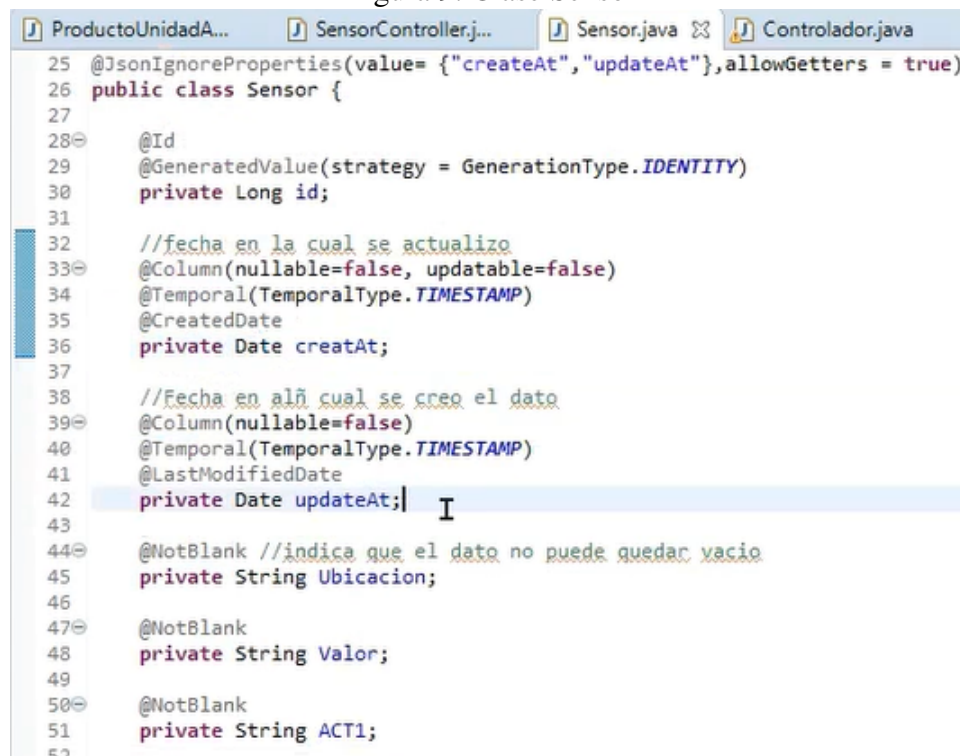
Figura 8: Libreria Spring Boot



Procedemos a programar en la clase sensor todos los atributos que esta va a tener.

En la Clase controlador de sensor añadimos la etiqueta de @RestController y especificamos el dominio en el que va a estar como se muestra en la Figura 10.

Figura 9: Clase Sensor



```

25 @JsonIgnoreProperties(value= {"createAt", "updateAt"}, allowGetters = true)
26 public class Sensor {
27
28     @Id
29     @GeneratedValue(strategy = GenerationType.IDENTITY)
30     private Long id;
31
32     //fecha en la cual se actualizo
33     @Column(nullable=false, updatable=false)
34     @Temporal(TemporalType.TIMESTAMP)
35     @CreatedDate
36     private Date creatAt;
37
38     //Fecha en la cual se creo el dato
39     @Column(nullable=false)
40     @Temporal(TemporalType.TIMESTAMP)
41     @LastModifiedDate
42     private Date updateAt;
43
44     @NotBlank //indica que el dato no puede quedar vacio
45     private String Ubicacion;
46
47     @NotBlank
48     private String Valor;
49
50     @NotBlank
51     private String ACT1;
52

```

Figura 10: Clase Sensor Controller



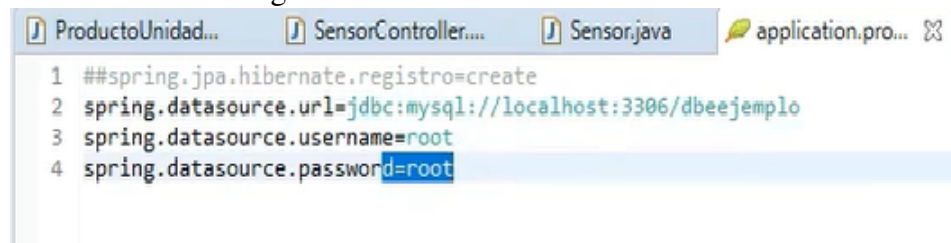
```

1 package espe.ProductoUnidad.controlador;
2
3 import java.util.List;
4
21
22 @RestController
23 @RequestMapping(path="/ProductoUnidad")
24
25 public class SensorController {
26
27     @Autowired //identifica el nombre del repositorio
28     SensorRepository sensorRepository;
29
30     @PostMapping(path="/sensor")
31     public Sensor createNote(@Valid @RequestBody Sensor sensor) {
32         return sensorRepository.save(sensor);
33     }
34
35     @GetMapping(path="/sensor")
36     public List<Sensor> getAll(){
37         return sensorRepository.findAll();
38     }
39
40     @GetMapping(path="/s")
41     public @ResponseBody String hola() {
42
43         return "Si Funciona ";
44     }
45

```

Procedemos a especificar la conexión a la base de datos como se muestra en la Figura 11, Especificando el usuario y la contraseña.

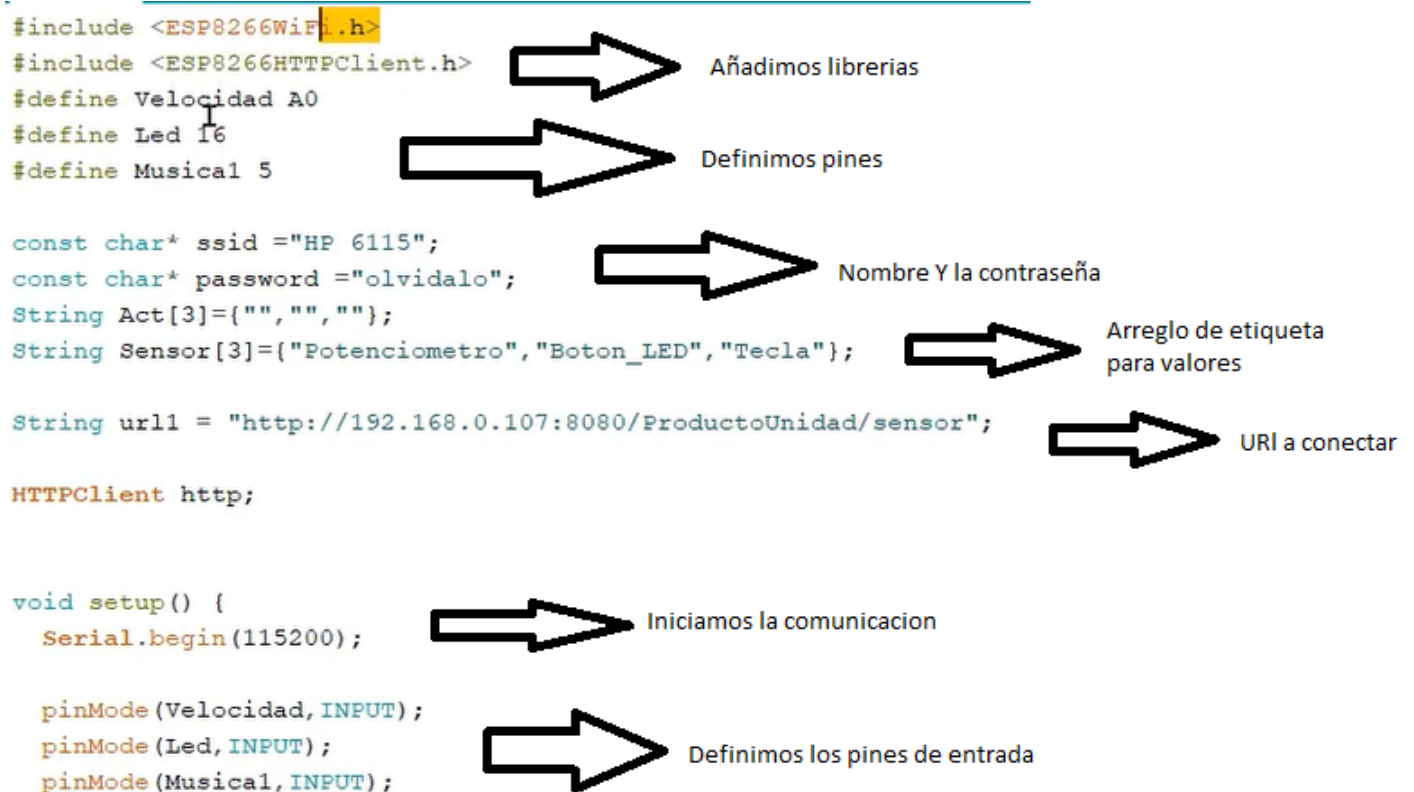
Figura 11: Conexión con base de datos



```

1 ##spring.jpa.hibernate.registro=create
2 spring.datasource.url=jdbc:mysql://localhost:3306/dbeejemplo
3 spring.datasource.username=root
4 spring.datasource.password=root
  
```

Figura 12: Descripción de Arduino



```

#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#define Velocidad A0
#define Led 16
#define Musical 5

const char* ssid = "HP 6115";
const char* password = "olvidalo";
String Act[3]={"", "", ""};
String Sensor[3]={"Potenciometro", "Boton_LED", "Tecla"};

String url1 = "http://192.168.0.107:8080/ProductoUnidad/sensor";

HTTPClient http;

void setup() {
  Serial.begin(115200);

  pinMode(Velocidad, INPUT);
  pinMode(Led, INPUT);
  pinMode(Musical, INPUT);
  
```

Annotations:

- Añadimos librerías (pointing to the include statements)
- Definimos pines (pointing to the pin definitions)
- Nombre Y la contraseña (pointing to the SSID and password)
- Arreglo de etiqueta para valores (pointing to the Sensor array)
- URI a conectar (pointing to the url1 variable)
- Iniciamos la comunicacion (pointing to Serial.begin)
- Definimos los pines de entrada (pointing to the pinMode statements)

0.9. DESCRIPCIÓN DE PRERREQUISITOS Y CONFIGURACIÓN

Uno de los prerrequisitos importantes para el presente trabajo es la configuración del entorno de trabajo **Arduino IDE** para poder hacer uso de la placa ESP8266 NodeMCU, para lo cual se debe seguir los siguientes pasos:

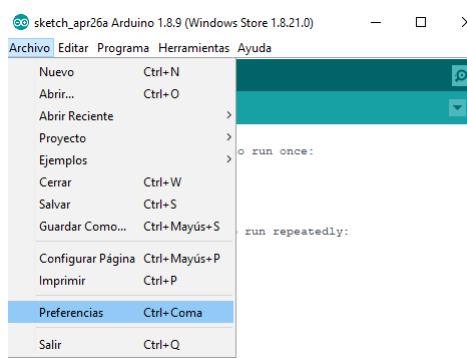
- Abrir el programa Arduino IDE

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

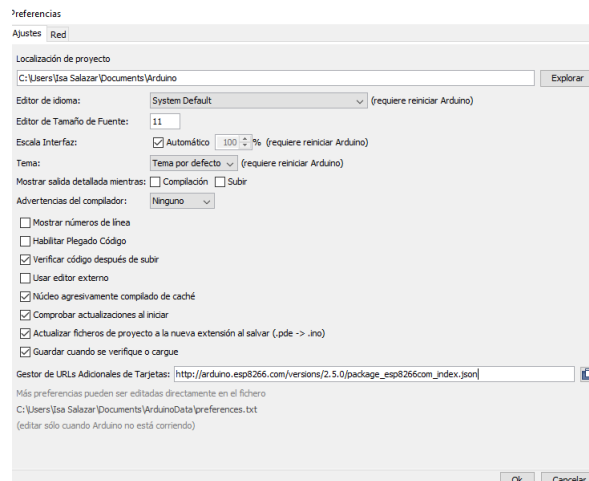
Figura 13: INICIO DE ARDUINO IDE



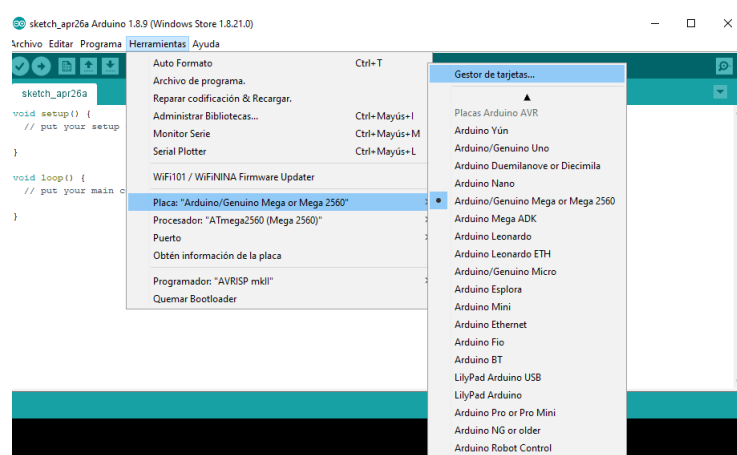
- Una vez en la hoja de trabajo se procede a dar click en **Archivo**, ubicado en la parte superior derecha de la hoja de trabajo, se despliega un menú y procedemos a dar click en **Preferencias**.



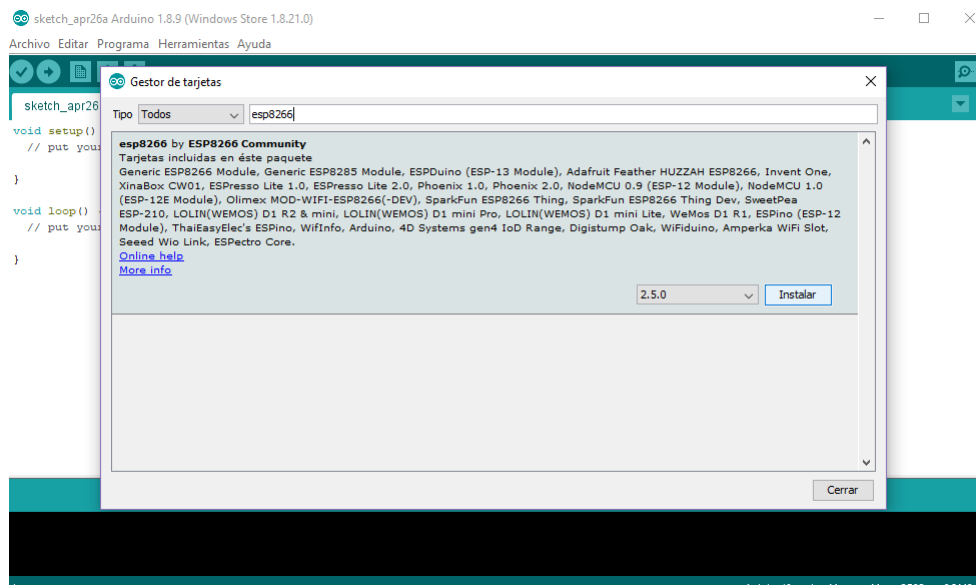
- En **Gestor de URLs Adicionales de Tarjetas** es necesario poner la siguiente url: `http://arduino.esp8266.com/stable/package_esp8266com_index.json` y damos click en **OK**. Esto nos va a permitir tener acceso a diferentes modelos de placas.



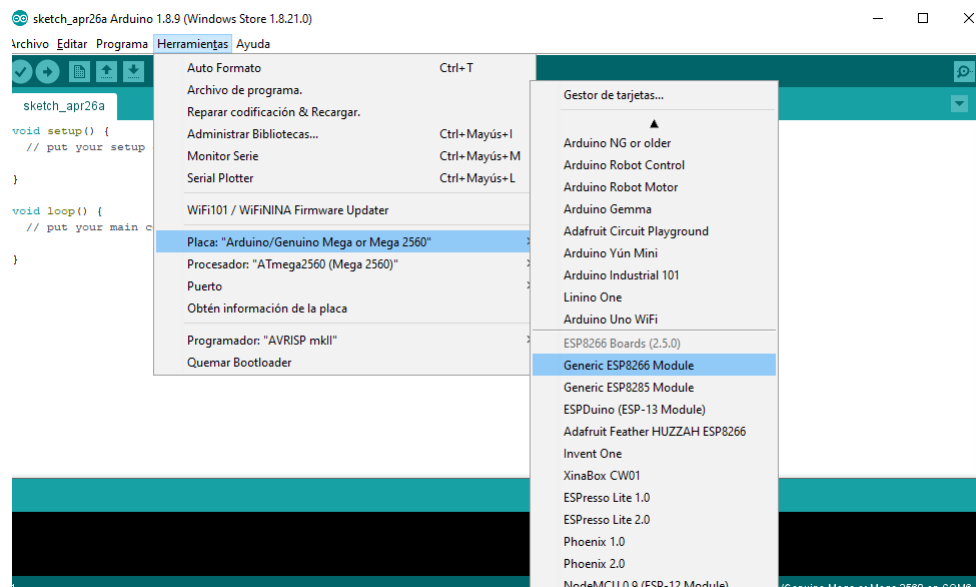
- De regreso en la hoja de trabajo damos click en **Herramientas**, en el menú que se despliega damos click en **Placa: xxxxxxxxxxxxxx** y se vuelve a desplegar otro menú en el cual damos click en **Gestor de Tarjetas**.



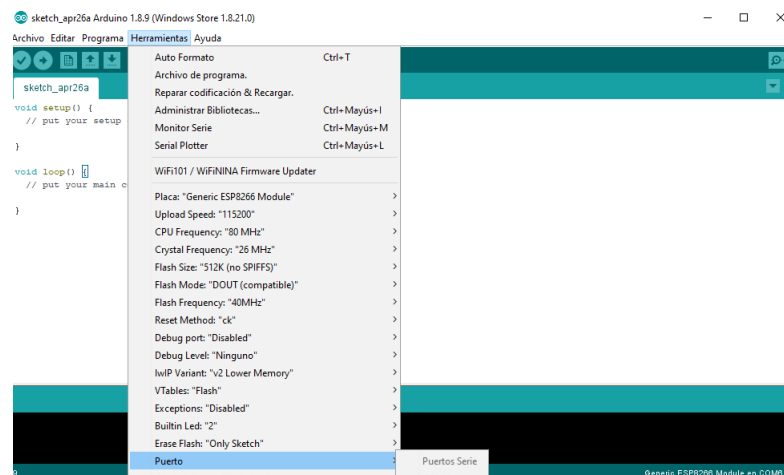
- En la ventana que se despliega buscamos **esp8266** y procedemos a instalar la versión **Community** que esté disponible para usted.



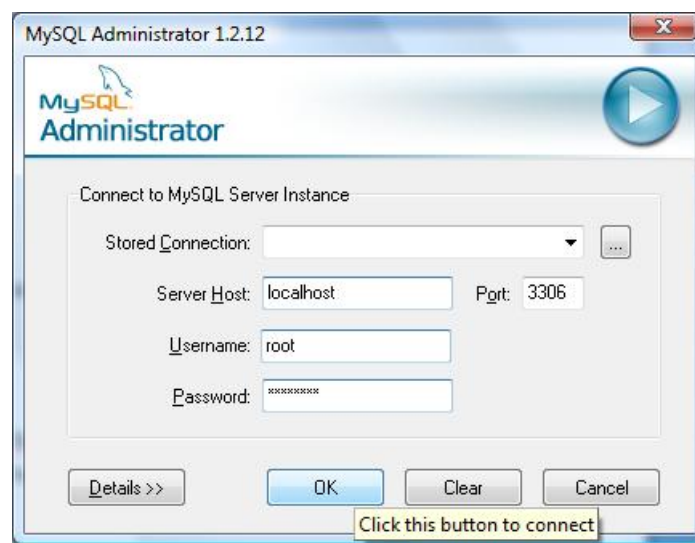
- En el mismo menú **Herramientas** damos click en **Placa: xxxxxxxxxxxxxx** y ibservamos que se nos despliega un menú más extenso con diferentes tios de placas **esp**. Escogemos **Generic ESP8266 Module**.



- Continuando en el menú **Herramientas** al conectar la placa podemos observar que se muestra mas opciones de configuración e incluso el puerto en el que se creo la placa.



Se debe tener muy claro el usuario y contraseña para el ingreso a MySQL, tambien para lograr la programación para el ingreso a esta.



0.10. CONCLUSIONES

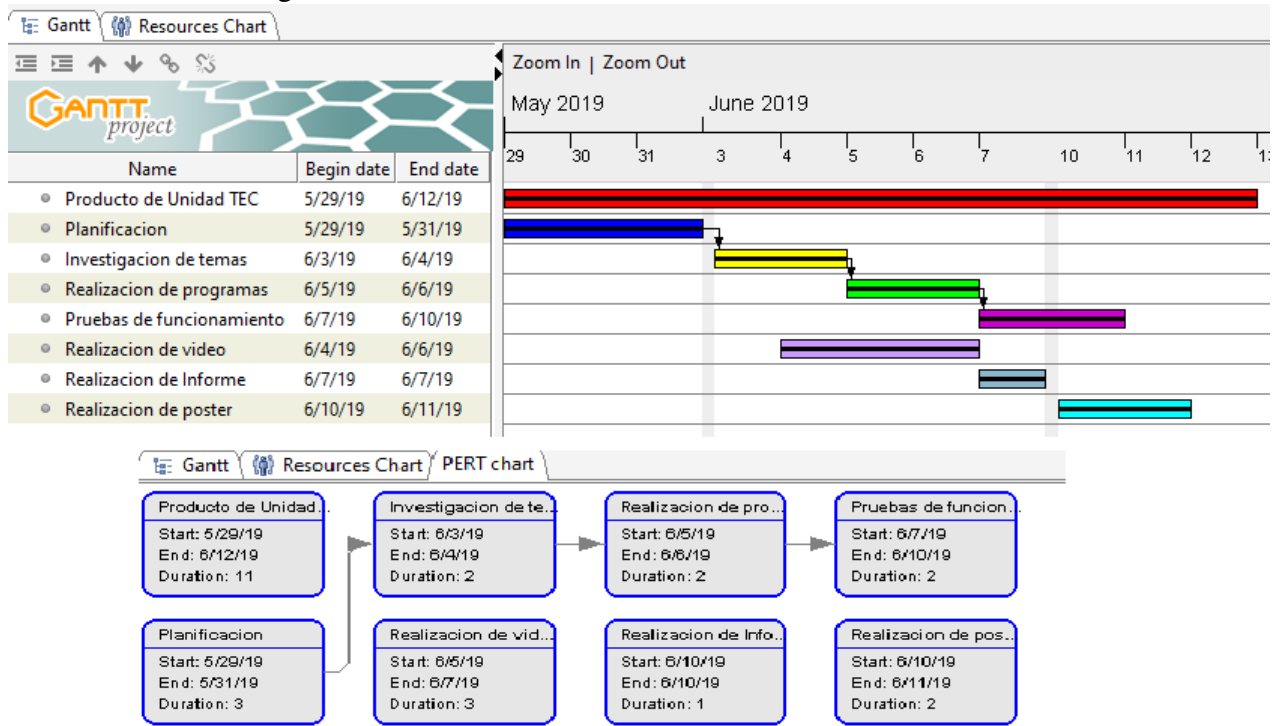
- Se estableció una conexión en tiempo real entre el modulo ESP8266 y MySQL, para poder visualizar en los actuadores.
- Se permitió la interacción entre los sensores y los actuadores mediante MySQL
- Se interactuó con MySQL, logrando observar los datos almacenados.

0.11. RECOMENDACIONES

- En el entorno de programación utilizado se recomienda comentar el código debido a que si en algún momento se necesita editar el mismo se pueda identificar la parte por cambiar y ubicar correctamente el funcionamiento.
- Tener muy en cuenta que el modulo establezca correctamente la conexión con los servicios web, de ese modo evitar no lograr captar correctamente los datos.
- El uso de MySQL hay q realizarlo con precaucion, ya que debido si no se realiza la conexión correcto no funcionaria, ademas que presenta vulnerabilidades en la seguridad.

0.12. CRONOGRAMA

Figura 14: CRONOGRAMA DE PROYECTO DE UNIDAD



0.13. LINKS DE OVERLEAF Y GITHUB

<https://www.overleaf.com/9211823427qtkqbwhfrmk>

<https://github.com/samorales6/ProductoDeUnidad2>

0.14. BIBLIOGRAFIA

- Rodrigues, C. M., & Castro, B. S. A Vision of Internet of Things in Industry 4.0 with ESP8266. International Journal of Electronics and Communication Engineering and Technology, 9(1), 20.
- Otiniano López, M. F. (2018). Sistema de Medición Acústica usando NODEMCU ESP8266 para Determinar el Nivel de Ruido en Av. Víctor Larco cuadra 14 Trujillo 2018.
- Candelario Elías, J. (2016). Implementación de WPS en el firmware NodeMCU para el ESP8266.
- Irigoyen Gallego, R. (2018). Internet de las cosas. Sistema electrónico de control basado en Arduino (Doctoral dissertation).
- Calle Zhañay, Juan Felipe (2017). Implantación de los servicios web 2.0 para la página del Departamento de Eléctrica y Electrónica de la Escuela Politécnica del Ejército. Carrera de Ingeniería en Electrónica, Redes y Comunicación de Datos. ESPE. Sede Sangolquí.
- Aguirre Rojas, Marco Esteban (2018). Emulador a escala de un sistema remoto de conducción vehicular terrestre mediante la transferencia de su dinámica. Carrera de Ingeniería en Electrónica y Telecomunicaciones. Universidad de las Fuerzas Armadas ESPE. Matriz Sangolquí.

0.15. MANUAL DE USUARIO

Configuración del Arduino IDE

La configuración del entorno de trabajo **Arduino IDE** para poder hacer uso de la placa ESP8266 No-deMCU, para lo cual se debe seguir los siguientes pasos:

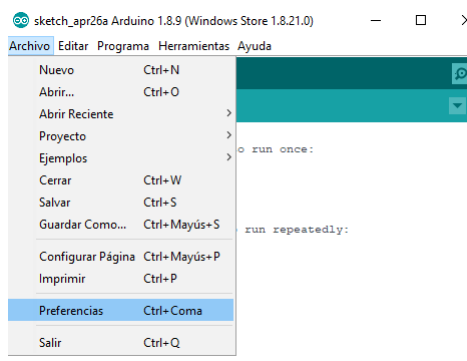
- Abrir el programa Arduino IDE

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

Figura 15: INICIO DE ARDUINO IDE

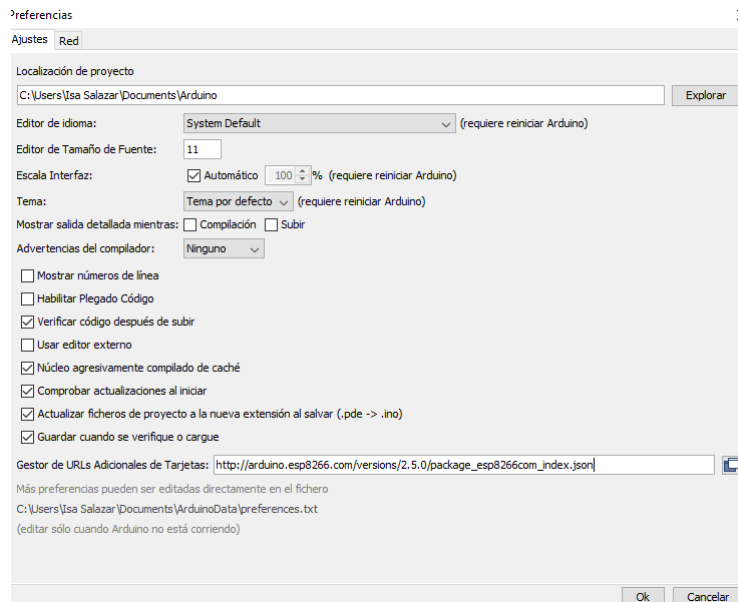


- Una vez en la hoja de trabajo se procede a dar click en **Archivo**, ubicado en la parte superior derecha de la hoja de trabajo, se despliega un menú y procedemos a dar click en **Preferencias**.

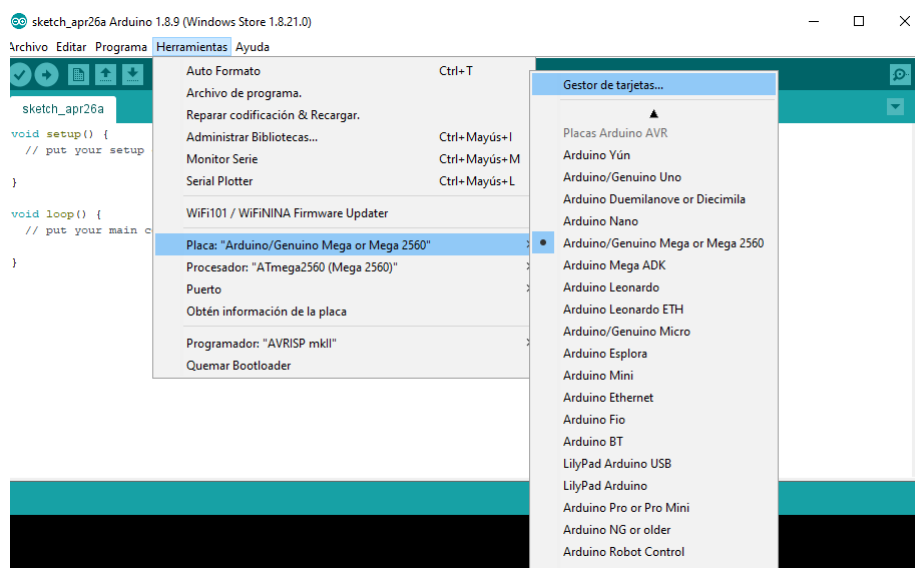


- En **Gestor de URLs Adicionales de Tarjetas** es necesario poner la siguiente url: <http://>

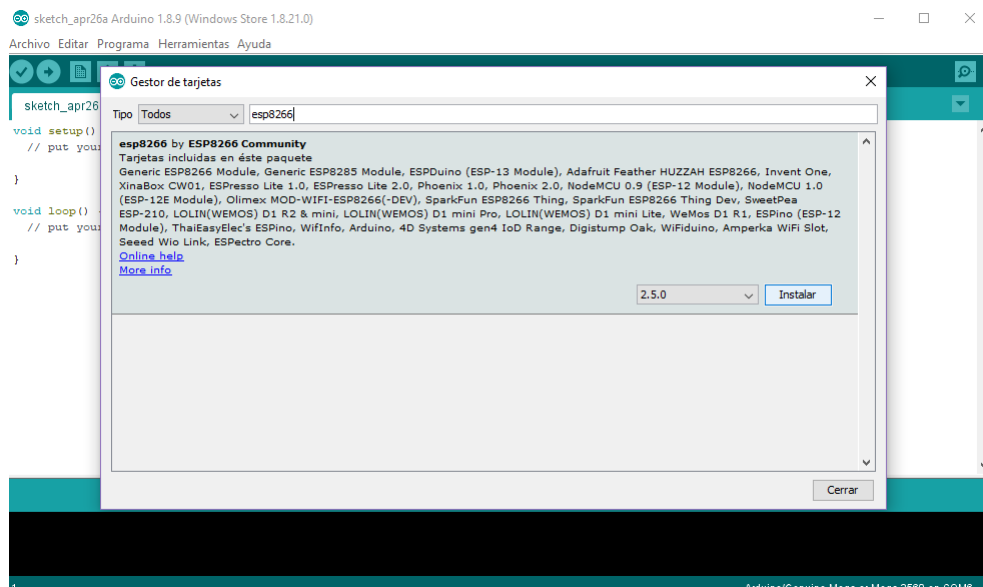
arduino.esp8266.com/stable/package_esp8266com_index.json y damos click en "OK". Este nos va a permitir tener acceso a diferentes modelos de placas.



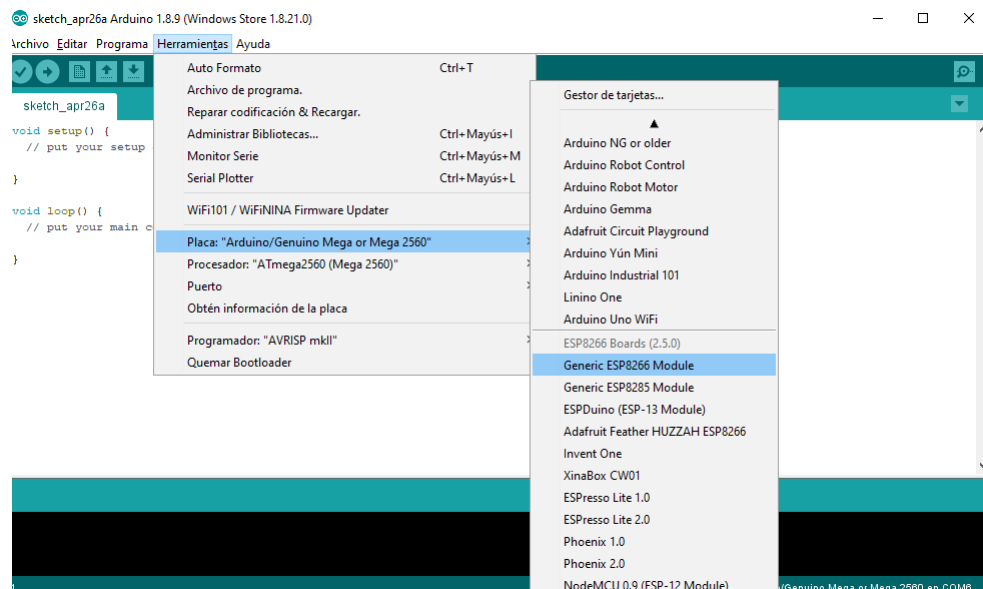
- De regreso en la hoja de trabajo damos click en **Herramientas**, en el menú que se despliega damos click en **Placa: xxxxxxxxxxxxxx** y se vuelve a desplegar otro menú en el cual damos click en **Gestor de Tarjetas**.



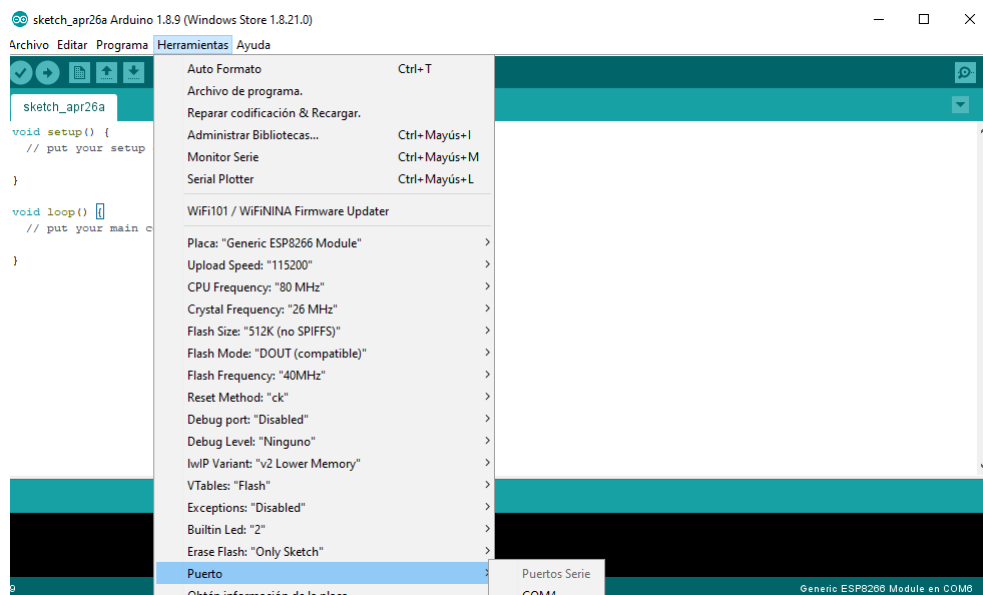
- En la ventana que se despliega buscamos **esp8266** y procedemos a instalar la versión **Community** que esté disponible para usted.



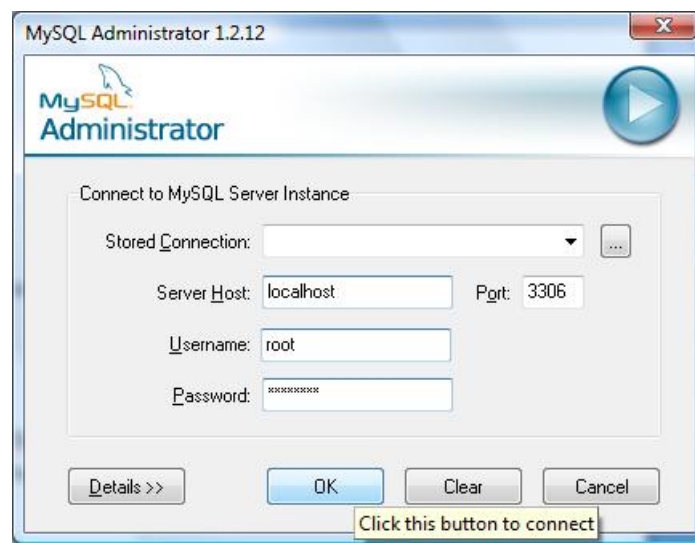
- En el mismo menú **Herramientas** damos click en **Placa: xxxxxxxxxxxxxx** y ibservamos que se nos despliega un menú más extenso con diferentes tios de placas **esp**. Escogemos **Generic ESP8266 Module**.



- Continuando en el menú **Herramientas** al conectar la placa podemos observar que se muestra mas opciones de configuración e incluso el puerto en el que se creo la placa.



Ingresamos sesion con nuestro usuario y contraseña en MySQL



0.16. HOJAS TÉCNICAS

Figura 16: ESPECIFICACIONES ESP8266

1.2. Specifications

Table 1-1. Specifications

Categories	Items	Parameters
Wi-Fi	Certification	Wi-Fi Alliance
	Protocols	802.11 b/g/n (HT20)
	Frequency Range	2.4G ~ 2.5G (2400M ~ 2483.5M)
	TX Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
		802.11 n: -72 dbm (MCS7)
	Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware	CPU	Tensilica L106 32-bit processor
	Peripheral Interface	UART/SPIO/SPi/I2C/I2S/IR Remote Control
		GPIO/ADC/PWM/LED Light & Button
	Operating Voltage	2.5V ~ 3.6V
	Operating Current	Average value: 80 mA
	Operating Temperature Range	-40°C ~ 125°C
	Package Size	QFN32-pin (5 mm x 5 mm)
Software	External Interface	-
	Wi-Fi Mode	Station/SoftAP/SoftAP+Station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming
	Network Protocols	IPv4, TCP/UDP/HTTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

Note:
The TX power can be configured based on the actual user scenarios.

Figura 17: DIAGRAMA PLACA ESP8266

