

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE TECNOLOGÍA
CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES



**“DISEÑO DE UN PROTOTIPO ELECTRÓNICO DE
MEDICIÓN, CONTROL Y REGISTRO DE CONSUMO DE
AGUA POTABLE PARA APLICACIÓN DOMÉSTICA”**

Proyecto de Grado presentado para obtener el grado de Licenciatura

POSTULANTE: OSCAR RAFAEL RIVAS FLORES

TUTOR: ING. JOSÉ ARTURO MARÍN THAMES

LA PAZ – BOLIVIA

2019

DEDICATORIA

*Dedicado a Dios y a mis padres quienes me apoyaron en todo momento,
por todo su amor y preocupación.*

AGRADECIMIENTOS

Mis agradecimientos:

A Dios quien me concedió la vida, por darme sabiduría y brindarme la oportunidad de llegar a esta ocasión.

A mis padres y hermanos por su apoyo incondicional.

A las personas que me apoyaron.

INDICE DE CONTENIDO

CAPITULO I: ANTECEDENTES DEL PROYECTO	1
1.1 ANTECEDENTES	2
1.2 PLANTEAMIENTO DEL PROBLEMA	2
1.3 OBJETIVOS	3
1.3.1 OBJETIVO GENERAL.....	3
1.3.2 OBJETIVOS ESPECÍFICOS.....	3
1.4 JUSTIFICACIÓN	3
1.4.1 JUSTIFICACIÓN ACADEMICA.....	3
1.4.2 JUSTIFICACIÓN TÉCNICA.....	4
1.4.3 JUSTIFICACIÓN SOCIAL.....	4
1.5 LIMITES Y ALCANCES	4
1.5.1 LIMITES.....	4
1.5.2 ALCANCES.....	4
1.6 METODOLOGÍA	5
CAPITULO II: MARCO TEÓRICO REFERENCIAL	6
2.1 SISTEMAS EMBEBIDOS	7
2.2 PLACA DE DESARROLLO NODEMCU BASADO EN ESP8266	8
2.2.1 CONOCIENDO AL ESP8266.....	8
2.2.1.1 POSIBLES USOS DEL ESP8266.....	9
2.2.1.2 ESPECIFICACIONES DEL MICROCONTROLADOR ESP8266.....	10
2.2.1.3 PROCESADOR.....	11
2.2.1.4 ARQUITECTURA.....	11
2.2.1.5 VARIANTES Y MODELOS DEL ESP8266.....	12
2.2.2 PLACA DE DESARROLLO NODEMCU.....	15
2.2.2.1 VERSIONES DE NODEMCU.....	18
2.2.2.2 DESCRIPCIÓN DE CADA UNO DE LOS PINES NODEMCU ESP12.....	21
2.2.3 ENTORNO DE PROGRAMACIÓN PARA NODEMCU CON "ARDUINO IDE".....	22
2.2.3.1 ESTRUCTURA DE UN PROGRAMA EN LA IDE ARDUINO.....	24

2.3	MEDIDORES DE FLUJO	27
2.3.1	CLASIFICACIÓN DE CAUDALIMETROS.....	28
2.3.1.1	CAUDALIMETRO ELECTRÓNICO DE TURBINA.....	28
2.3.1.2	CAUDALIMETRO MECÁNICO DE MOLINO.....	29
2.3.1.3	CAUDALIMETRO ELECTROMAGNÉTICO.....	29
2.4	SERVIDOR WEB LAMP	31
2.4.1	APACHE.....	32
2.4.2	MYSQL.....	33
2.4.3	LENGUAJE DE PROGRAMACIÓN PHP.....	34
2.4.4	GESTOR DE BASE DE DATOS phpMyAdmin.....	35
2.4.5	MODELO ENTIDAD RELACION DE UNA BASE DE DATOS.....	37
2.4.5.1	ELEMENTOS DEL MODELO ENTIDAD RELACIÓN.....	37
2.4.6	LEGUAJES DE PROGRAMACIÓN PRINCIPALES PARA EL FRONT- END.....	40
2.4.6.1	HTML5.....	40
2.4.6.2	CSS.....	41
2.4.6.3	JAVASCRIPT.....	42
2.5	SERVIDOR PRIVADO VIRTUAL VPS	43
2.5.1	PROVEEDOR DE VPS “DigitalOcean”.....	44
2.5.1.1	RECURSOS DEL SERVIDOR.....	45
 CAPITULO III: INGENIERÍA DE PROYECTO		46
3.1	INTRODUCCIÓN	47
3.2	PROGRAMACIÓN DEL HARDWARE	48
3.2.1	INTSLACIÓN DE LA LIBRERÍA DEL ESP8266 PARA ARDUINO.....	48
3.2.2	DETALLES TÉCNICOS DEL CAUDALÍMETRO.....	50
3.2.3	CALCULO DEL CAUDAL Y VOLUMEN.....	52
3.3	IMPLEMENTACIÓN DE LA BASE DE DATOS	61
3.3.1	MODELO ENTIDAD-RELACION DE LA BASE DE DATOS.....	63
3.4	RECEPCION Y ALAMCENAMIENTO EN LA BASE DE DATOS (enviardatos.php)	64
3.5	INTERFAZ WEB DEL USUARIO	67
3.5.1	REGISTRO DE NUEVO USUARIO.....	70
3.5.2	REGISTRO DE NUEVO DISPOSITIVO.....	70
3.5.3	VERIFICACIÓN DE DATOS ALAMACENADOS DE USUARIO Y DISPOSITIVO EN LA BASE DE DATOS.....	71

3.5.4	REPORTE DEL CALCULO DE CONSUMO DE AGUA Y EL TOTAL A PAGAR.....	72
3.5.5	CONSIDERACIONES PARA EL CÁLCULO DEL MONTO A PAGAR SEGÚN EL CONSUMO REALIZADO.....	74
3.6	INSTALACIÓN DEL SERVIDOR WEB EN DIGITALOCEAN.....	76
3.6.1	CREACIÓN DEL DROPELT.....	76
3.6.2	INSTALACIÓN DE LA PILA LAMP EN EL DROPLET.....	80
3.6.2.1	INSTALACIÓN DE APACHE	81
3.6.2.2	INSTALACION MySQL.....	83
3.6.2.3	INSTALACIÓN PHP.....	84
3.6.2.4	INSTLACION DE PHPMyADMIN.....	84
3.6.3	SUBIENDO LOS ARCHIVOS AL SERVIDOR.....	88
3.7	PRUEBA DE FUNCIONAMIENTO HARDWARE Y SOFTWARE.....	90
3.7.1	DIAGRAMA DE CONEXIÓN HARDWARE.	90
3.7.1.1	IMPLANTACIÓN DEL CIRCUITO EN PCB.....	90
3.7.2	INSTALACIÓN DEL HARDWARE.....	92
3.7.3	ACCESO Y PRUEBA DE LA INTERFAZ WEB DEL USUARIO.....	95
 CAPITULO IV: ANÁLISIS DE COSTOS.....		101
4.1	INTRODUCCIÓN.....	102
4.2	COSTOS FIJOS.....	102
4.3	COSTOS VARIABLES.....	103
4.4	COSTO TOTAL.....	103
 CAPITULO V: CONCLUSIONES Y RECOMENDACIONES.....		104
5.1	CONCLUSIONES.....	105
5.2	RECOMENDACIONES.....	105
 REFERENCIAS BIBLIOGRÁFICAS.....		107
 GLOSARIO.....		109
 ANEXOS.....		111

INDICE DE FIGURAS

Figura N°1. Arquitectura típica de un sistema embebido.....	8
Figura N°2. Procesador ESP8266EX.....	11
Figura N°3. ESP8266 modelo ESP-01.....	12
Figura N°4. ESP8266 modelo ESP-02.....	13
Figura N°5. ESP8266 modelo ESP-05.....	14
Figura N°6. ESP8266 modelo ESP-12.....	15
Figura N°7. Esquema general de placas de desarrollo.....	15
Figura N°8. Kit o placas de desarrollo.....	16
Figura N°9. NodeMCU V0.9 / V1.....	19
Figura N°10. NodeMCU V1.0 / V2.....	20
Figura N°11. NodeMCU V1.0 / V3.....	20
Figura N°12. Pinaje de NodeMCU.....	21
Figura N°13. Software Arduino IDE.....	23
Figura N°14. Estructura de un programa.....	24
Figura N°15. Modelos caudalímetros electromagnéticos.....	31
Figura N°16. Página web determinada de apache para ubuntu.....	33
Figura N°17. Instalación MySQL.....	35
Figura N°18. Instalación MySQL.....	36
Figura N°19. Interfaz de autenticación PHPMyAdmin.....	36
Figura N°20. Interfaz PHPMyAdmin.....	37
Figura N°21. Esquema de relación 1 a 1.....	39
Figura N°22. Esquema de relación 1 a M.....	39
Figura N°23. Esquema de relación M a M.....	40
Figura N°24. Elementos de una Entidad-Relación.....	40
Figura N°25. Estándar de una página web dinámica e interactiva.....	43
Figura N°26. Planes que ofrece DigitalOcean.....	45
Figura N°27. Diagrama de bloques del sistema.....	47
Figura N°28. Instalación librería del esp8266.....	48
Figura N°29. Instalación librería del esp8266.....	49
Figura N°30. Selección de la placa NodeMCU.....	49
Figura N°31. Caudalímetro FS300A.....	50
Figura N°32. Caudalímetro YF-S201.....	51
Figura N°33. Número de pulsos por litro del modelo YF-S201.....	53
Figura N°34. Número de pulsos por litro del modelo FS300A.....	54

Figura N°35. Muestra del caudal y volumen acumulado.....	57
Figura N°36. Muestra del caudal y volumen acumulado.....	57
Figura N°37. Creación de una nueva Base de Datos.....	61
Figura N°38. Creación de la tabla registro.....	62
Figura N°39. Creación de la tabla usuario.....	62
Figura N°40. Creación de la tabla chip.....	63
Figura N°42. Conexión a la red Wi-Fi.....	65
Figura N°43. Registro y envío de datos.....	65
Figura N°44. Registro y envío de datos.....	66
Figura N°45. Datos registrados en la base de datos.....	66
Figura N°46. Login.....	67
Figura N°47. Diagrama de flujo para el Login.....	68
Figura N°48. Vista de la interfaz web del usuario como administrador.....	69
Figura N°49. Registro de nuevo usuario.....	70
Figura N°50. Registro de nuevo dispositivo.....	71
Figura N°51. Datos nuevo usuario almacenado.....	71
Figura N°52. Datos nuevo dispositivo almacenado.....	72
Figura N°53. Reporte consumo de agua.....	73
Figura N°53. Reporte consumo de agua.....	73
Figura N°55. Inicio de cesión en DigitalOcean.....	76
Figura N°56. Creación nuevo Droplet.....	77
Figura N°57. Características nuevo Droplet.....	77
Figura N°58. Selección de la región.....	78
Figura N°59. Nombre de Droplet.....	78
Figura N°60. IP asignada de nuevo Droplet.....	79
Figura N°61. Encender y apagar el Droplet.....	79
Figura N°62. Actualización de repositorio.....	80
Figura N°63. Comando instalación de Apache.....	81
Figura N°64. Apache instalado.....	81
Figura N°65. Página web de apache de instalación satisfactoria.....	82
Figura N°66. Comando instalación de Apache.....	83
Figura N°67. MySQL instalado.....	83
Figura N°68. PHP instalado.....	84
Figura N°69. Comando instalación de phpMyAdmin.....	85
Figura N°70. Selección de servidor apache2.....	85
Figura N°71. Selección de servidor apache2.....	86
Figura N°72. Confirmación contraseña.....	86

Figura N°73. Acceso a phpMyAdmin.....	87
Figura N°74. Interfaz web phpMyAdmin.....	87
Figura N°75. Inicio de WinSCP con la IP asignada.....	88
Figura N°76. Conexión al servidor.....	88
Figura N°77. Archivos cargados en el servidor.....	89
Figura N°78. Base de datos cargada en phpMyAdmin.....	89
Figura N°79. Diagrama de conexión del circuito	90
Figura N°80. Diseño para placa PCB.....	90
Figura N°81. Circuito impreso PCB.....	90
Figura N°82. Circuito implementado en PCB cara inferior	91
Figura N°83. Circuito implementado en PCB cara superior.....	91
Figura N°84. Diseño final del circuito en PCB.....	91
Figura N°85. Diseño final del circuito en PCB.....	91
Figura N°86. Caudalímetro de 1/2" instalado en tubería.....	92
Figura N°87. Caudalímetro de 1/2" instalado en tubería.....	92
Figura N°88. Conexión tarjeta de desarrollo Esp8266NodeMCU.....	93
Figura N°89. Caudalímetro de 3/4" instalado en tubería.....	93
Figura N°90. Caudalímetro de 3/4" instalado en tubería.....	94
Figura N°91. Conexión tarjeta de desarrollo Esp8266NodeMCU.....	94
Figura N°92. Acceso a la interfaz web como usuario.....	95
Figura N°93. Pantalla principal de la interfaz web.....	96
Figura N°94. Reporte de consumo.....	97
Figura N°95. Datos de factura de agua.....	97
Figura N°96. Reporte de consumo.....	98
Figura N°97. Reporte final de consumo.....	97
Figura N°98. Datos de factura de agua.....	99
Figura N°99. Reporte de consumo.....	99
Figura N°100. Reporte final de consumo.....	100

INDICE DE TABLAS

Tabla N°1. Versiones del NodeMCU.....	18
Tabla N°2. Factor de conversión K para el modelo YF-S201.....	54
Tabla N°3. Factor de conversión K para el modelo FS300A.....	54
Tabla N°4. Escala de precios según consumo.....	74
Tabla N°5. Detalle costo de componentes electrónicos.....	100
Tabla N°6. Servicios de VPN.....	101
Tabla N°7. Costo total de la implementación del Prototipo.....	101

RESUMEN

El proyecto aporta el diseño de un prototipo electrónico que permite la lectura del consumo del servicio de agua potable.

El sistema consta en primera instancia de una parte que se encarga de acondicionar la señal proveniente del caudal del agua, para luego pasar a la siguiente etapa que está compuesta por un micro controlador, el mismo que recibe la señal acondicionada del caudal tratada previamente y la convierte en medidas numéricas para luego a través de un módulo wi-fi se envíen todas estas medidas a una base de datos para su posterior requerimiento.

En cada instante que el prototipo a través de todas las etapas detecte un caudal de agua obtendrá medidas que serán almacenadas automáticamente en una base de datos. El usuario podrá visualizar todos los datos mediante en una interfaz web a la cual tendrá acceso, la cual contendrá toda la información necesaria referente al consumo de agua potable como ser: cantidad, fecha, hora y el monto a pagar por el consumo realizado.



CAPITULO I

ANTECEDENTES DEL PROYECTO

1.1 ANTECEDENTES

Las nuevas tecnologías ayudan a los ciudadanos a desarrollar las actividades de su día a día, actualmente pocos ámbitos de la vida cotidiana pueden prescindir de ellas, uno de esos ámbitos son las viviendas. Hoy en día no se concibe un hogar que no cuente con un mínimo aporte tecnológico.

En la actualidad la tecnología dentro de las viviendas puede integrarse en otros ámbitos más allá de los electrodomésticos, en los últimos años se han desarrollado un sinnúmero de sistemas tecnológicos automatizados que han sido aplicados en la vida cotidiana del hogar, pero hay pocos sistemas enfocados al monitoreo de los servicios básicos en especial al servicio de agua potable, ya que en la mayoría de los hogares de la ciudad solo se cuenta con los medidores estándar instalados, de esta manera no existe un sistema que pueda ser utilizado para controlar de una manera más eficaz el consumo correspondiente que realiza cada familia en una vivienda multifamiliar.

Debido a estas situaciones es necesario tener un sistema que sea capaz de controlar el consumo de agua potable en todo momento.

1.2 PLANTEAMIENTO DEL PROBLEMA

Hoy en día, las familias que habitan en viviendas multifamiliares no tienen una idea exacta de cuanto es que tienen que pagar por el consumo del servicio de agua potable, limitándose a pagar por el servicio realizando un promedio del total consumido. Esto por falta de herramientas que permita medir en tiempo real el consumo del servicio enfocado de manera particular a cada una de las familias.

Debido a estas limitaciones es necesario contar con un prototipo electrónico que permita al usuario tener el control, el registro y la medición del consumo de agua potable que realizan cada uno de ellos.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Diseñar un prototipo electrónico de medición, control y registro de consumo de agua potable para aplicación doméstica.

1.3.2 OBJETIVOS ESPECÍFICOS

- Implementar el caudalímetro al microcontrolador que permita medir el caudal del agua.
- Acondicionar las variables eléctricas a medir.
- Implementar el prototipo de medición, monitoreo y registro.
- Realizar pruebas de control con el fin de verificar el correcto funcionamiento del registro de las variables medidas utilizando el sistema de gestión de datos.
- Diseñar una interfaz WEB que permita interactuar con la información reportada por un prototipo electrónico que hace medición, control y registro del consumo de agua.

1.4 JUSTIFICACIÓN

1.4.1 JUSTIFICACIÓN ACADEMICA

Empleando en la práctica importantes conocimientos adquiridos en nuestra carrera con referente a programación, manejo de protocolos de red, el manejo de microcontroladores

y variedad de sensores, se tiene fundamentos sólidos para diseñar un prototipo con funcionalidad de registro, control y medición, que trabaje de forma eficiente y confiable.

1.4.2 JUSTIFICACIÓN TÉCNICA

La disponibilidad de materiales y componentes electrónicos comerciales, así como los sistemas embebidos y las plataformas de software libre permiten realizar proyectos electrónicos cada vez más complejos y sofisticados a un costo accesible a nivel de hardware y software para su desarrollo e implementación.

1.4.3 JUSTIFICACIÓN SOCIAL

Dentro de los argumentos que justifican este tema, principalmente haremos mención a las ventajas de un dispositivo que permita medir y monitorear casi de manera instantánea la lectura de consumo de agua potable que será de gran beneficio para el usuario.

1.5 LIMITES Y ALCANCES

El proyecto de grado tiene los siguientes límites y alcances:

1.5.1 LIMITES

El proyecto usará sistemas embebidos disponibles en el país.

Se trabajará con información y datos técnicos que nos brinden cada dispositivo en sus páginas oficiales.

El presente proyecto está limitado a domicilios los cuales no cuenten con servicio de internet.

1.5.2 ALCANCES

El firmware del sistema, será capaz de actuar de forma independiente basándose en las consideraciones del usuario.

El proyecto podrá extender su aplicación a la domótica y telemetría.

1.6 METODOLOGÍA

Para la elaboración del presente proyecto de grado se analizarán circuitos digitales, protocolos de comunicaciones, lenguajes de programación. Lo que nos induce a realizar diseños analíticos así como pruebas experimentales, buscando que la parte teórica se relacione con la práctica.

El proyecto presenta diferentes etapas, por lo cual se utilizaran diferentes tipos de metodologías las cuales se detallan a continuación.

- El método analítico será utilizado en el análisis teórico de los protocolos de comunicación que serán utilizados como interfaz para la etapa de comunicación.
- El método lógico deductivo se lo empleara en los circuitos de toma de datos y el software de visualización para la interfaz, puesto que los procesos se llevaran dentro de los mismos.
- El método experimental es primordial en el presente proyecto porque el sistema propuesto deberá ser compatible con tecnologías existentes, cuya eficiencia y estabilidad será puesta a prueba de forma exhaustiva experimentando el correcto funcionamiento de los diversos subsistemas que la conforman.



CAPITULO II

MARCO TEÓRICO CONCEPTUAL

2.1 SISTEMAS EMBEBIDOS

Un sistema embebido consiste en una electrónica programable especialmente diseñada para aplicaciones específicas, frecuentemente en tiempo real y con requerimientos de alta confiabilidad. El nombre de embebido o empotrado deriva del hecho que forma parte de un sistema más amplio como puede ser una máquina o proceso.

Encontramos sistemas embebidos en productos tan diversos como electrodomésticos, todo tipo de dispositivos para comunicaciones y cómputo, automotores, etc.

Los sistemas embebidos integran recursos de hardware y software.

El hardware está constituido por :

- Un núcleo que consiste en una o más CPUs (Central Processing Unit) en alguno de los siguientes formatos: Microprocesador, Microcontrolador.
- La electrónica de acondicionamiento de señal que permite la interacción entre la CPU y el entorno.

El software de programación puede ser el propio lenguaje ensamblador de la CPU o un lenguaje de alto nivel con un compilador apropiado. Hay disponibles diversos entornos de desarrollo para la edición, compilación, carga y depuración del software, también utilizando los compiladores específicos, pueden utilizarse lenguajes como C o C++; en algunos casos, cuando el tiempo de respuesta de la aplicación no es un factor crítico, también pueden usarse lenguajes interpretados como JAVA.

La estrecha relación entre hardware y software ha posibilitado el diseño de dispositivos muy complejos atendiendo más al proceso de las señales sin tener que involucrarse en los circuitos electrónicos. De todas maneras un conocimiento más profundo en estos últimos conduce a obtener una mayor eficiencia.

En la figura se presenta la arquitectura típica de un sistema embebido

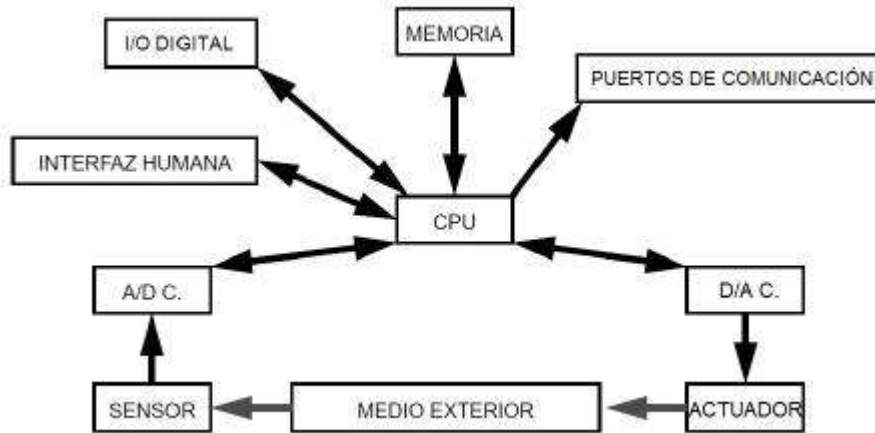


Figura N°1. Arquitectura típica de un sistema embebido
Fuente: www.panamahitek.com/Sistemas-Embebidos

2.2 PLACA DE DESARROLLO NODEMCU BASADO EN ESP8266

2.2.1 CONOCIENDO AL ESP8266

ESP8266 es el nombre de un micro controlador diseñado por Espressif Systems, una compañía china con sede en Shanghai. El volumen de producción de estos microcontroladores no empezó hasta principios de 2014.

El ESP8266 se anuncia a sí mismo como una solución autónoma de redes Wi-Fi que se ofrece como un puente entre los microcontroladores que hasta ahora existían hasta los MCU^[G] con Wi-Fi, siendo además capaz de ejecutar aplicaciones independientes.

Si se fuera a usar un ESP8266 salido de fábrica probablemente no se sabría qué hacer. Y es gracias a que los fabricantes los construyen encima de circuitos impresos y placas pre-fabricadas, que estos quedan listos para nuestro uso. Por eso se da lugar a varias versiones de ESP8266 (Ej.: ESP-01, ESP-02...) pero todas con el mismo procesador, lo que las diferencian son el número de pines GPIO^[G] expuestos, la cantidad de memoria flash, las dimensiones, la forma de exponer los pines, y otras consideraciones varias relativas a su construcción. Pero desde una perspectiva de programación todas son iguales. [1]

El circuito integrado ESP8266 viene en un pequeño paquete, tal vez cinco milímetros cuadrado. Obviamente, a menos que se sea un maestro soldador no va a hacer mucho con eso. Las buenas noticias son que una serie de proveedores han creado placas breakout que nos hacen el trabajo mucho más simple, ya que estas tarjetas contienen un único componente que, debido a sus reducidas dimensiones no pueden ser conectados directamente a una placa de desarrollo para sus salidas y por lo que son "rotos" o ampliados con el fin de tener acceso a sus terminales. Esto nos permite trabajar con este componente único acoplado a un microcontrolador, para desarrollar un proyecto para evaluar su funcionamiento, o para complementar un proyecto que requiere su presencia.

Poco a poco, la gran comunidad que hay detrás, comenzó a traducir la documentación y crear firmwares para el ESP8266. Un firmware no es más que el software de bajo nivel, la lógica que controla los circuitos electrónicos. Todo empezó a crecer y las aplicaciones se multiplicaron. En la actualidad podemos encontrar multitud de módulos que incorporan este microcontrolador.

A mediados de este año (2016), la empresa anunció una nueva versión, el ESP8285. Parecía un clon de su predecesor. La gran diferencia es que incorpora una memoria Flash de 1MB integrada dentro del propio chip. El ESP8266 no dispone de esta memoria y necesita de un módulo externo para almacenar los programas. Esto va a permitir crear dispositivos del IoT, o de otro tipo, del tamaño de una tecla de un teclado, todo un reto. Todas gracias a la conectividad y a la memoria Flash que incorpora. [1]

2.2.1.1 POSIBLES USOS DEL ESP8266

En este punto debemos de distinguir entre los módulos y los microcontroladores. El ESP8266 es un microcontroladores. Podemos trabajar con ellos sueltos o podemos comprarlo integrado dentro de un PCB. ^[G]

Ocurre lo mismo que con Arduino. Tenemos la posibilidad de comprar un microcontrolador Atmel o comprar la placa de Arduino donde ya viene integrado.

Dicho todo esto, el uso que le demos dependerá de si lo tenemos como chip o como módulo. Dentro de la gran cantidad de usos cabe destacar los siguientes: [5]

- Electrodomésticos conectados.
- Automatización del hogar.
- Casas inteligente.
- Automatización de la industria.
- Monitor de bebés.
- Cámaras IP.
- Redes de sensores.
- Woreables.
- IoT (Internet of Things o Internet de las Cosas)
- IIoT (Industrial Internet of Things o Internet de las Cosas para el sector Industrial)

2.2.1.2 ESPECIFICACIONES DEL MICROCONTROLADOR ESP8266

Incluye todo lo necesario para conectarse a la Wi-Fi ^[G] y un procesador interno que hace ruborizarse a nuestros queridos Arduinos.

Las principales características del ESP8266 son las siguientes:

Hardware

- Utiliza una CPU Tensilica L106 32 bits corriendo 80 MHz (que puede ser overclokeado a 160MHz)
- Voltaje de operación entre 3V y 3,6V
- Corriente de operación 80 mA
- Temperatura de operación -40°C y 125°C
- 16 pines GPIO programables disponibles.
- Un convertidor Analógico-Digital (ADC) de 10 bit
- UART en los pines dedicados (usada para la programación del chip)
- SPI e I2C

- Consumo en reposo <10 mW.

Conectividad

- Soporta los protocolos TCP/IP/UDP/HTTP/FTP ^[G]
- Gestión completa del WIFI con amplificador incluido.
- Protocolo IEEE 802.11 b/g/n y Wifi Direct (P2P) ^[G] .
- 30 Metros de alcance teórico.

2.2.1.3 PROCESADOR

El system on a chip (SoC) ESP9266EX usa un microcontrolador Tensilica Xtensa L106, que es un procesador de 32 bit.



Figura N°2. Procesador ESP8266EX
Fuente: idoneos.com/esp8266

2.2.1.4 ARQUITECTURA

Tiene una arquitectura de Harvard, con lo cual la CPU puede tanto leer una instrucción como realizar un acceso a la memoria de datos al mismo tiempo, incluso sin una memoria caché. En consecuencia, una arquitectura de computadores Harvard puede ser más rápida para un circuito complejo, debido a que la instrucción obtiene acceso a datos y no compite por una única vía de memoria.

2.2.1.5 VARIANTES Y MODELOS DEL ESP8266

Como bien se ha comentado con anterioridad el ESP8266 tiene siempre el mismo procesador, pero su versión varia a la hora de construirlo sobre una placa impresa ya que sus características de construcción difieren en diferentes aspectos.

Es decir el chip siempre es el mismo, pero según el soporte en que lo montan pueden sacar más o menos pines al exterior, lo que no deja de ser una curiosidad teniendo en cuenta la cantidad de modelos que se ofertan

Modelo ESP-01

Este modelo es el más frecuente de encontrar. Es el que hemos usado en nuestros tutoriales hasta la fecha y por defecto trae el firmware que permite conectarlo al puerto serie de nuestros Arduinos y aceptar comandos AT.

La virtud de este modelo es que incluye un par de pines digitales disponibles GPIO0 y GPIO2, pero si al arrancar el modulo el GPIO0 está a GND, entra en modo de programación por lo que no conviene usarlo como entrada sino solo como salida.

Además las patillas no sirven para protoboard, lo que es un auténtico asco y el mayor defecto de este módulo. Venden adaptadores por ahí, para protoboard pero suelen costar más que el propio modulo.

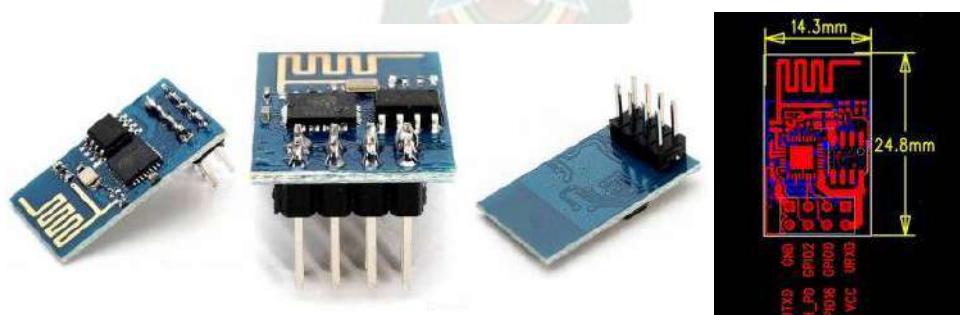


Figura N°3. ESP8266 modelo ESP-01

Fuente: prometec.net/esp8266/modelos

Modelo ESP-02

Tiene 3 pines digitales al exterior GPIO0, GPIO2, y GPIO15 y además acepta una antena Wi-Fi externa lo que le hace muy interesante, para montajes que requieran alcance de la señal Wi-Fi^[G].

No está muy claro si se puede montar en protoboard, porque no he encontrado confirmación, pero tiene dimensiones reducidas y eso siempre viene bien para cualquier invento.

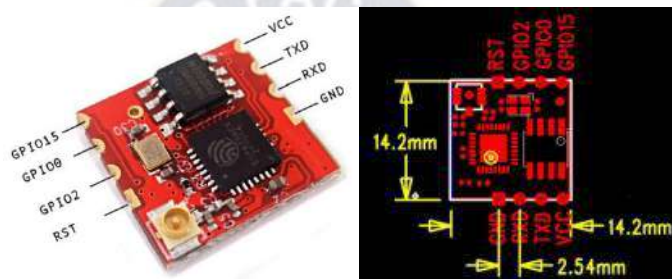


Figura N°4. ESP8266 modelo ESP-02
Fuente: prometec.net/esp8266/modelos

Modelo ESP-03

Es uno de los módulos más interesantes porque dispone de más pines que el ESP-01, Saca los pines GPIO 2, 12, 13, 14, 15,16 e incluye una antena cerámica.

Siguiendo la costumbre de marearnos, este módulo tampoco puede conectarse a la protoboard directamente y necesita un adaptador



Figura N°4. ESP8266 modelo ESP-03
Fuente: prometec.net/esp8266/modelos

Modelo ESP-05

Su mayor interés es para usarlo con Arduino y otros micros como un simple módulo Wi-Fi con comandos AT. No extra ningún pin GPIO ^[G] y viene con conectores supuestamente capaces de encajar en la protoboard (Sin confirmar)

También se puede usar como un mini procesador que saca los pines de comunicación al exterior.

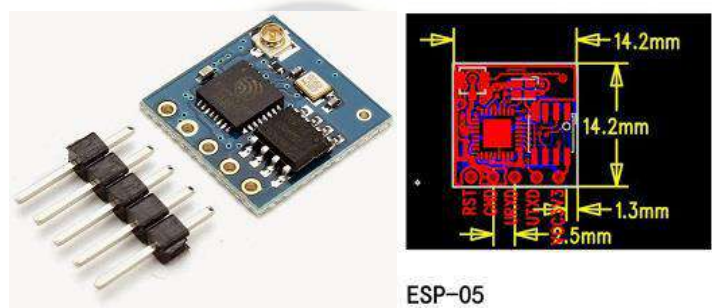


Figura N°5. ESP8266 modelo ESP-05
Fuente: prometec.net/esp8266/modelos

Modelo ESP-12

Este es probablemente uno de los modulas más interesantes, porque te da acceso a todos los pines del ESP8266, 11 GPIOs más 1 convertidor analógico a digital con antena integrada y buen alcance.

Como siempre su mayor defecto es que no puede insertarse en protoboard para pruebas, pero existen, naturalmente, adaptadores para ello.

Ha sido el modelo estrella hasta hace unos pocos meses que aparecieron los modelos de Olimex y NodeMCU, y es un modelo de lo mas interesanta para incluir en producto final si necesitas mas pines de las que dispone el ESP-01, pero para hacer prototipos sigues necesitando un adaptador FTDI a USB. [2]

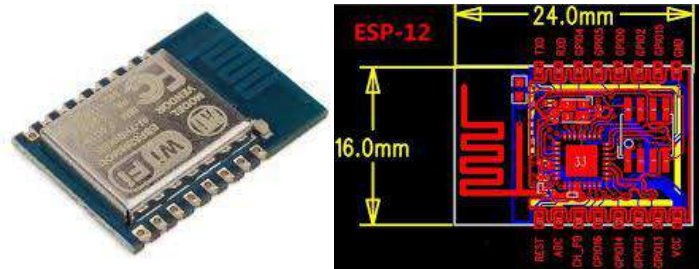


Figura N°6. ESP8266 modelo ESP-12
 Fuente: prometec.net/esp8266/modelos

2.2.2 PLACA DE DESARROLLO NODEMCU

NodeMCU es una placa de desarrollo totalmente abierta, a nivel de software y de hardware. Al igual que ocurre con Arduino, en NodeMCU todo está dispuesto para facilitar la programación de un microcontrolador o MCU (del inglés Microcontroller Unit).

No hay que confundir microcontrolador con placa de desarrollo. NodeMCU no es un microcontrolador al igual que Arduino_MKR1000 tampoco lo es. Son placas o kits de desarrollo que llevan incorporados un chip que se suele llamar SoC (System on a Chip) que dentro tiene un microcontrolador o MCU

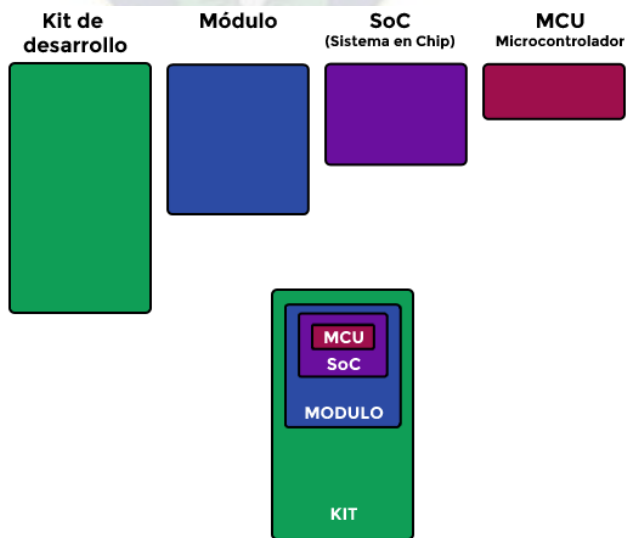


Figura N°7. Esquema general de placas de desarrollo
 Fuente: programafacil.com/nodemcu

El objetivo es programar la MCU o microcontrolador a través del kit o placa de desarrollo. Todo lo demás nos sirve de apoyo para que crear nuestros propios proyectos sea lo más sencillo posible.

Esto no solo sucede con NodeMCU, las placas como Arduino UNO o Arduino MKR1000 utilizan la misma arquitectura eso si, cada una con sus características particulares.

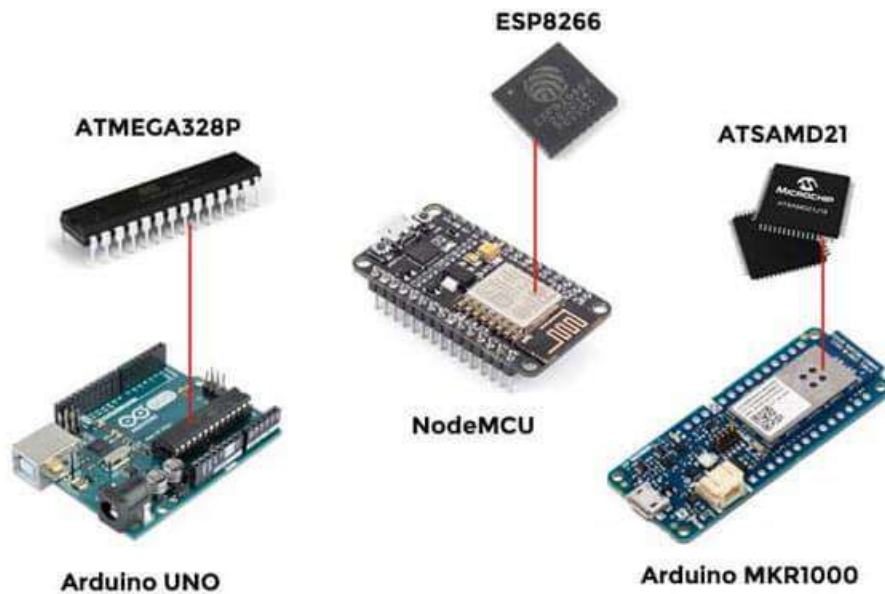


Figura N°8. Kit o placas de desarrollo
Fuente: programafacil.com/nodemcu

En principio el nombre NodeMCU se refería principalmente al firmware. Actualmente esto se ha invertido y cuando hablamos de NodeMCU normalmente nos referimos a la placa de desarrollo.

El firmware NodeMCU fue creado poco después de aparecer el ESP8266, el 30 de diciembre de 2013. Unos meses después, en octubre de 2014 se publicó la primera versión del firmware NodeMCU en Github. Dos meses más tarde se publicaba la

primera placa de desarrollo NodeMCU, denominada devkit v0.9, siendo también Open Hardware.

La mayoría del interés de la comunidad se limitaba al ESP01, que se consideraba más un módulo Wifi barato para procesadores como Arduino que una placa de desarrollo independiente.

El firmware NodeMCU podía grabarse en un ESP8266, tras lo cual podíamos programarlo con el lenguaje script Lua. Con el paso del tiempo y la aparición de otras alternativas para programar ESP8266, como (especialmente) con C++ con el entorno del Arduino y otras como MicroPython, el interés en Lua ha disminuido considerablemente.

A pesar de que la programación en Lua tenía aspectos interesantes, no es un lenguaje tan extendido como C++ y Python. Además, nunca consiguieron hacerlo totalmente estable en el ESP8266. Por otro lado, al ser un lenguaje interpretado (en lugar de compilado) el rendimiento y aprovechamiento de los recursos es inferior.

Básicamente, la placa de desarrollo NodeMCU está basada en el ESP12E y expone las funcionalidades y capacidad del mismo.

La tarjeta NodeMCU está diseñada especialmente para trabajar en protoboard. Posee un regulador de voltaje en placa que le permite alimentarse directamente del puerto USB. Los pines de entradas/salidas trabajan a 3.3V. El chip CP2102 se encarga de la comunicación USB-Serial. Pero además añade las siguientes ventajas propias de placas de desarrollo:[3]

- Puerto micro USB y conversor Serie-USB
- Programación sencilla a través del Micro-USB
- Alimentación a través del USB
- Terminales (pines) para facilitar la conexión
- LED y botón de reset integrados

2.2.2.1 VERSIONES DE NODEMCU

Existe una gran confusión con respecto a las diferentes versiones que hay de NodeMCU. Todo esto es debido a que se trata de una placa de hardware abierto y cualquier fabricante puede crear su propia distribución.

Lo que te tiene que quedar claro es que todos los NodeMCU se basan en los mismos módulos el ESP-12 y ESP-12E que se a su vez se basan en el SoC ESP8266.

Partiendo de esta premisa, las diferencias que vamos a encontrar son básicamente el número de pines a los que tenemos acceso y el tamaño de cada placa. Actualmente se cuentan con tres versiones.

Existe cierta confusión al respecto y es debido a que se han nombrado de diferentes maneras. Podemos encontrar el nombre dependiendo de la generación a la que pertenecen, la versión o el nombre común que se les ha dado.

Generación	Versión	Módulo	Anchura
Primera	0.9 V1	ESP12	10 pines
Segunda	1.0 V2	ESP12E	8 pines
Tercera	1.0 V3	ESP12E	10 pines

Tabla N°1. Versiones del NodeMCU
Fuente: programafacil.com/nodemcu

La tercera columna es el nombre que encontrarás en diferentes tiendas online como Amazon, Baggood o AliExpress. Lo lógico es que te guíes por este nombre a la hora de comprar.

La placa V3 es básicamente la V2 con algunas mejoras y pertenece a la 2ª generación de NodeMCU aunque no es una nueva especificación oficial.

El gran problema que existe con respecto a los nombres es debido a que se trata de una placa de hardware abierto y cualquiera puede fabricar un NodeMCU y comercializarlo.

Primera generación v0.9

La versión 0.9 amarilla-anaranjada y muy ancha. Con unas dimensiones de 47x21mm ocupaba 10 hileras de pins de una placa breadboard, por lo que la tapa por completo. Esto la hacía muy poco práctica de usar porque no deja pines libres en la breadboard para realizar conexiones.

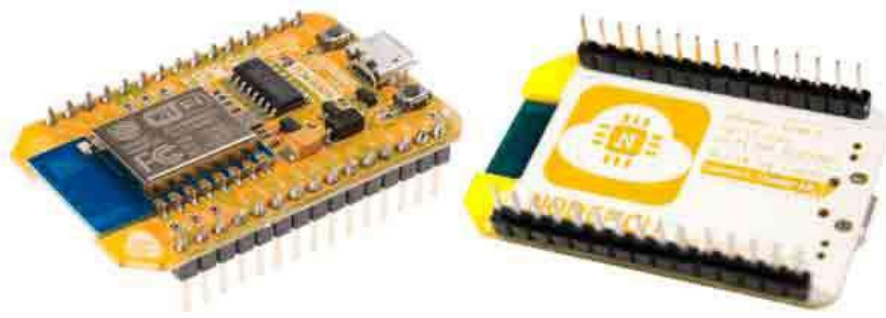


Figura N°9. NodeMCU V0.9 / V1
Fuente: programafacil.com/nodemcu

Segunda generación v1.0 / V2

La siguiente versión del NodeMCU es la v1.0 V2. De forma resumida Amica, una compañía creada por el alemán Gerwin Janssen, fabricó su propia versión mejorada de la v0.9. Al equipo original de NodeMCU le gustó y **la declararon versión “oficial” de NodeMCU.**

La principal diferencia de la versión v1.0 v2 es que **monta un ESP12E en lugar de un ESP12**, por lo que tiene más pines disponibles que el modelo original. Además, es más estrecha que la versión 0.9, tapando sólo 8 hileras de una breadboard. Esto deja una hilera adicional a cada lado para realizar conexiones.

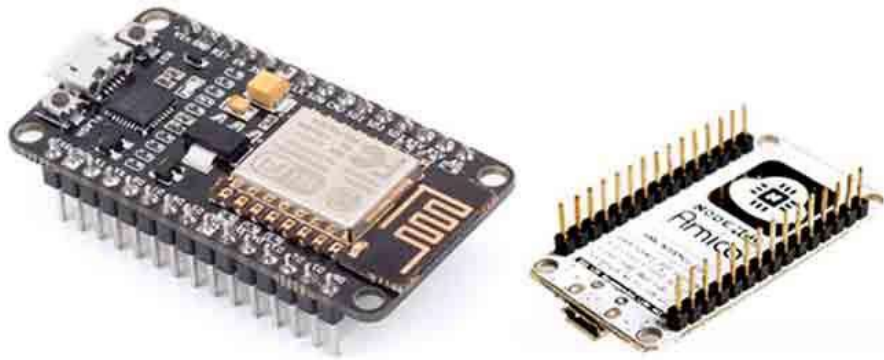


Figura N°10. NodeMCU V1.0 / V2
Fuente: programafacil.com/nodemcu

Tercera generación v1.0 / V3

Llegamos a lo que a veces se denomina “tercera generación”, la versión 1.0 V3. Básicamente el fabricante Lolin/Wemos decidió crear su propio diseño “mejorado” con unos cuantos cambios menores.

El cambio principal es que la V3 monta un conversor serial CH340G en lugar del CP2102. El fabricante asegura que hace que el puerto USB sea más robusto. Por otro lado, reusaron los dos pines reservados de la V2 para sacar un GND y un VUSB.

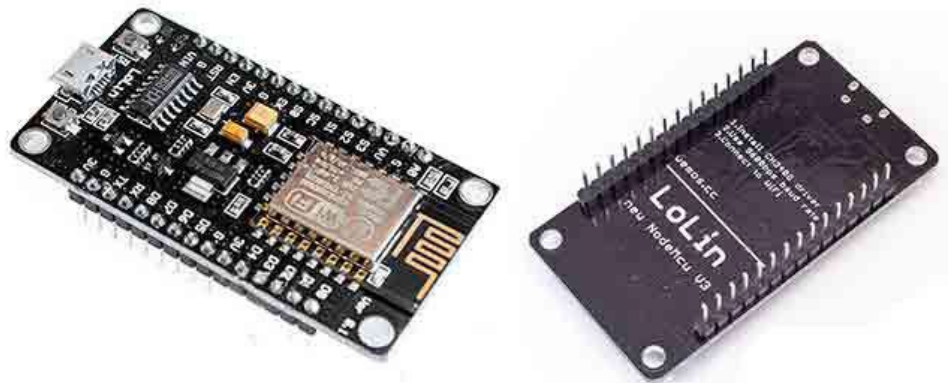


Figura N°11. NodeMCU V1.0 / V3
Fuente: programafacil.com/nodemcu

2.2.2.2 DESCRIPCIÓN DE CADA UNO DE LOS PINES NODEMCU ESP12.

Importante conocer a que pin de la NodeMCU corresponde el numero asignado en el programa, aquí la relación que existe:

- El pin D0 corresponde con el número 16 .
- El pin D1 corresponde con el número 5 .
- El pin D2 corresponde con el número 4 .
- El pin D3 corresponde con el número 0 .
- El pin D4 corresponde con el número 2 .
- El pin D5 corresponde con el número 14 .
- El pin D6 corresponde con el número 12 .
- El pin D7 corresponde con el número 13 .
- El pin D8 corresponde con el número 15 .
- El pin D9 corresponde con el número 3 .
- El pin D10 corresponde con el número 1 .

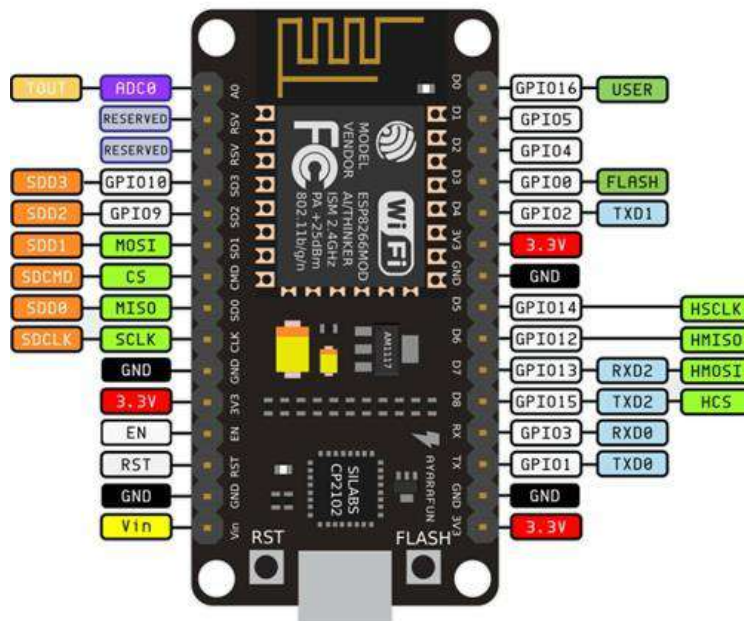


Figura N°12. Pinaje de NodeMCU
Fuente: www.luisllamas.es/esp8266-nodemcu

- **Pins GPIO.** Los pins gpio (General Purpose Input/Output) son los destinados a entradas y salidas digitales.
- **El pin A0 (ADC0) o TOUT,** es el destinado a entradas analógicas. Este pin recibe una señal que dependiendo del receptor modula la tensión de entrada al pin A0, el rango de la tensión puede estar entre 0 y 5 Voltio (dependiendo del emisor). La tensión es recibida por la entrada y convertida en valores digitales que van del 0 al 1024.
- **Los pines PWM** se pueden utilizar para las salidas analógicas.
- **Los pines SPI** (Serial Peripheral Interface) se utilizan para las comunicaciones entre circuitos integrados siguiendo un protocolo de comunicación síncrono, como por ejemplo con pantallas OLED. Entre los pines SPI tenemos SPICLK, SPIQ (MISO), SPID (MOSI) y SPICS.
- **Pines UART** (Universal asynchronous Receiver / Transmitter) Los pines UART los utilizamos para la recepción (RX) y transmisión (TX) de datos de la nodemcu. Los datos enviados a través de los pines UART nos permiten configurar tanto la velocidad de transmisión de los datos como su formato.

Los pines UART en la nodemcu son TXD0 (GPIO1), RXD0 (GPIO3), TXD1 (GPIO2), TXD2 (GPIO15), RXD2 (GPIO13).

- **Pin Flash.** El pin flash (GPIO0) es recomendable tenerlo en un estado "LOW", es decir puesto a tierra cada vez que quieras flashear tu nodemcu, al igual que el pin GPIO15. Por el contrario se deberá dar tensión de 3.3V (estado HIGH) al pin GPIO2, a través de una resistencia de 3.3k.[3]

2.2.3 ENTORNO DE PROGRAMACIÓN PARA NODEMCU CON "ARDUINO IDE".

Al hablar de estas placas debemos hacer mención de Arduino IDE, que es un entorno de desarrollo conocido como IDE y es en donde se realiza la programación de cada una de las placas.

Con esta labor la compañía ha buscado facilitar el acceso y uso no solo de la electrónica sino también de la programación de sistemas embebidos. Además, es necesario considerar que toda la plataforma presentada por la compañía se libera bajo la licencia de código abierto y es lo que da total libertad de acceso.

Arduino IDE es un editor de texto y compilador para programar y transferir el contenido de las instrucciones a la placa de Arduino en su lenguaje máquina. El lenguaje de programación utilizado es Processing.

El Software Arduino IDE se compone de 3 partes principalmente:

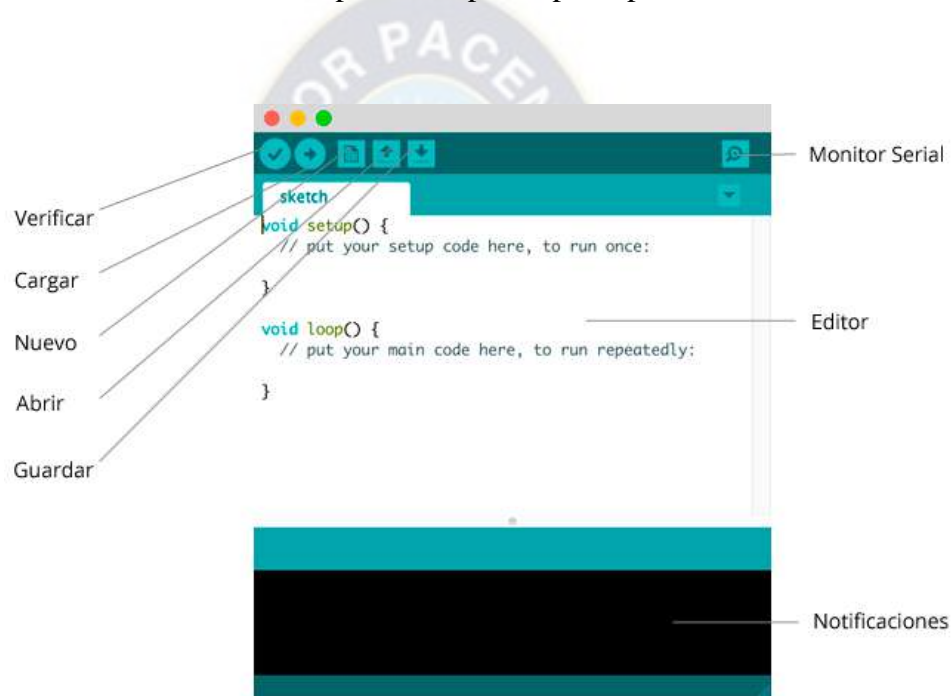


Figura N°13. Software Arduino IDE

Fuente: www.arduino.cc

- **Botonera** o barra de navegación:
 - Verificar: Se encarga de verificar la sintaxis de nuestro programa.
 - Cargar: Si la verificación ha sido correcta, podemos cargar el código en nuestra placa de Arduino.
 - Nuevo: Simplemente abrimos un documento vacío (salvo funciones principales) para comenzar un nuevo programa.
 - Abrir: Para abrir proyectos en otros directorios o rutas.

- Guardar: Simplemente guarda el programa en el directorio que especifiquemos (si es la primera vez que lo guardamos).
- Monitor serial: Supongamos que necesitamos saber en algún momento qué ocurre dentro de nuestra placa de Arduino, pues bien, mediante el monitor serial podemos enviar datos que se mostrarán en nuestro monitor.
- **Editor** de programación: Es la parte principal de Arduino IDE, básicamente donde se programan las líneas y líneas de código en lenguaje processing.
- **Notificaciones**: Conocido normalmente por consola, es la parte de depuración donde notifica al programador sobre errores de sintaxis, comunicación, etc.

2.2.3.1 ESTRUCTURA DE UN PROGRAMA EN LA IDE ARDUINO

La estructura básica del lenguaje de programación de Arduino es bastante simple y se compone de al menos dos partes. Estas dos partes necesarias, o funciones, encierran bloques que contienen declaraciones, estamentos o instrucciones, como se indica en la Figura N°1

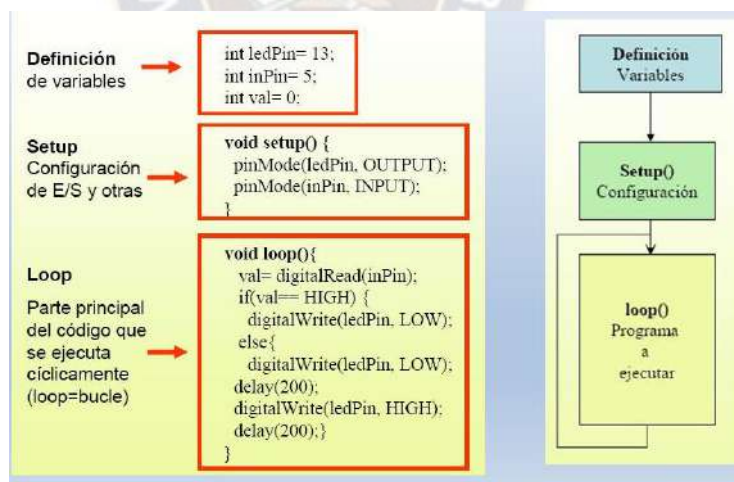


Figura N°14. Estructura de un programa
Fuente: aprendiendoarduino.wordpress.com

En donde **setup()** es la parte encargada de recoger la configuración y **loop()** es la que contiene el programa que se ejecutará cíclicamente (de ahí el término loop –bucle-). Ambas funciones son necesarias para que el programa trabaje.

La función de configuración (**setup**) debe contener la declaración de las variables. Es la primera función a ejecutar en el programa, se ejecuta sólo una vez, y se utiliza para configurar o inicializar **pinMode** (modo de trabajo de las E/S), configuración de la comunicación en serie y otras.

La función bucle (**loop**) siguiente contiene el código que se ejecutara continuamente (lectura de entradas, activación de salidas, etc) Esta función es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo.

setup()

La función setup() se invoca una sola vez cuando el programa empieza. Se utiliza para inicializar los modos de trabajo de los pins, o el puerto serie. Debe ser incluido en un programa aunque no haya declaración que ejecutar. Así mismo se puede utilizar para establecer el estado inicial de las salidas de la placa.

```
void setup()
{
  pinMode(pin, OUTPUT); // configura el 'pin' como salida
  digitalWrite(pin, HIGH); // pone el 'pin' en estado HIGH
}
```

loop()

Después de llamar a setup(), la función loop() hace precisamente lo que sugiere su nombre, se ejecuta de forma cíclica, lo que posibilita que el programa esté respondiendo continuamente ante los eventos que se produzcan en la placa.

```
void loop()
{
  digitalWrite(pin, HIGH); // pone en uno (on, 5v) el 'pin'
  delay(1000);           // espera un segundo (1000 ms)
  digitalWrite(pin, LOW); // pone en cero (off, 0v.) el 'pin'
  delay(1000);
}
```

funciones

Una función es un bloque de código que tiene un nombre y un conjunto de instrucciones que son ejecutadas cuando se llama a la función. Son funciones setup() y loop() de las

que ya se ha hablado. Las funciones de usuario pueden ser escritas para realizar tareas repetitivas y para reducir el tamaño de un programa. Las funciones se declaran asociadas a un tipo de valor “type”. Este valor será el que devolverá la función, por ejemplo 'int' se utilizará cuando la función devuelve un dato numérico de tipo entero. Si la función no devuelve ningún valor entonces se colocará delante la palabra “void”, que significa “función vacía”. Después de declarar el tipo de dato que devuelve la función se debe escribir el nombre de la función y entre paréntesis se escribirán, si es necesario, los parámetros que se deben pasar a la función para que se ejecute.

```
type nombreFunción(parámetros)
{
instrucción;
}
```

delayVal()

Se utiliza para poner un valor de retraso en un programa que lee una variable analógica de un potenciómetro conectado a una entrada de Arduino. Al principio se declara como una variable local, 'v' recoge el valor leído del potenciómetro que estará comprendido entre 0 y 1023, luego se divide el valor por 4 para ajustarlo a un margen comprendido entre 0 y 255, finalmente se devuelve el valor 'v' y se retornaría al programa principal. Esta función cuando se ejecuta devuelve el valor de tipo entero 'v'.

```
int delayVal()
{
int v;           // crea una variable temporal 'v'
v= analogRead(pot); // lee el valor del potenciómetro
v /= 4;         // convierte 0-1023 a 0-255
return v;       // devuelve el valor final
}
```

{ } entre llaves

Las llaves sirven para definir el principio y el final de un bloque de instrucciones. Se utilizan para los bloques de programación setup(), loop(), if..., etc.

```
type funcion()
{
instrucciones;
}
```

Una llave de apertura “{“ siempre debe ir seguida de una llave de cierre “}”, si no es así el programa dará errores.

El entorno de programación de Arduino incluye una herramienta de gran utilidad para comprobar el total de llaves. Sólo tienes que hacer click en el punto de inserción de una llave abierta e inmediatamente se marca el correspondiente cierre de ese bloque (llave cerrada).

; punto y coma

El punto y coma “;” se utiliza para separar instrucciones en el lenguaje de programación de Arduino. También se utiliza para separar elementos en una instrucción de tipo “bucle for”.

```
int x = 13; /* declara la variable 'x' como tipo entero de valor 13 */
```

Nota: Olvidaos de poner fin a una línea con un punto y coma o se producirá en un error de compilación. El texto de error puede ser obvio, y se referirá a la falta de una coma, o puede que no. Si se produce un error raro y de difícil detección lo primero que debemos hacer es comprobar que los puntos y comas están colocados al final de las instrucciones.[4]

2.3 MEDIDORES DE FLUJO

También llamados caudalímetros, son dispositivos utilizados para la medición de los caudales o gastos volumétricos de un fluido.

Un caudalímetro es un sensor que permite medir la cantidad de agua que atraviesa una tubería. Podemos conectar un caudalímetro a un procesador como Arduino para obtener la medición del sensor.

El nombre del caudalímetro proviene del término caudal, que es la relación entre volumen y tiempo. Las unidades en el sistema internacional son m^3/s (metros cúbicos por segundo), siendo otras unidades habituales l/s (litros por segundo) y l/min (litros por minuto). El caudal depende de diversos factores, principalmente de la sección de tubería y la presión de suministro.

En instalaciones domésticas diámetros habituales de tuberías 1", 3/4" y 1/2", siendo esta última la normal en grifos. La presión debería estar en el rango de 100 kPA (1 Kg/cm²) a 500kPA (5 Kg/cm²)

Caudales habituales para instalaciones para tuberías de 1/2" (las normales en grifos) 0.1 l/s (6 l/min) y 0.2 l/s (12 l/min). Para tuberías de 3/4" podemos tener caudales en torno a 20 l/min, y para tuberías de 1" en torno a 35 l/min.

Los medidores de agua se pueden clasificar en función de la forma de contar el caudal.[7]

2.3.1 CLASIFICACIÓN DE CAUDALIMETROS

Básicamente se clasifican en tres tipos de acuerdo a su funcionamiento.

2.3.1.1 CAUDALIMETRO ELECTRÓNICO DE TURBINA

Los caudalímetros de Turbina van provistos de una hélice que gira cuando la corriente de líquido incide sobre ella. La velocidad de giro es proporcional al caudal, y para determinarla, se emplea un captador que genera pulsos cuando gira la hélice. Este captador genera:

1 pulso cada vez que un aspa de la hélice pasa frente a él, en caso de los caudalímetros de turbina de acero inoxidable y los de plástico de líquidos corrosivos, y 2 pulsos por cada vuelta de la hélice, en el caso de turbinas económicas.

De esta forma se obtiene un tren de pulsos cuya frecuencia permite determinar el caudal.

La precisión de los caudalímetros de Turbina depende del modelo y del rango de medición, pudiendo variar el error máximo de medida entre el 0,1%, para los modelos de Acero inoxidable y rango pequeño de medición, y el 1,5% para los modelos económicos. Los caudalímetros de Turbina están pensados para medir líquidos limpios con una viscosidad máxima de 30 centiPoises. Los caudalímetros de Turbina se fabrican en diferentes tipos de materiales dependiendo de la aplicación a que se vaya a dedicar, pudiendo ser de acero inoxidable y de diferentes plásticos.

Los caudalímetros de Turbina deben estar correctamente instalados si queremos medir con precisión. En el caso de los de acero inoxidable y los de plástico para líquidos corrosivos. [7]

2.3.1.2 CAUDALIMETRO MECÁNICO DE MOLINO

Consisten en un molino cuyas aspas están transversales a la circulación de fluido. El flujo hace girar el molino cuyo eje mueve un contador que acumula lecturas.

Un ejemplo de este uso son los contadores de agua de las viviendas. [8]

2.3.1.3 CAUDALIMETRO ELECTROMAGNÉTICO

Su principio de medida está basado en la Ley de Faraday, la cual expresa que al pasar un fluido conductivo a través de un campo magnético, se produce una fuerza electromagnética (F.E.M.), directamente proporcional a la velocidad del mismo, de donde se puede deducir también el caudal.

Está formado por un tubo, revestido interiormente con material aislante. Sobre dos puntos diametralmente opuestos de la superficie interna se colocan dos electrodos metálicos, entre los cuales se genera la señal eléctrica de medida. En la parte externa se colocan los dispositivos para generar el campo magnético, y todo se recubre de una protección externa, con diversos grados de seguridad. [8]

El flujo completamente sin obstrucciones es una de las ventajas de este medidor. El fluido debe ser ligeramente conductor debido a que el medidor opera bajo el principio de que cuando un conductor en movimiento corta un campo magnético, se induce un voltaje.

Dentro del campo de caudalímetros que podemos emplear en nuestros proyectos de electrónica y domótica tenemos diversos modelos como el:

- **YF-S201**
- **FS300A**
- **FS400A.**

Cada uno dispone de distintas características, aunque el criterio de selección entre estos tres será el diámetro de la tubería.

Los caudalímetros como el YF-S201, FS300A y el FS400A están constituidos por una carcasa plástica y un rotor con paletas en su interior. Al atravesar el fluido el interior el sensor el caudal hace girar el rotor.

La velocidad de giro se determina mediante un imán fijado al rotor, que es detectado mediante un sensor hall externo a la carcasa. Por tanto, ninguna parte eléctrica está en contacto con el fluido.

La salida del sensor es una onda cuadrada cuya frecuencia es proporcional al caudal atravesado.

$$f (Hz) = K \cdot Q (l/min) \Rightarrow Q (l/min) = \frac{f(Hz)}{K}$$

El factor K de conversión entre frecuencia (Hz) y caudal (L/min) depende de los parámetros constructivos del sensor. El fabricante proporciona un valor de referencia en sus Datasheet. No obstante, la constante K depende de cada caudalímetro. Con el valor

de referencia podemos tener una precisión de +-10%. Si queremos una precisión superior deberemos realizar un ensayo para calibrar el caudalímetro.

Sensor de flujo	Modelo	Conexión	Caudal
	YF-5201	½"	1-30 L/min
	F5300A	¾"	1-60 L/min
	F5400A	1"	1-60 L/min

Figura N°15. Modelos caudalímetros electromagnéticos
 Fuente: naylampmechatronics.com/blog/47_tutorial-sensor-de-flujo

Todos los modelos tienen tres cables para su conexión, rojo y negro para la alimentación y amarillo para la salida de los pulsos. [8]

2.4 SERVIDOR WEB LAMP

Las siglas LAMP se emplean para definir el trabajo conjunto con Linux, Apache, MySQL y uno de los siguientes lenguajes: Perl, Python o PHP.

El término hace referencia al sistema creado por la conjunción de esas aplicaciones libres (de código abierto). Este grupo de aplicaciones generalmente son usados para crear servidores web.

LAMP está considerada como una de las mejores herramientas disponibles para que cualquier organización o individuo pueda emplear un servidor web^[G] versátil y potente. Aunque creados por separado, cada una de las tecnologías que lo forman dispone de una serie de características comunes. Especialmente interesante es el hecho que estos cuatro productos pueden funcionar en una amplia gama de hardware, con requerimientos

relativamente pequeños sin perder estabilidad. Esto ha convertido a LAMP en la alternativa más adecuada para pequeñas y medianas empresas. [9]

2.4.1 APACHE

Para entender mejor lo que es Apache, primeramente definiremos lo que es un servidor web. La definición más sencilla de **servidor web**^[G], que es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc). Estos servidores web utilizan el protocolo http.

Los servidores web están alojados en un ordenador que cuenta con conexión a Internet. El web server, se encuentra a la espera de que algún navegador le haga alguna petición, como por ejemplo, acceder a una página web y responde a la petición, enviando código HTML mediante una transferencia de datos en red. [10]

Por defecto ya lo tendremos instalado, pero en el caso de que no sea así lo podemos instalar con los siguientes comandos.

```
sudo apt-get update  
sudo apt-get install apache2
```

Puedes hacer una comprobación instantánea de que todo ha ido según lo planeado visitando la dirección IP pública de tu servidor en un navegador.

```
http://your_server_ip
```

Y tendrá que aparecer la pantalla por defecto de apache.



Figura N°16. Página web determinada de apache para ubuntu
Fuente: www.digitalocean.com/community/apache

2.4.2 MYSQL

MySQL es un sistema de gestión de base de datos. Básicamente, se encarga de organizar y facilitar el acceso a las bases de datos donde nuestro sitio puede almacenar información.

Para la instalación debemos ejecutar:

```
sudo apt-get install mysql-server
```

Durante la instalación, el servidor te pedirá que selecciones y confirmes una contraseña para el usuario “root” de MySQL. Esta es una cuenta administrativa en MySQL que ha aumentado privilegios. Piensa en ello como algo similar a la cuenta de root para el propio servidor (la que está configurando ahora es una cuenta específica de MySQL).

El siguiente paso sera ejecutar:

```
sudo mysql_secure_installation
```

Te pedirá la contraseña del root del mysql así como si deseas cambiar la contraseña, se la colocas y le das a no en cambiar contraseña; el resto de campos aceptarlos por defecto.

Este comando lanza un script que eliminará algunos usuarios de ejemplo y bases de datos, desactivara las conexiones root remotas, y cargara estas nuevas reglas para que MySQL respete inmediatamente los cambios que hemos hecho. [11]

2.4.3 LENGUAJE DE PROGRAMACIÓN PHP

PHP (acrónimo recursivo de *PHP*: Hypertext Preprocessor) es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

¿Cómo trabaja PHP? El lenguaje PHP se procesa en servidores, que son potentes ordenadores con un software y hardware especial. Cuando se escribe una dirección tipo `http://www.*****/index.php` en un navegador web como Internet Explorer, Firefox o Chrome, ¿qué ocurre? Se envían los datos de la solicitud al servidor que los procesa, reúne los datos (por eso decimos que es un proceso dinámico) y el servidor lo que devuelve es una página HTML como si fuera estática.

En resumen:

Páginas estáticas: Petición --> Respuesta

Páginas dinámicas: Petición --> Procesado y preparación --> Respuesta

PHP es el componente de nuestra configuración que procesará código para mostrar contenido dinámico. Puede ejecutar secuencias de comandos, conectarse a nuestras bases de datos MySQL para obtener información, y entregar el contenido procesado a nuestro servidor web para mostrarlo.[12]

Para instalarlo ejecutar:

```
sudo apt-get install libapache2-mod-php5 php5 mcrypt php5-mcrypt
```

2.4.4 GESTOR DE BASE DE DATOS phpMyAdmin

PHPMYAdmin es un software de código abierto, diseñado para manejar la administración y gestión de bases de datos MySQL a través de una interfaz gráfica de usuario. Escrito en PHP, phpMyAdmin se ha convertido en una de las más populares herramientas basadas en web de gestión de MySQL. PhpMyAdmin viene con una documentación detallada y está siendo apoyado por un gran multi-idioma de la comunidad. PhpMyAdmin es cada vez mayor lista de características soporta todas las operaciones de uso común tales como la navegación, pasando, crear, modificar las bases de datos MySQL, las tablas, campos e índices. Además, phpMyAdmin le permite administrar usuarios MySQL y privilegios de usuario. [13]

La instalación de phpMyAdmin es simple en máquinas Ubuntu. Abre el terminal y ejecuta el siguiente comando:

```
$ sudo apt-get install phpmyadmin
```

Cuando comience la instalación, obtendremos el siguiente mensaje que te pedirá que selecciones el servidor web, selecciona 'apache' y presiona ok.

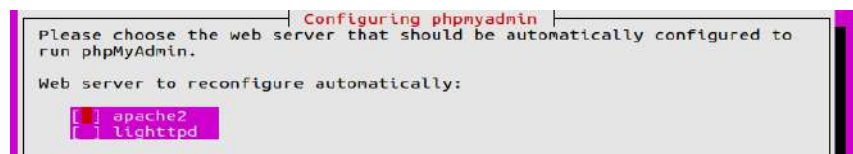


Figura N°17. Instalación MySQL

Fuente: www.digitalocean.com/community/mysql

La siguiente ventana desplegada será el paquete de phpMyAdmin que contendrá la base de datos (dbconfig-common). Aceptamos la opción que está por defecto. A **continuación** seleccionamos una contraseña para el usuario root de phpMyAdmin, ingresamos la contraseña y damos clic en Aceptar.

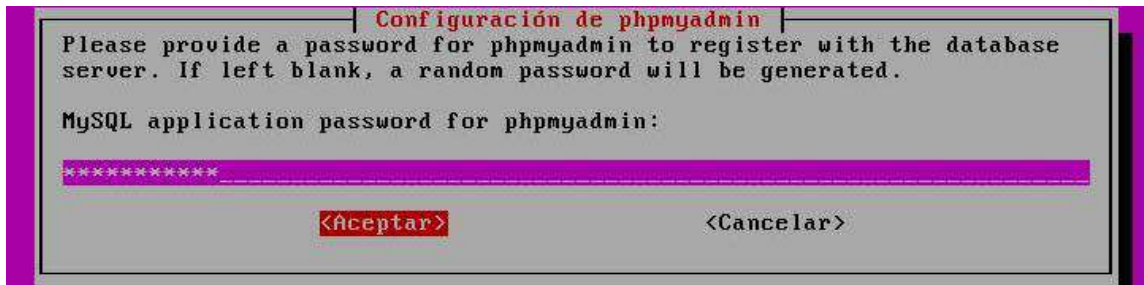


Figura N°18. Instalación MySQL
Fuente: www.digitalocean.com/community/mysql

Vamos a acceder a la plataforma de phpMyAdmin ingresándola siguiente sintaxis. http://Direccion_Ip/phpmyadmin. Al ingresar esto en la dirección veremos que se despliega la siguiente ventana:



Figura N°19. Interfaz de autenticación PHPMyAdmin
Fuente: maslinux.es/myadmin

Después de una autenticación exitosa, ahora podemos acceder a la interfaz *phpMyAdmin* y usarla para administrar las bases de datos.[13]

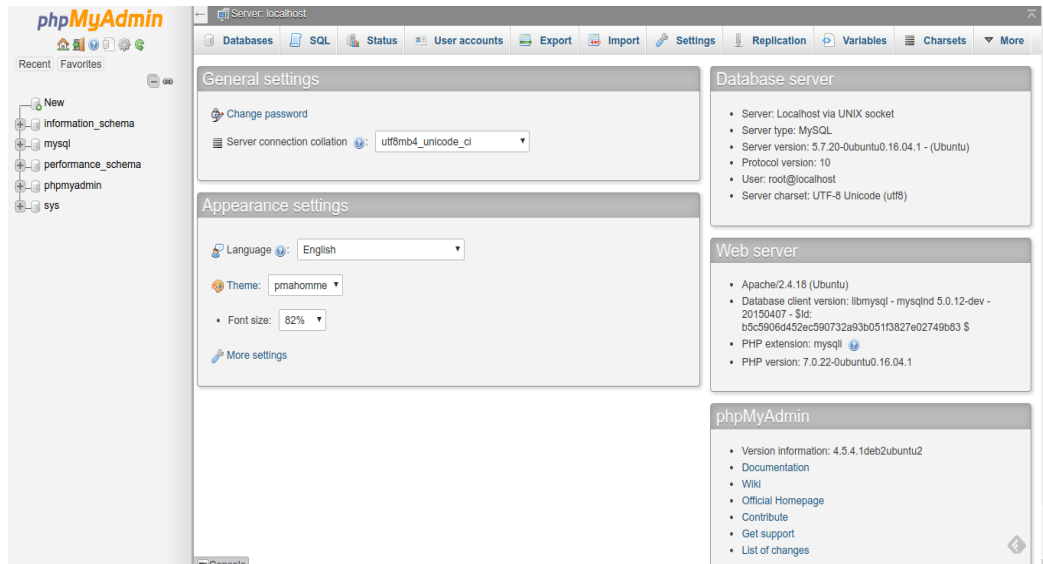


Figura N°20. Interfaz PHPMyAdmin

Fuente: maslinux.es/myadmin

2.4.5 MODELO ENTIDAD RELACION DE UNA BASE DE DATOS

Este modelo es solo y exclusivamente un método del que disponemos para diseñar estos esquemas que posteriormente debemos de implementar en un gestor de BBDD (bases de datos). Este modelo se representa a través de diagramas y está formado por varios elementos.

Este modelo habitualmente, además de disponer de un diagrama que ayuda a entender los datos y como se relacionan entre ellos, debe de ser completado con un pequeño resumen con la lista de los atributos y las relaciones de cada elemento.

2.4.5.1 ELEMENTOS DEL MODELO ENTIDAD RELACIÓN

ENTIDAD

La entidad es cualquier clase de objeto o conjunto de elementos presentes o no, en un contexto determinado dado por el sistema de información o las funciones y procesos que se definen en un plan de automatización. Dicho de otra forma, las entidades las constituyen las tablas de la base de datos que permiten el almacenamiento de los ejemplares o registros del sistema, quedando recogidos bajo la denominación o título de la tabla o entidad. Por ejemplo, la entidad usuarios guarda los datos personales de los usuarios de la biblioteca, la entidad catalogo registra todos los libros catalogados, la entidad circulación todos los libros prestados y devueltos y así sucesivamente con todos los casos.

ATRIBUTO

Son las características, rasgos y propiedades de una entidad, que toman como valor una instancia particular. Es decir, los atributos de una tabla son en realidad sus campos descriptivos, el predicado que permite definir lo que decimos de un determinado sujeto. Por ejemplo de una entidad o tabla catálogo, se pueden determinar los atributos título, subtítulo, título paralelo, otras formas del título, autor principal, otras menciones de responsabilidad, edición, mención de edición, editorial, lugar de publicación, fecha de publicación, etc.

RELACION

Vínculo que permite definir una dependencia entre los conjuntos de dos o más entidades. Esto es la relación entre la información contenida en los registros de varias tablas. Por ejemplo, los usuarios suelen clasificarse según una lista de tipos de usuarios, ya sean profesores, alumnos o investigadores. De esta forma es posible emitir la relación entre el usuario Jorge Martínez como alumno y Enrique Valtierra como profesor. Las relaciones son definidas de forma natural en un diagrama relacional para expresar un modelo cognitivo que dará lugar posteriormente a las interrelaciones de las entidades.

CARDINALIDAD

La cardinalidad se representan en un diagrama ER como una etiqueta que se ubica en ambos extremos de la línea de relación de las entidades y que puede contener diversos valores entre los que destacan comúnmente el 1 y el M (muchos), obteniendo los siguientes tipos:

- **Relación 1 a 1.** La relación uno a uno, define que un único registro de la tabla puede estar relacionado con un único registro de la tabla relacionada.

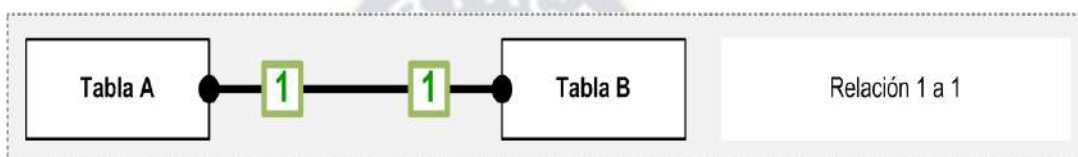


Figura N°21. Esquema de relación 1 a 1
Fuente: basesdedatos.blogspot.com

- **Relación 1 a M.** La relación de uno a varios, define que un registro dado de una tabla auxiliar o secundaria sólo puede estar vinculado con un único registro de la tabla principal con la que está relacionada.

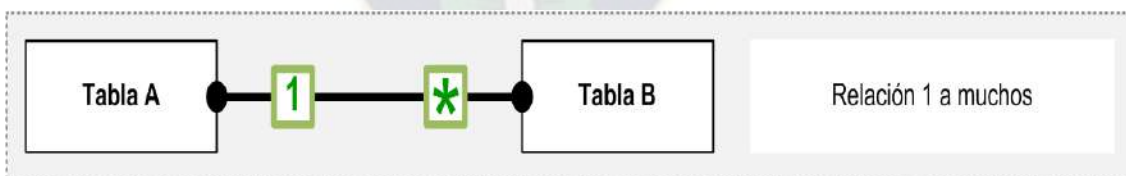


Figura N°22. Esquema de relación 1 a M
Fuente: basesdedatos.blogspot.com

- **Relación * a *.** La relación de varios a varios, define que un registro de una tabla puede estar relacionado con varios registros de la tabla relacionada y viceversa.

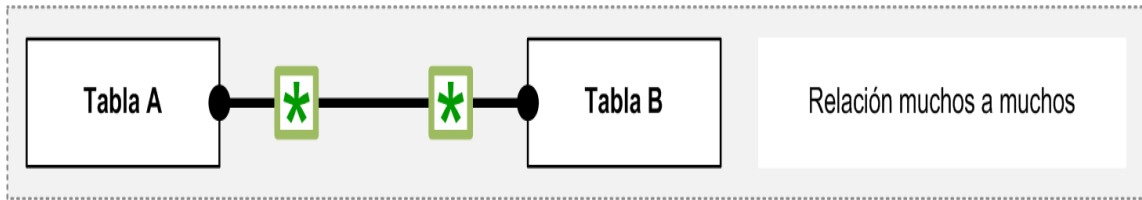


Figura N°23. Esquema de relación M a M
Fuente: basesdedatos.blogspot.com

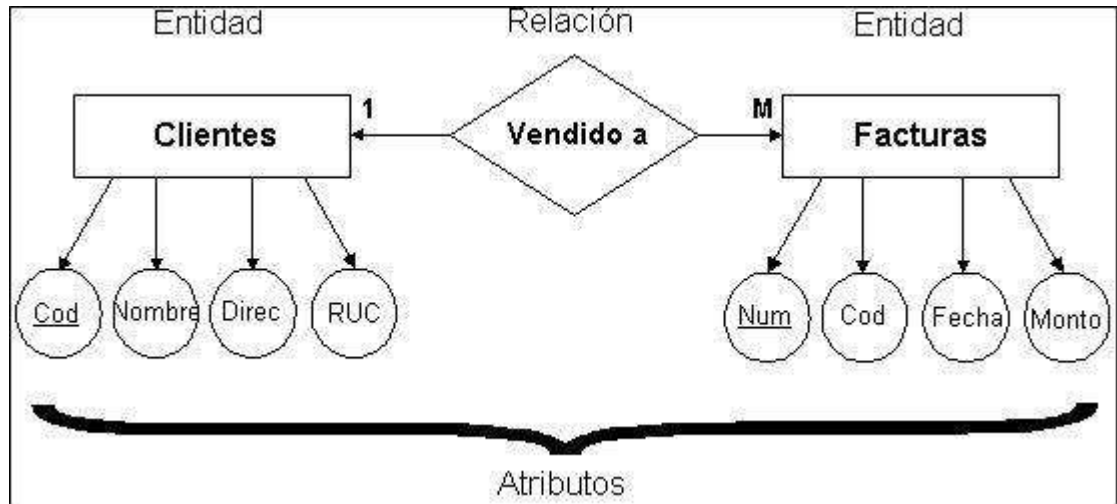


Figura N°24. Elementos de una Entidad-Relación
Fuente: basesdedatos.blogspot.com

2.4.6 LEGUAJES DE PROGRAMACIÓN PRINCIPALES PARA EL FRONT-END

2.4.6.1 HTML5

HTML es un lenguaje de etiquetas que se utiliza para el desarrollo de páginas de Internet. Se trata de las siglas que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto

HTML5 es la quinta revisión del lenguaje HTML. Esta nueva versión (aún en desarrollo), y en conjunto con CSS3, define los nuevos estándares de desarrollo web, rediseñando el código para resolver problemas y actualizándolo así a nuevas necesidades. No se limita solo a crear nuevas etiquetas o atributos, sino que incorpora

muchas características nuevas y proporciona una plataforma de desarrollo de complejas aplicaciones web.

La estructura principal de una página HTML es:

- Cada página comienza con: **< HTML >**
- A continuación viene la cabecera, delimitada por **< HEAD >** y **< /HEAD >**
- Después, el comando **< BODY >**, que indica el comienzo del cuerpo de la página. Las instrucciones HTML se escribirán a continuación, y finalizarán con **< /BODY >**.
- La página acabará con **< /HTML >**.

Es decir:

`<HTML>`

`<HEAD>`

Definiciones de la cabecera

`</HEAD>`

`<BODY>`

Instrucciones HTML

`</BODY>`

`</HTML>`

2.4.6.2 CSS

CSS (Cascading Style Sheets, u Hojas de Estilo en Cascada) es la tecnología desarrollada por el World Wide Web Consortium (W3C) con el fin de separar la estructura de la presentación. A pesar de que la recomendación oficial del grupo de trabajo de la W3C ya había alcanzado la estabilidad requerida para que fuera soportada por los principales navegadores comerciales, como Netscape e Internet Explorer, tan tempranamente como en el año 1998, la situación de entonces, comúnmente conocida como la “guerra de los navegadores”, hacía que los intereses comerciales de las dos compañías en lucha por el mercado de usuarios de Internet se interpusieran en el camino de las CSS.

La filosofía de CSS se basa en intentar separar lo que es la estructura del documento HTML de su presentación. Por decirlo de alguna manera: la página web sería lo que hay debajo (el contenido) y CSS sería un cristal de color que hace que el contenido se vea de una forma u otra. Usando esta filosofía, resulta muy fácil cambiarle el aspecto a una página web: basta con cambiar “el cristal” que tiene delante. Piensa por ejemplo qué ocurre si tienes un libro de papel y lo miras a través de un cristal de color azul: que ves el libro azul. En cambio, si lo miras a través de un cristal amarillo, verás el libro amarillo. El libro (el contenido) es el mismo, pero lo puedes ver de distintas maneras.

2.4.6.3 JAVASCRIPT

Javascript es un lenguaje de programación que surgió con el objetivo inicial de programar ciertos comportamientos sobre las páginas web, respondiendo a la interacción del usuario y la realización de automatismos sencillos. En ese contexto podríamos decir que nació como un "lenguaje de scripting" del lado del cliente, sin embargo, hoy Javascript es mucho más. Las necesidades de las aplicaciones web modernas y el HTML5 ha provocado que el uso de Javascript que encontramos hoy haya llegado a unos niveles de complejidad y prestaciones tan grandes como otros lenguajes de primer nivel.

Pero además, en los últimos años Javascript se está convirtiendo también en el lenguaje "integrador". Lo encontramos en muchos ámbitos, ya no solo en Internet y la Web, también es nativo en sistemas operativos para ordenadores y dispositivos, del lado del servidor y del cliente. Aquella visión de Javascript "utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web" se ha quedado muy pequeña.

A Javascript se le denomina "del lado del cliente" porque **donde se ejecuta es en el navegador (cliente web)**, en contraposición a lenguajes como PHP que se ejecutan del "lado del servidor". En el lado que nos ocupa con Javascript, el cliente, es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con todos los

navegadores modernos se ha convertido en un estándar como lenguaje de programación del lado del cliente.

Es la tercera capa del pastel de los estándares en las tecnologías para la web, dos de las cuales son (HTML y CSS)

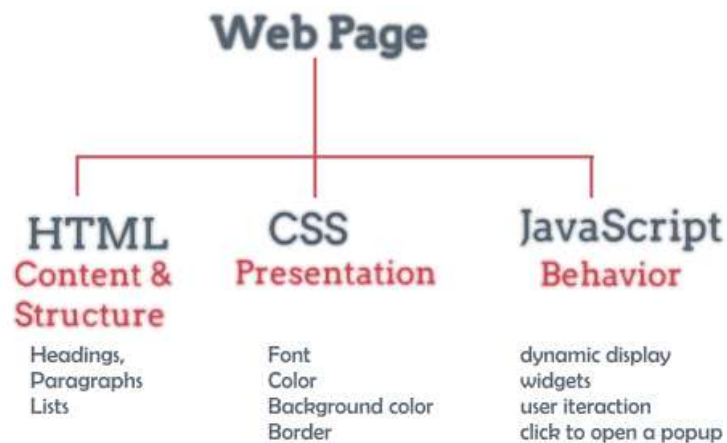


Figura N°25. Estándar de una página web dinámica e interactiva

Fuente: desarrolloweb.com

2.5 SERVIDOR PRIVADO VIRTUAL VPS

Un servidor virtual privado (VPS, por sus siglas en inglés) hace referencia a una parte de un servidor físico, aunque cuenta con un funcionamiento autónomo e independiente, pese a compartir dicho servidor material con otros servidores virtuales. Por tanto, un VPS se encuentra a mitad camino entre un servidor dedicado y un alojamiento compartido, ya que trabaja como una máquina privada para un cliente, pero comparte espacio físico con otros VPS.

Un servidor virtual forma parte de un servidor físico, que se encuentra dividido en diferentes servidores virtuales. Esta partición permite a cada VPS trabajar para un cliente como si de una sola máquina se tratase, con su propio disco duro, memoria RAM y velocidad de transmisión, entre otros aspectos. Sin embargo, el servidor real es compartido con otros clientes del resto de VPS alojados en él. [14]

Por tanto, el funcionamiento de un VPS es independiente y autónomo, de tal forma que opera como un servidor dedicado, con todos sus recursos destinados a un cliente concreto.

2.5.1 PROVEEDOR DE VPS “DigitalOcean”

DigitalOcean es un proveedor estadounidense de servidores virtuales privados, con sede principal en la ciudad de Nueva York.

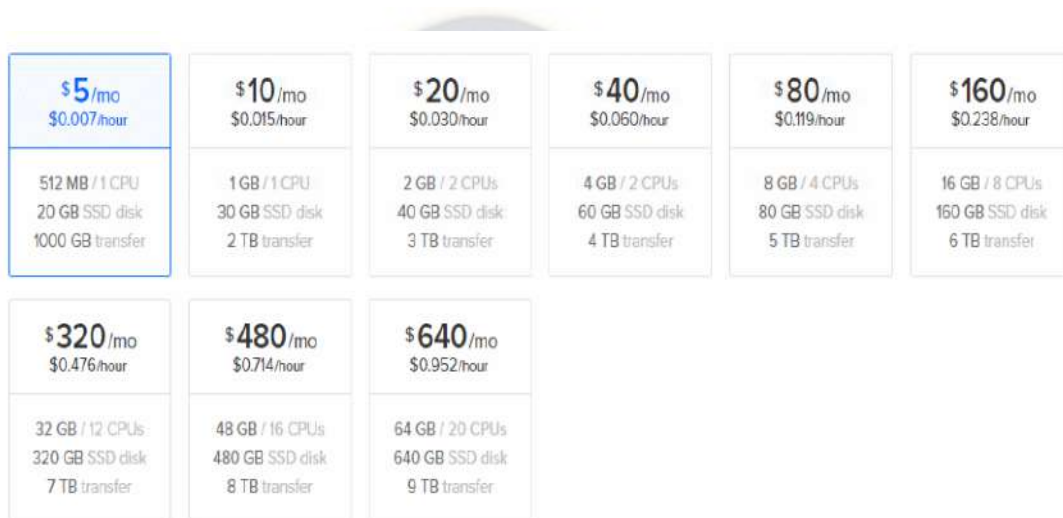
DigitalOcean maneja el concepto de droplet para designar a cada uno de los servidores virtuales, exactamente servidores virtuales privados, los cuales ofrecen en alquiler. Son privados porque DigitalOcean no interviene en nada en su instalación y manejo, limitándose a ofrecer imágenes de los principales sistemas operativos junto con sus repositorios de manera local, lo cual ahorra tiempo y transporte de datos a la larga gracias a los repositorios locales. Además cada droplet de un mismo cliente puede comunicarse única y exclusivamente con otros droplets del mismo cliente por medio de una o más tarjetas de red creando así una red privada virtual de manera local (por ejemplo un *droplet* con un servidor MySQL y otro droplet con Wordpress instalado y dos tarjetas de red virtuales, una para conectar a la base de datos y otra expuesta a la internet).

Tampoco ofrecen venta de dominios web sino que ofrecen la interfaz web necesaria para que los clientes agreguen, de ser necesario, los dominios web que hayan adquirido a los registros de los DNS de DigitalOcean.

DigitalOcean afirma que sus droplets, su término para llamar a los servidores en la nube, pueden ser provisionados típicamente en 55 segundos y como máximo dos minutos; inicialmente con una dirección IPv4 fija. Para ello la compañía ofrece discos duro SSD para mayor rapidez. El primer paso será elegir una distribución para nuestro servidor. Con el tiempo, Digital Ocean va añadiendo nuevos sistemas, pero de momento, podemos elegir entre **Ubuntu, FreeBSD, Fedora, Debian, CoreOS y CentOS**. [14]

2.5.1.1 RECURSOS DEL SERVIDOR

Una vez elegido lo que queremos instalar, debemos elegir **los recursos** de nuestro servidor. Esta siempre es una elección compleja, que depende de lo que usamos, el tráfico que tengamos, etc. Sin embargo, Digital Ocean es escalable verticalmente, por lo que en cualquier momento, podemos cambiar de plan sin tener que volver a crear otro diferente, lo que lo hace muy flexibles. [14]



\$5/mo \$0.007/hour	\$10/mo \$0.015/hour	\$20/mo \$0.030/hour	\$40/mo \$0.060/hour	\$80/mo \$0.119/hour	\$160/mo \$0.238/hour
512 MB / 1 CPU 20 GB SSD disk 1000 GB transfer	1 GB / 1 CPU 30 GB SSD disk 2 TB transfer	2 GB / 2 CPUs 40 GB SSD disk 3 TB transfer	4 GB / 2 CPUs 60 GB SSD disk 4 TB transfer	8 GB / 4 CPUs 80 GB SSD disk 5 TB transfer	16 GB / 8 CPUs 160 GB SSD disk 6 TB transfer
\$320/mo \$0.476/hour	\$480/mo \$0.714/hour	\$640/mo \$0.952/hour			
32 GB / 12 CPUs 320 GB SSD disk 7 TB transfer	48 GB / 16 CPUs 480 GB SSD disk 8 TB transfer	64 GB / 20 CPUs 640 GB SSD disk 9 TB transfer			

Figura N°26. Planes que ofrece DigitalOcean
Fuente: digitalocean.com



CAPITULO III

INGENIERÍA DE PROYECTO

3.1 INTRODUCCIÓN

En el capítulo anterior se describe las características de cada uno de los subsistemas del dispositivo, conceptos teóricos estrechamente relacionados con el desarrollo del prototipo tanto en hardware y software, en este capítulo se verá el desarrollo de la programación de la tarjeta de desarrollo y a la interfaz gráfica que será utilizada por el usuario final (software). El esquema del sistema se muestra en la Figura N° 27.

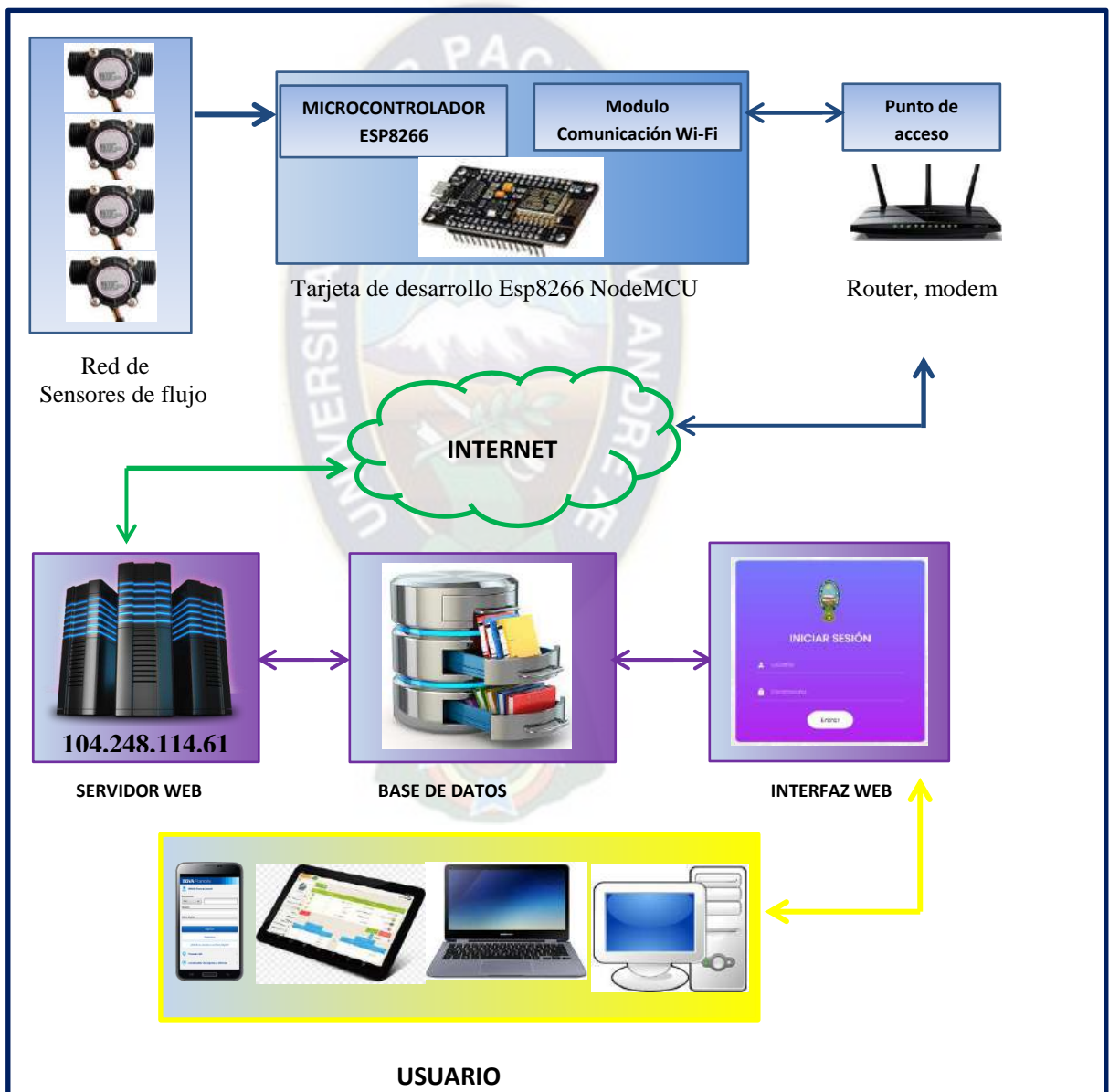


Figura N°27. Diagrama de bloques del sistema

Fuente: Elaboración propia

3.2 PROGRAMACIÓN DEL HARDWARE

Para poder programar el NodeMCU en el entorno de Arduino, es necesario instalar una librería para el desarrollo con módulos ESP8266.

3.2.1 INTSLACIÓN DE LA LIBRERÍA DEL ESP8266 PARA ARDUINO

Se instaló el Arduino IDE con versión 1.7 o superior, este paso es necesario para que nuestra IDE de arduino reconozca a nuestro ESP8266 como una tarjeta de desarrollo.

Para la instalaicón vamos a **Archivo** en la pestaña **Preferencias** y en la casilla “Gestor de URLs Adicionales de Tarjetas” agregamos el siguiente enlace:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

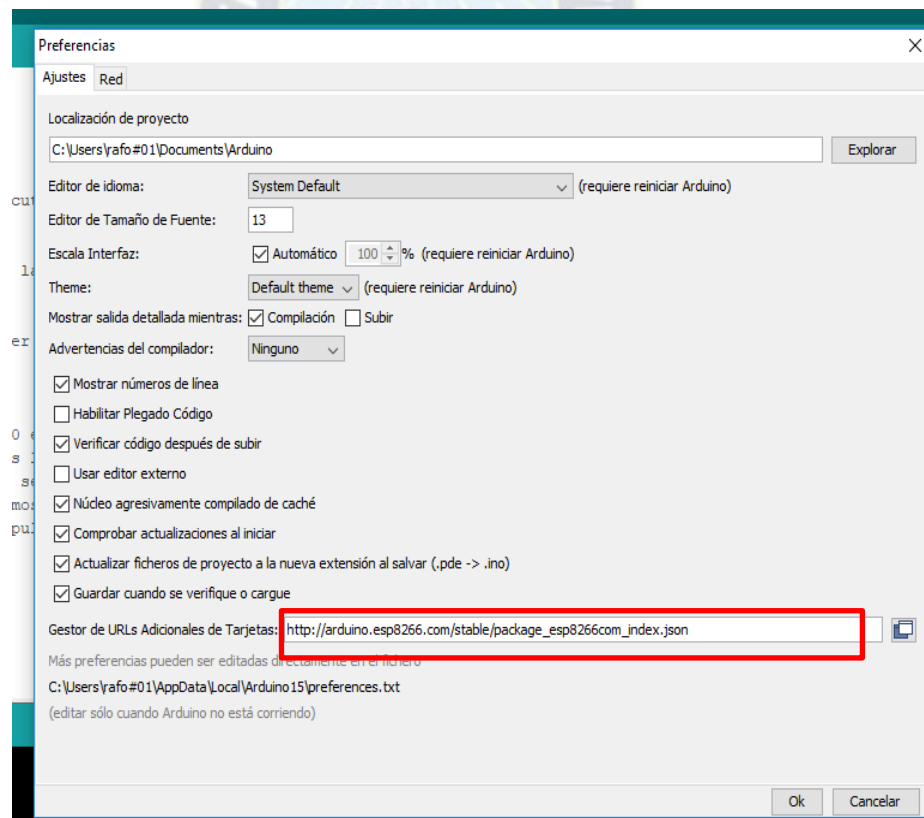


Figura N°28. Instalación librería del esp8266

Fuente: Elaboración propia

Luego vamos a **Herramientas > Placa > Gestor de tarjetas** y buscamos ESP8266 para instalar la última versión de la comunidad ESP8266.

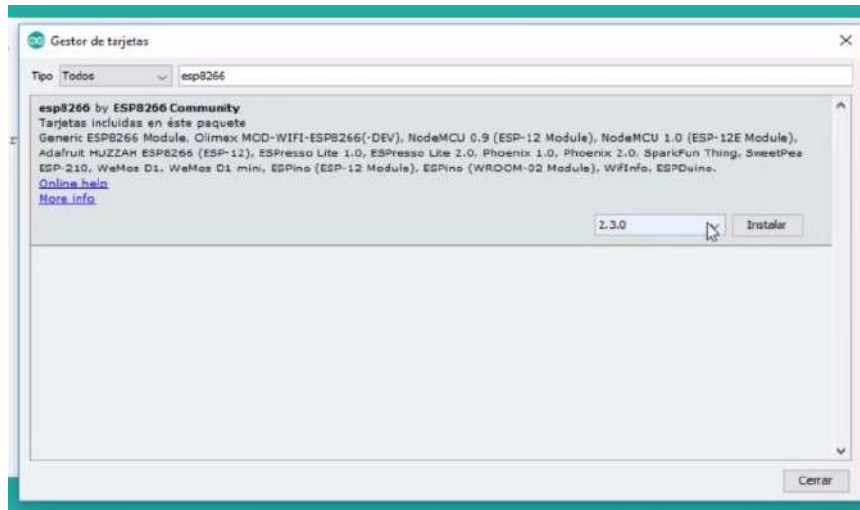


Figura N°29. Instalación librería del esp8266

Fuente: Elaboración propia

Se configura la placa en **Herramientas > Placa > NodeMCU 1.0** y selecciona el puerto COM donde conectamos el NodeMCU.

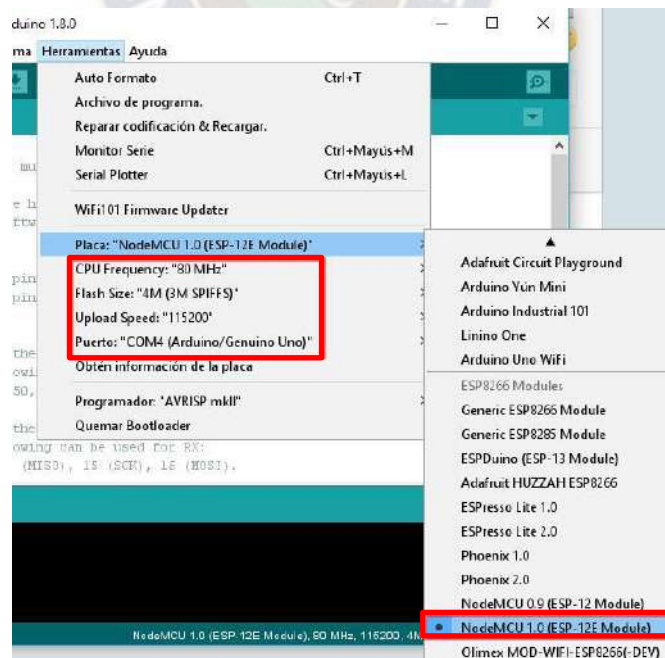


Figura N°30. Selección de la placa NodeMCU

Fuente: Elaboración propia

3.2.2 DETALLES TÉCNICOS DEL CAUDALÍMETRO

El sensor está diseñado en base de plástico sólido, es adecuado para un tubo estándar de 1/2 pulgada y de 3/4 de pulgada y se puede insertar fácilmente en un sistema de tuberías estándar.



Figura N°31. Caudalímetro FS300A

Fuente: Elaboración propia

Datos técnicos

- Código: FS300A
- Tipo de sensor: Hall effect
- Tensión de operación: 3.5V - 18V DC
- Corriente de operación: 15mA (5V)
- Temperatura de funcionamiento: -25°C a 80°C
- Humedad de trabajo Rango: 35% -80% de humedad relativa
- Precisión: $\pm 2\%$
- Modo de detección: Vertical
- Tiempo de subida de la salida: 0.04us
- Tiempo de caída de la salida: 0.18us
- Presión de funcionamiento máximo: 1.2MPa (12 bar)
- Rango de funcionamiento: 1-60L/min
- 3/4" conexiones nominales de tubería



Figura N°32. Caudalímetro YF-S201

Fuente: Elaboración propia

Datos técnicos

- Sensor de Flujo YF-S201.
- Tipo de sensor: Hall effect
- Voltaje de funcionamiento: 3.5 a 18 V DC
- Max consumo de corriente: 15 mA a 5 V
- Trabajo Caudal: de 1 a 30 litros / minuto
- Temperatura de funcionamiento: -25 a 80 °C
- Humedad de trabajo Rango: 35% -80% de humedad relativa
- Precisión: $\pm 2\%$
- Rango de flujo: 1-30L/min
- Modo de detección: Vertical
- Presión máxima del agua: 2,0 MPa
- Tiempo de subida de la salida: 0.04us
- Tiempo de caída de la salida: 0.18us
- 1/2" conexiones nominales de tubería

3.2.3 CALCULO DEL CAUDAL Y VOLUMEN

Para calcular el caudal usamos un factor de conversión y una formula provistos por el fabricante. Este factor de conversión sólo nos servirá de referencia, ya que en la mayoría de los casos deberemos ajustarlo realizando pruebas.

$$\text{Fórmula: } f(\text{Hz}) = K * Q(\text{L/min}) \quad \dots\dots\dots(1)$$

Donde f es la frecuencia de pulsos, K es el factor de conversión y Q es el caudal en litros por minuto

Los distintos modelos de caudalímetros vienen por defecto con un factor de conversión. En este proyecto se usan dos modelos distintos de caudalímetros, mas específicos los modelos YF-S201 y el FS300A por ello se realizó la calibración de cada uno de los modelos para obtener un nuevo factor de conversión y con ello obtener una medida del caudal mas precisa, con la siguietne formula.

$$\text{Factor de conversión: } K = \frac{\text{n}^\circ \text{ Pulsos}}{\text{Volumen} * 60} \quad \dots\dots\dots(2)$$

- **Codigo para obtener numero de pulsos por litro**

```
volatile long NumPulsos;  
int PinSensor = 6;
```

```
void ContarPulsos ()  
{  
  NumPulsos++;  
}
```

```
//*****Función para obtener los pulsos*****  
void setup()  
{
```

```
Serial.begin(XXXX);
pinMode(PinSensor, INPUT);
  attachInterrupt(digitalPinToInterrupt(PinSensor),ContarPulsos,RISING);

interrupts;
}

void loop ()
{

  Serial.print ("Numero de Pulsos = ");
  Serial.println (NumPulsos);
  delay(10);
}
```

- **Resultados obtenidos en monitor serie para cada modelo de caudalímetro**

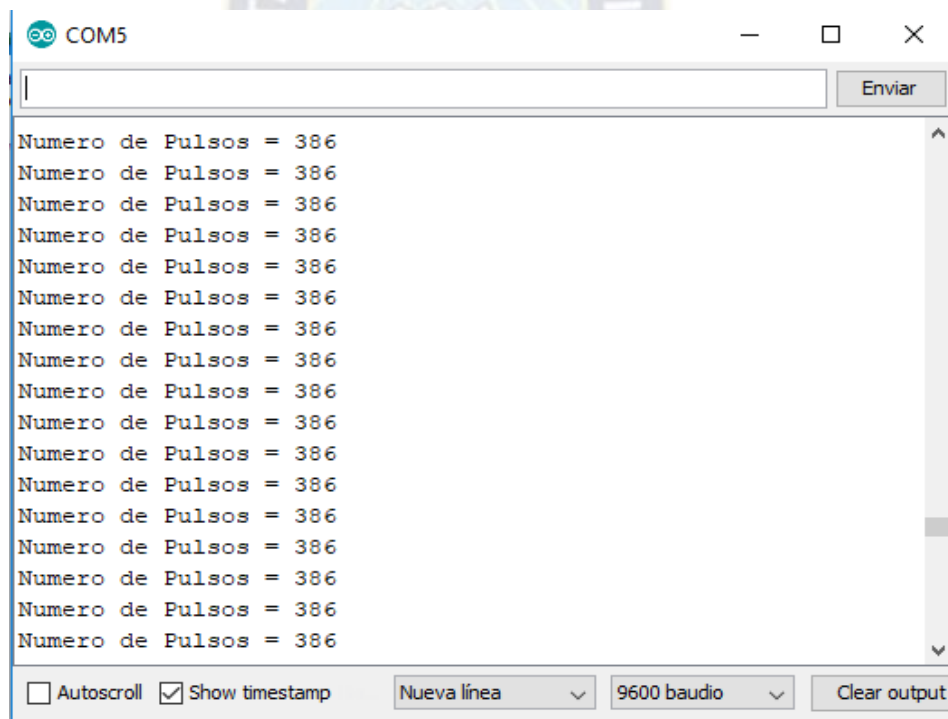


Figura N°33. Número de pulsos por litro del modelo YF-S201

Fuente: Elaboración propia

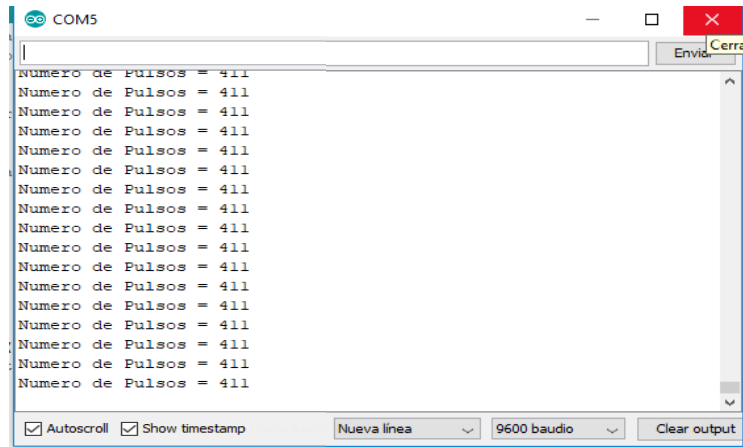


Figura N°34. Número de pulsos por litro del modelo **FS300A**
Fuente: Elaboración propia

Se tomó 8 muestras para cada modelo de las cuales se obtuvo un promedio para obtener el factor de conversión K

Litros	Pulsos	K
1	386	6.43
1	387	6.45
2	777	6.48
3	1150	6.38
1	382	6.36
4	1550	6.40
1	389	6.48
4	1560	6.50
Promedio		K= 6.40

Tabla N°2. Factor de conversión K para el modelo YF-S201
Fuente: Elaboración propia

Litros	Pulsos	K
1	420	7,00
1	421	7,01
1	414	6,90
2	845	7,04
1	418	6,96
4	1675	6,97
1	415	6,91
3	1262	7,01
Promedio		K= 6.97

Tabla N°3. Factor de conversión K para el modelo FS300A
Fuente: Elaboración propia

- **Codigo para obtener el caudal Q y volumen V**

Una vez obtenido el factor de conversion K pasamos a calcular el caudal Q de la formula provista por el fabricante:

$$\text{Frecuencia } f(\text{Hz}) = K * Q(\text{L/min}) \Rightarrow Q(\text{L/min}) = \frac{f(\text{Hz})}{K} \dots\dots\dots(3)$$

Teniendo el caudal podemos medir el volumen de agua, puesto que el caudal es la variación del volumen con respecto al tiempo, con estos datos podemos obtener el volumen con:

$$\text{Caudal } Q = \frac{\Delta V}{\Delta t} \dots\dots\dots(4)$$

Deduciendo de la formula (4) obtenemos lo siguiente :

$$\text{Volumen } V = V_0 + Q * \Delta t \dots\dots\dots(5)$$

Con esta fromula (5)obtenemos el volumen acumulado

Para realizar ambos calculos usamos el siguiente programa:

```

//*****

volatile long NumPulsos;
int PinSensor = 4;
float factor_conversion=6.44;
float volumen=0;
long dt=0;
long t0=0;

void ContarPulsos ()
{
  NumPulsos++;
}

int ObtenerFrecuecia()

```

```

{
  int frecuencia;
  NumPulsos = 0;
  interrupts();
  delay(1000);
  noInterrupts();
  frecuencia=NumPulsos;
  return frecuencia;
}

void setup()
{
  Serial.begin(XXXX);
  pinMode(PinSensor, INPUT);
  attachInterrupt(digitalPinToInterrupt(PinSensor),ContarPulsos,RISING);

  Serial.println ("restablecer el volumen a 0 Litros");
  t0=millis();
}

void loop ()
{
  Serial.available();
  if (Serial.available()) {
    if (Serial.read() == 'r'){ volumen = 0; }
  }
  float frecuencia=ObtenerFrecuecia();
  float caudal_Lm=frecuencia/factor_conversion;
  dt=millis()-t0;
  t0=millis();
  volumen=volumen+(caudal_Lm/60)*(dt/1000);
  Serial.print ("Caudal: ");
  Serial.print (caudal_Lm,3);
  Serial.print ("L/min\tVolumen: ");
  Serial.print (volumen,3);
  Serial.println (" L");
}

```

En el monitor serie observamos la variación del caudal y el volumen acumulado

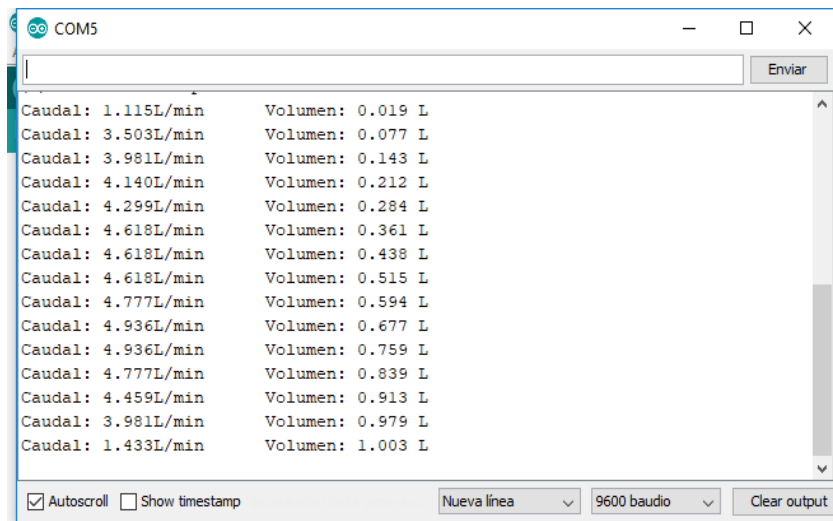


Figura N°35. Muestra del caudal y volumen acumulado
Fuente: Elaboración propia

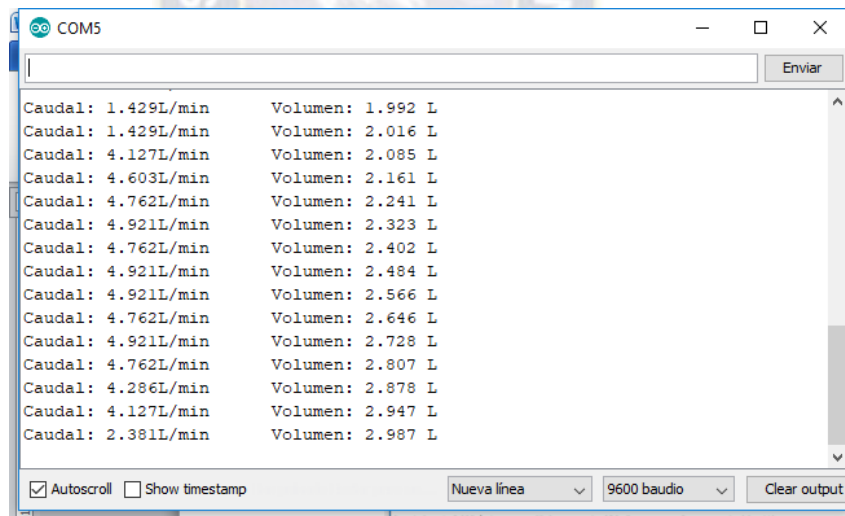


Figura N°36. Muestra del caudal y volumen acumulado
Fuente: Elaboración propia

Ahora es necesario que estos datos sean enviados a una base de datos donde serán almacenados para su posterior requerimiento.

- **Código para enviar los resultados obtenidos a una base de datos**

A continuación incluimos las siguientes líneas de código a todo el código anterior escrito. Esto nos permitirá enviar los datos obtenidos a una base de datos en MySQL.

```
//*****Librerias necesarias para el Esp8266*****
```

```
#include <ESP8266WiFi.h>
```

```
#include <WiFiClient.h>
```

```
//*****Adicionamos a las variables globales*****
```

```
int contconexion = 0;
```

```
const char *ssid = "*****";
```

```
const char *password = "*****";
```

```
char host[48];
```

```
String strhost = "192.168.1.12" ;
```

```
String strurl = "../enviardatos.php";
```

```
String chipid = "";
```

```
float m3 = 0;
```

```
float ac = 0;
```

```
unsigned long anterior_Millis = 0;
```

```
String enviardatos(String datos) {
```

```
    String linea = "error";
```

```
    WiFiClient client;
```

```
    strhost.toCharArray(host, 49);
```

```
    if (!client.connect(host, XX)) {
```

```
        Serial.println("Fallo de conexion");
```

```
        return linea;
```

```
    }
```

```
    client.print(String("POST ") + strurl + " HTTP/1.1" + "\r\n" +
```

```
        "Host: " + strhost + "\r\n" +
```

```
        "Accept: */*" + "\r\n" +
```

```
        "Content-Length: " + datos.length() + "\r\n" +
```

```

        "Content-Type: application/x-www-form-urlencoded" + "\r\n" +
        "\r\n" + datos);
delay(10);

Serial.print("Enviando datos a SQL...");

unsigned long timeout = millis();
while (client.available() == 0) {
  if (millis() - timeout > 5000) {
    Serial.println("Cliente fuera de tiempo!");
    client.stop();
  }
}

Serial.print("chipId: ");
chipid = String(ESP.getChipId());
Serial.println(chipid);

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED and contconexion < 50) {

  ++contconexion;
  delay(500);
  Serial.print(".");
}
if (contconexion < 50) {
  //para usar con ip fija
  IPAddress ip(192, 168, 1, 147);
  IPAddress gateway(192, 168, 1, 1);
  IPAddress subnet(255, 255, 255, 0);
  WiFi.config(ip, gateway, subnet);

  Serial.println("");
  Serial.println("WiFi conectado");
  Serial.println(WiFi.localIP());
}

```

```

else {
  Serial.println("");
  Serial.println("Error de conexion");
}
}

ac = (volumen / 1000);    //almacenamos volumen dividido entre 1000
if (volumen >= 1000) {
  m3 = m3 + 1;    // acumulamos en m3
  volumen = 0;
  ac = 0;
}

if (caudal_L_m != 0) {
  if (actual_Millis - anterior_Millis >= 1000) {
    Serial.print ("Caudal: ");
    Serial.print (caudal_Lm, 3);
    Serial.print ("L/min\tVolumen: ");
    Serial.print (volumen, 0);
    Serial.print (" L\tConsumo:");
    Serial.print (m3 + ac, 3);
    Serial.println (" M3");

    enviardatos("chipid=" + chipid + "caudal=" + String(caudal_Lm, 3) +
"litros=" + String(volumen, 0) + "consumo_M3=" + String(m3 + ac, 3));
    delay(50);
  }
}
}

```

3.3 IMPLEMENTACIÓN DE LA BASE DE DATOS

Se realiza el diseño e implementación de una base de datos que guarde la información registrada por el caudalímetro, también los datos de cada usuario, para así posteriormente poder realizar las consultas de acuerdo a requerimiento.

A continuación se muestra en la Figura N°37 la creación de una nueva base de datos en phpMyAdmin.

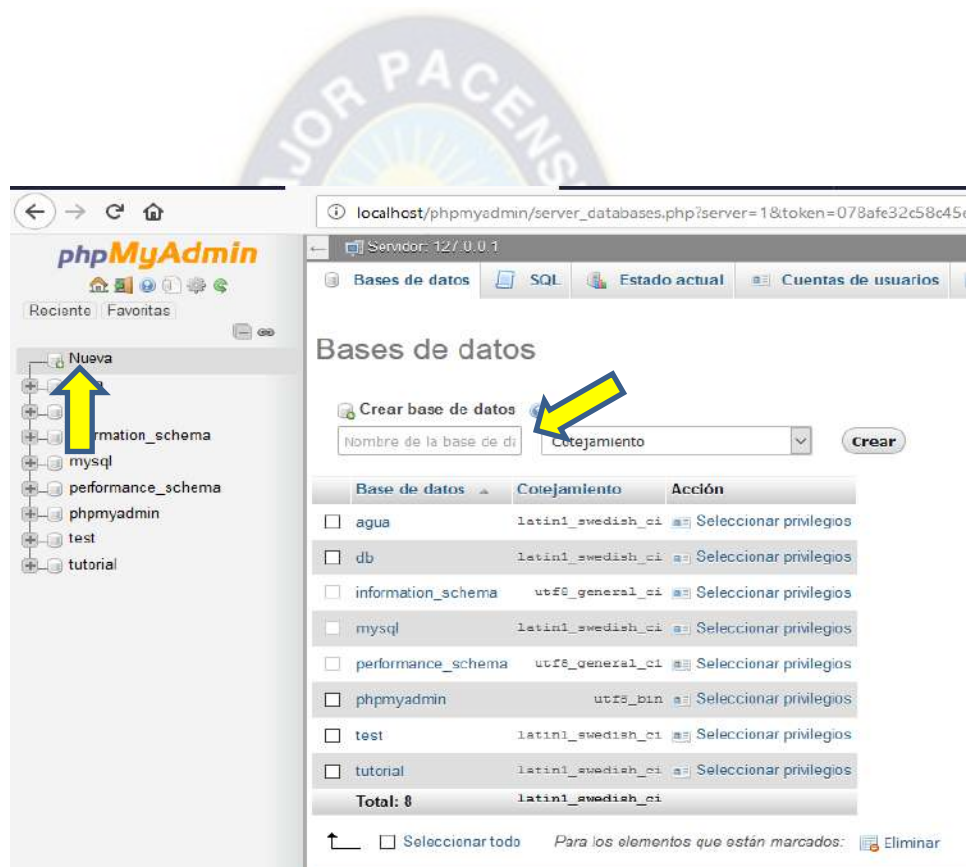


Figura N°37. Creación de una nueva Base de Datos
Fuente: Elaboración propia

Una vez creada la nueva base de datos, pasará a crearse las tablas que son necesarias para posteriormente poder realizar los cálculos de consumo de cada usuario registrado en la base de datos.

En la Figura 3.38 se observa la creación de la tabla registro, que se encargara de almacenar los datos provenientes de cada uno de las tarjetas de desarrollo.

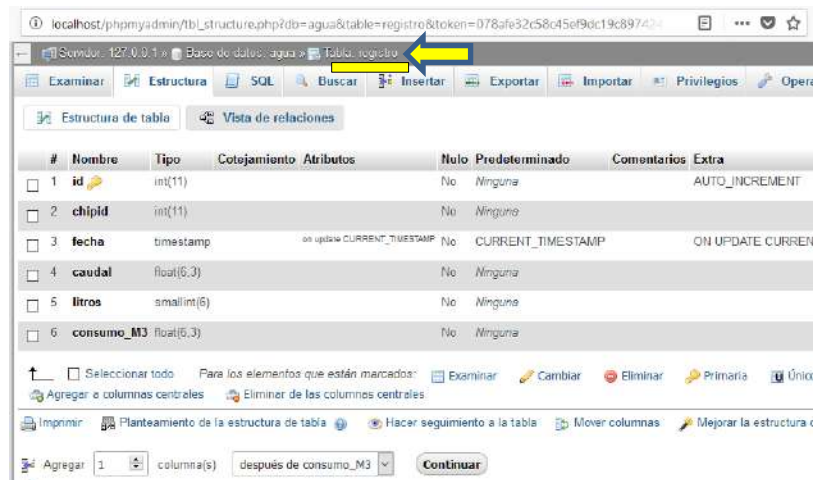


Figura N°38. Creación de la tabla registro
Fuente: Elaboración propia

En la Figura N°39 se observa la creación de la tabla usuario, que se encargara de almacenar los datos de todos los usuarios registrados.

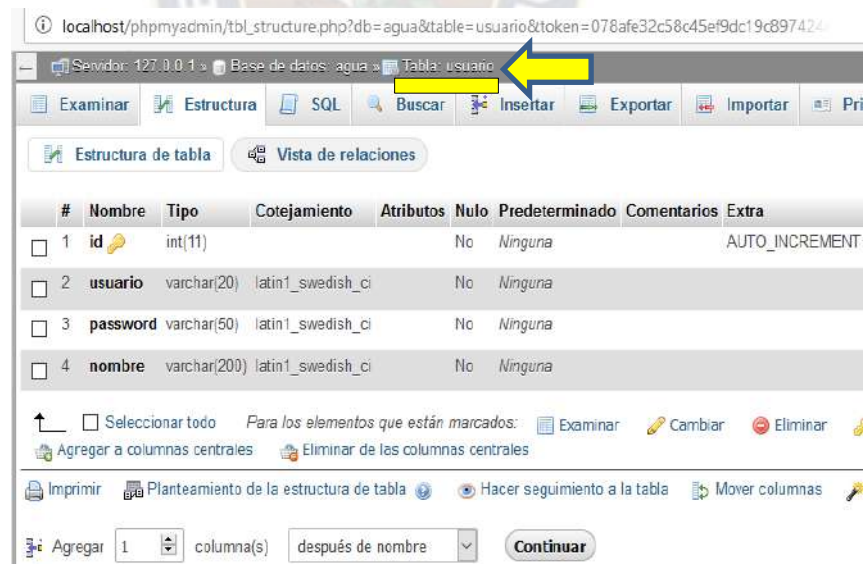


Figura N°39. Creación de la tabla usuario
Fuente: Elaboración propia

En la Figura N°40 se observa la creación de la tabla chip, que se encargara de almacenar los datos de descripción de cada tarjeta de desarrollo Esp8266NodeMCU.

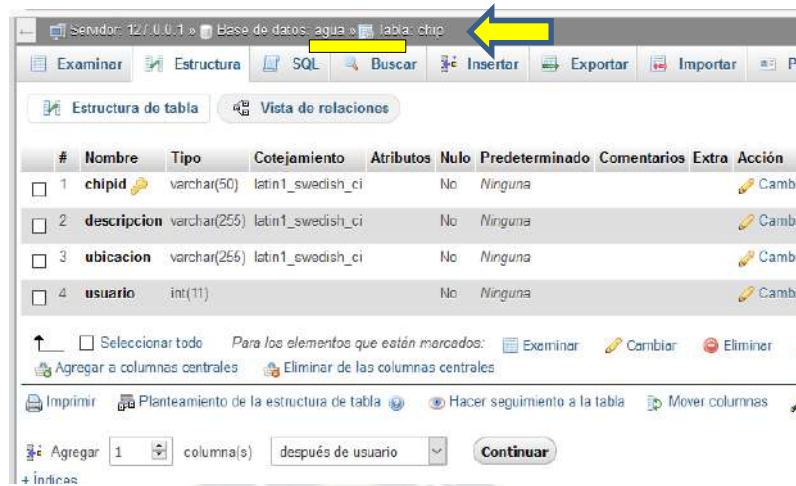


Figura N°40. Creación de la tabla chip
Fuente: Elaboración propia

3.3.1 MODELO ENTIDAD-RELACION DE LA BASE DE DATOS

El modelo Entidad-Relación o modelo E/R se lo realizó para el modelo de datos del sistema de información. Este modelo expresa entidades relevantes para el sistema de información así como sus propiedades. En él pueden verse los grupos de tablas y las relaciones que tienen con el resto de conjuntos de tablas.

A continuación se muestra el modelo entidad-relación de toda la base de datos anteriormente creada.



Figura N°41. Entidad relación de la base de datos
Fuente: Elaboración propia

3.4 RECEPCION Y ALAMCENAMIENTO EN LA BASE DE DATOS (enviardatos.php)

El siguiente código esta realizado en el lenguaje PHP, estará encargado de recibir y almacenar los datos que se enviaran a través del código realizado en la ide de arduino.

```
<?php
$conexion = mysql_connect('localhost', 'xxxxxxx', 'xxxxxxx');
//$conexion = mysql_connect('xxxxxx' 'xxxx', 'xxxx');
mysql_select_db('db', $conexion);
mysql_query("SET NAMES 'utf8'");

$chipid = $_POST ['chipid'];
$caudal = $_POST ['caudal'];
$litros = $_POST ['litros'];
$consumo_M3 = $_POST ['consumo_M3'];
mysql_query("INSERT INTO `db`.`registro` (`id`, `chipId`, `fecha`, `caudal`,
`litros`, `consumo_M3`) VALUES (NULL, '$chipid',
CURRENT_TIMESTAMP, '$caudal' '$litros' '$consumo_M3' );");

mysql_close();
echo "Datos ingresados correctamente.";
?>
```

A continuación realizamos las pruebas respectivas y las observamos a través del monitor serie de la ide de arduino, para corroborar que los datos se envíen de manera correcta a la base de datos.

En la Figura N°42 la correcta conexión a la red Wi-Fi y la captación del número de identificación de la placa de desarrollo Esp8266NodeMCU y mientras no haya flujo de agua el programa no enviara dato alguno.

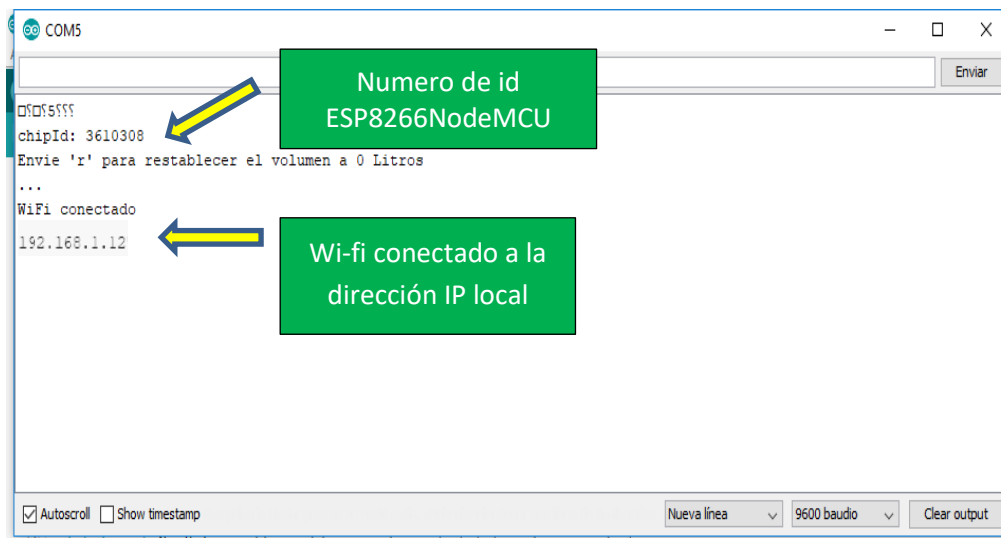


Figura N°42. Conexión a la red Wi-Fi
Fuente: Elaboración propia

Vertimos dos litros de agua a través del caudalímetro y observamos que registra y envía los datos registrados de manera correcta.

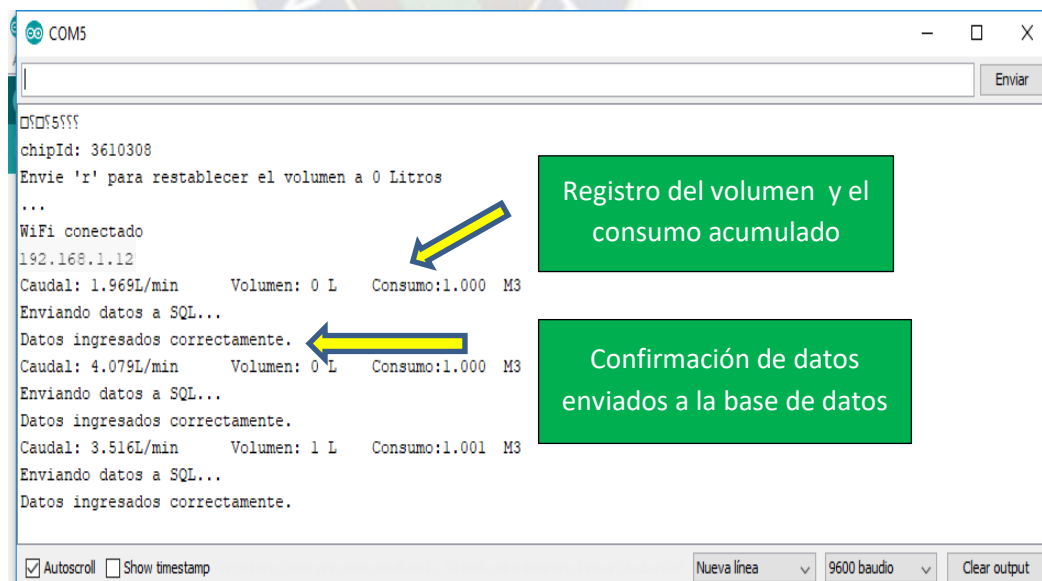


Figura N°43. Registro y envío de datos
Fuente: Elaboración propia

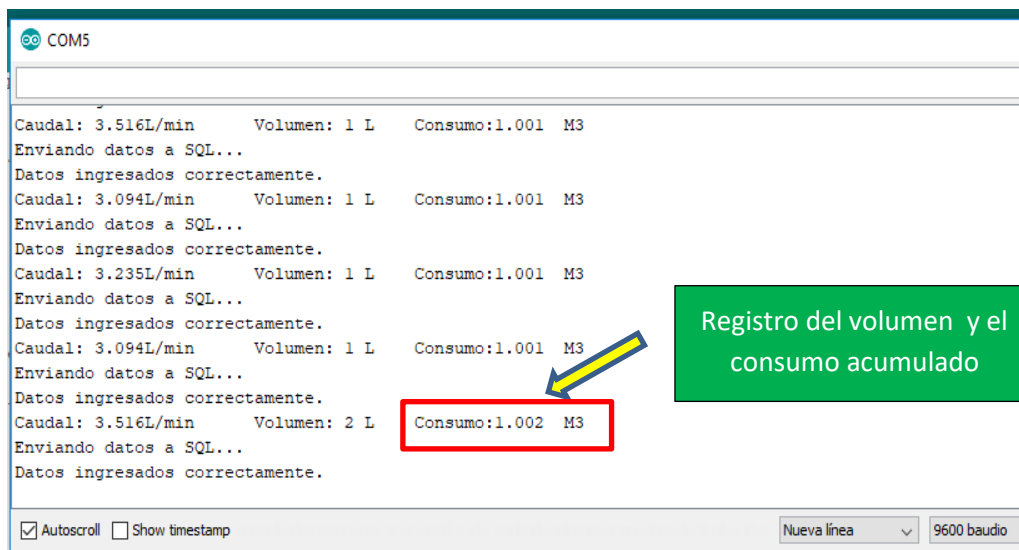


Figura N°44. Registro y envío de datos
Fuente: Elaboración propia

En la Figura N°45 podemos observar que los datos enviados a través de la placa de desarrollo a la base de datos fueron registrados de manera satisfactoria en la tabla registro anteriormente creada.

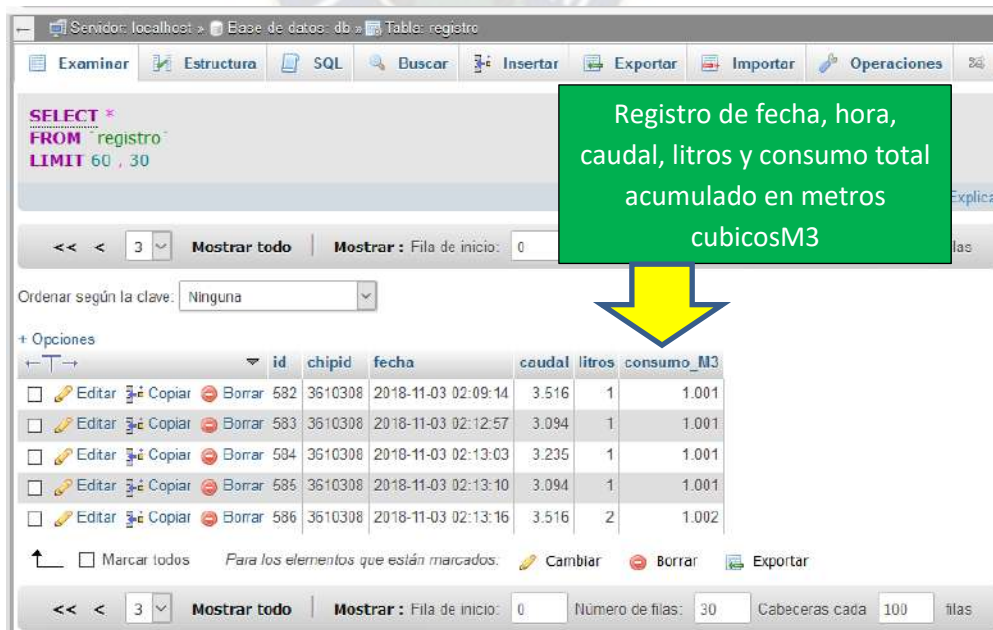


Figura N°45. Datos registrados en la base de datos
Fuente: Elaboración propia

3.5 INTERFAZ WEB DEL USUARIO

Esta parte de la programación está orientada al diseño de la interfaz web para el usuario y al almacenamiento de datos en la BD previamente diseñada, así también es donde se realizará los cálculos de consumo respectivo de cada usuario registrado y obtener un reporte de consumo final.

En primera instancia debemos ingresar al sistema con el nombre y contraseña asignados a cada usuario. En la Figura N°46, se muestra el diseño y la forma para el acceso al sistema de control y cálculo de consumo de agua.



Figura N°46. Login
Fuente: Elaboración propia

El ingreso podrá realizarlo como usuario o administrador, a continuación se muestra un diagrama de flujo al momento del ingreso.



Figura N°47. Diagrama de flujo para el Login
Fuente: Elaboración propia

La Figura N°48 muestra una vista general de la interfaz web que permite al usuario administrador generar varios eventos relacionados con la Base de Datos.

Los posibles eventos que genera cuando se ingresa en modo administrador son los siguientes:

- PAGINA PRINCIPAL
- REPORTE
- USUARIOS
- DISPOSITIVOS
- SALIR

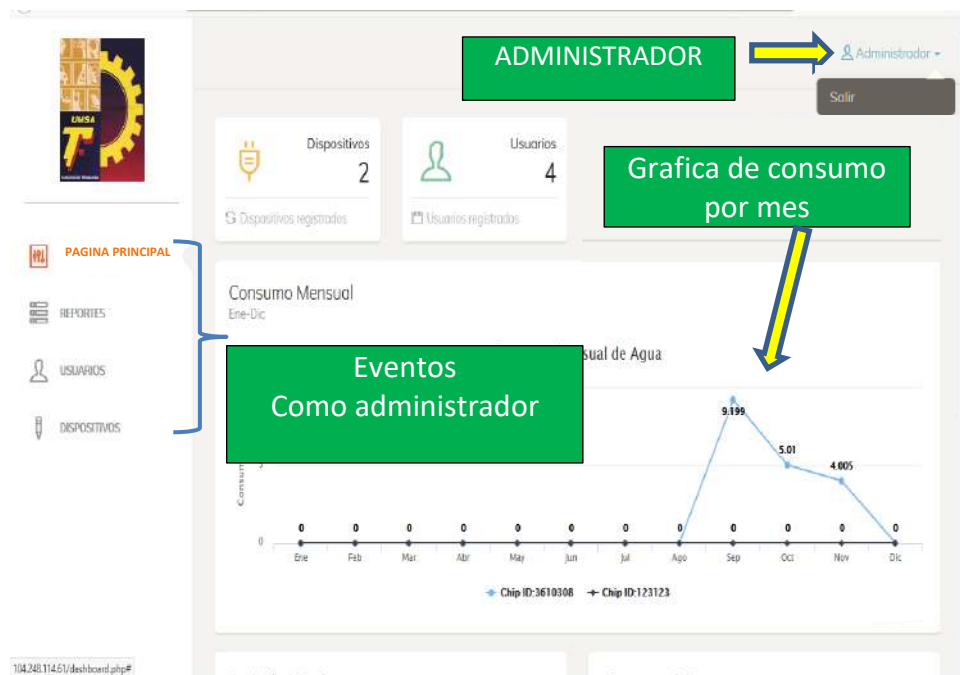


Figura N°48. Vista de la interfaz web del usuario como administrador
Fuente: Elaboración propia

A continuación procederemos a explicar de manera resumida la función de cada uno de los eventos:

- **PAGINA PRINCIPAL.-** Muestra una vista general del consumo de agua de un usuario determinado con una gráfica que señala el consumo por mes en metros cúbicos M3, así como también la última muestra de consumo capturada por la tarjeta de desarrollo
- **REPORTE.-** Es donde se procederá a realizar los cálculos de consumo de cada usuario y el monto a pagar por consumo.
- **USUARIO.-** Registra un nuevo usuario en la BD, también modifica y elimina a un usuario seleccionado.
- **DISPOSITIVO.-** Registra la Id de la tarjeta de desarrollo y algunos datos necesarios para ser asignado un determinado usuario.
- **SALIR.-** Nos permite salir de forma segura de la interfaz web.

3.5.1 REGISTRO DE NUEVO USUARIO

En la siguiente opción podremos agregar un nuevo usuario con los siguientes datos a registrar: Nombre de usuario, contraseña, nombre completo y la asignación entre usuario o administrador. Toda esta información será registrada y almacenada en la base de datos.

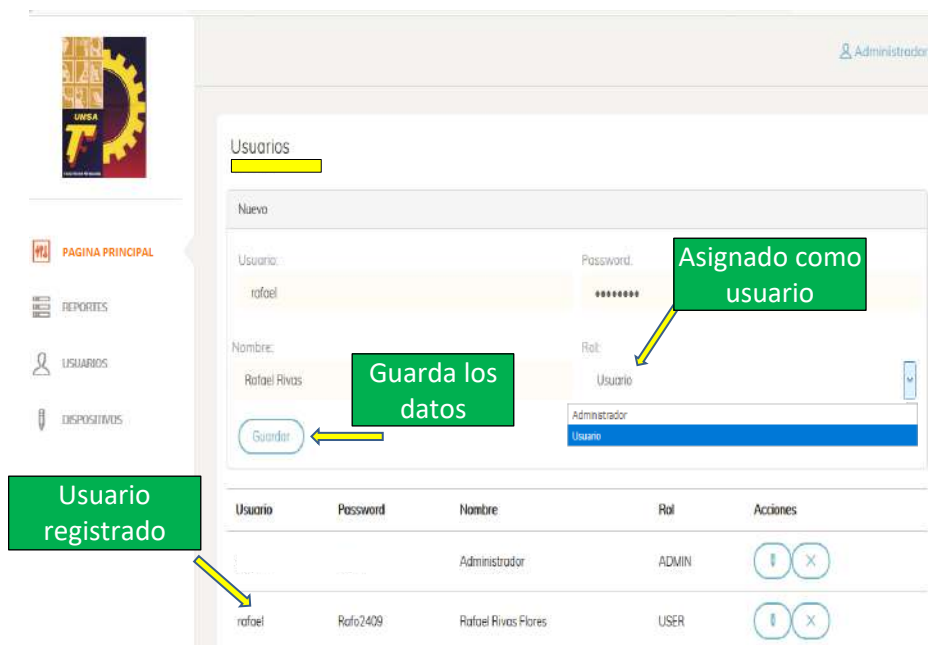


Figura N°49. Registro de nuevo usuario
Fuente: Elaboración propia

3.5.2 REGISTRO DE NUEVO DISPOSITIVO

En la siguiente opción es donde se registrara la Id de cada una de las tarjetas de desarrollo que estén en funcionamiento y algunos datos que son necesarios para mayor información respecto al lugar y donde está instalado el prototipo de control y registro de consumo de agua, la Id será vinculada a un determinado usuario, los datos a registrar son: Chip ID, descripción, ubicación, usuario y tarifa. Todo esto registrado y almacenado en la base de datos.



Figura N°50. Registro de nuevo dispositivo
Fuente: Elaboración propia

3.5.3 VERIFICACIÓN DE DATOS ALMACENADOS DE USUARIO Y DISPOSITIVO EN LA BASE DE DATOS

Después de haber registrado nuevos datos de usuario y dispositivo verificamos el correcto almacenamiento de estos en la base de datos.

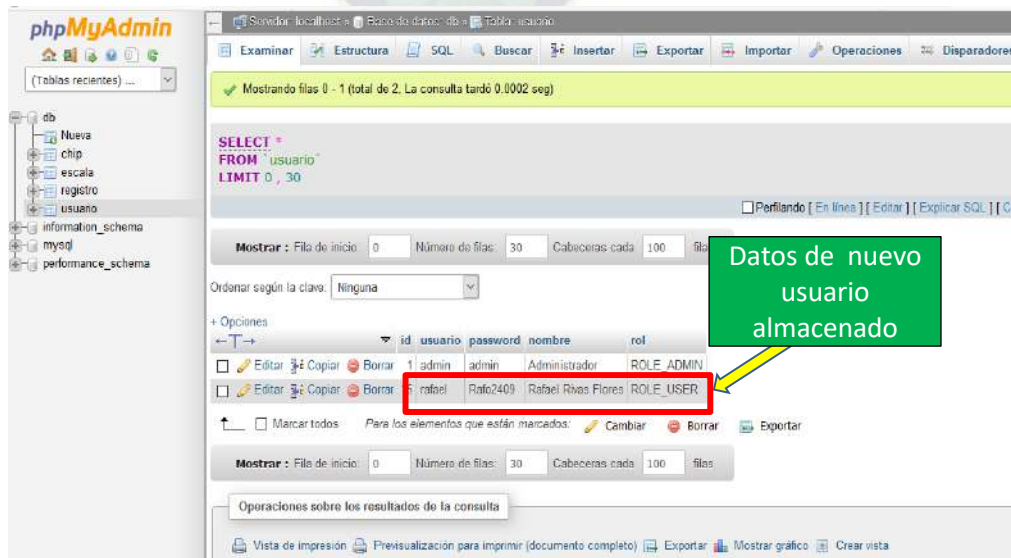


Figura N°51. Datos nuevo usuario almacenado
Fuente: Elaboración propia

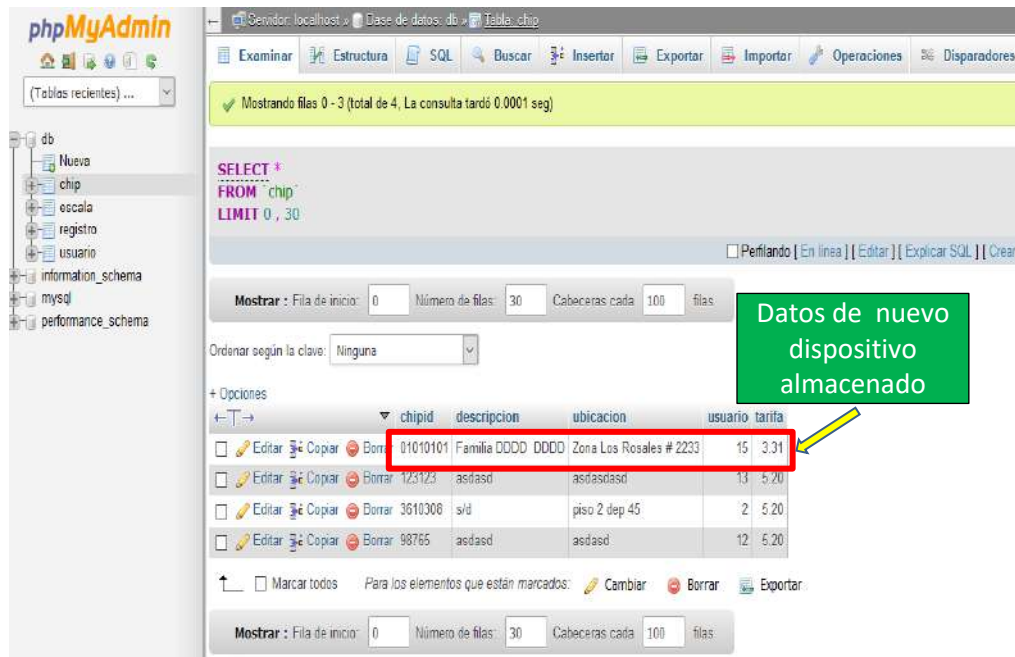


Figura N°52. Datos nuevo dispositivo almacenado
Fuente: Elaboración propia

3.5.4 REPORTE DEL CALCULO DE CONSUMO DE AGUA Y EL TOTAL A PAGAR

En esta pestaña es donde procederemos a realizar el cometido de este proyecto. Es donde se procederá con los cálculos respectivos para determinar el precio a pagar por el consumo que cada usuario haya realizado, los datos a ingresar para poder obtener el reporte son los siguientes:

- Fechas: calcula el consumo entre el rango de fechas ingresadas.
- Total consumo factura: debemos ingresar el total en metros cúbicos M3 consumido.

Son los únicos datos que el usuario debe ingresar para obtener un reporte donde le indique el consumo y el monto a pagar en bolivianos Bs.

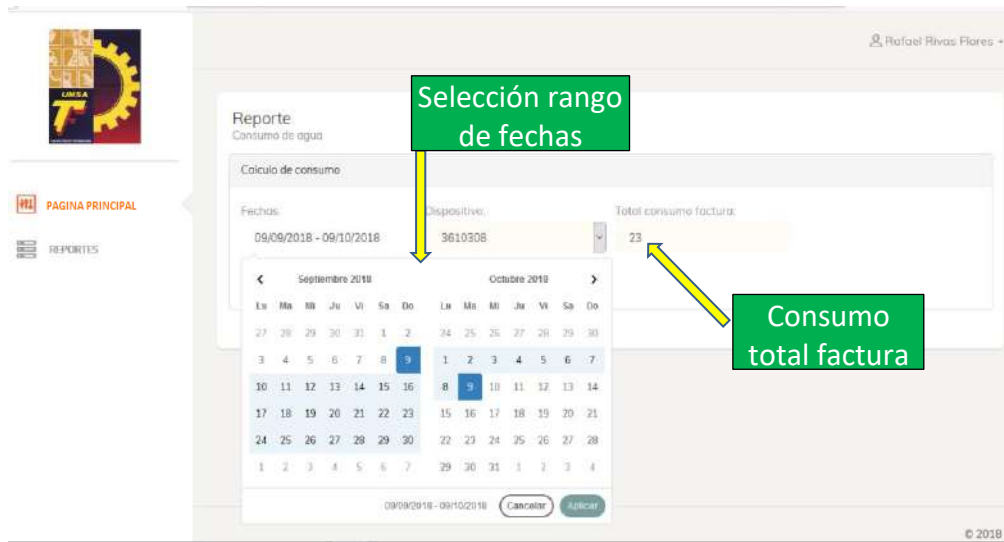


Figura N°53. Reporte consumo de agua
Fuente: Elaboración propia

Una vez ingresados los datos solo es necesario presionar el botón CALCULAR para observar el reporte final, como se muestra en la Figura N°54.

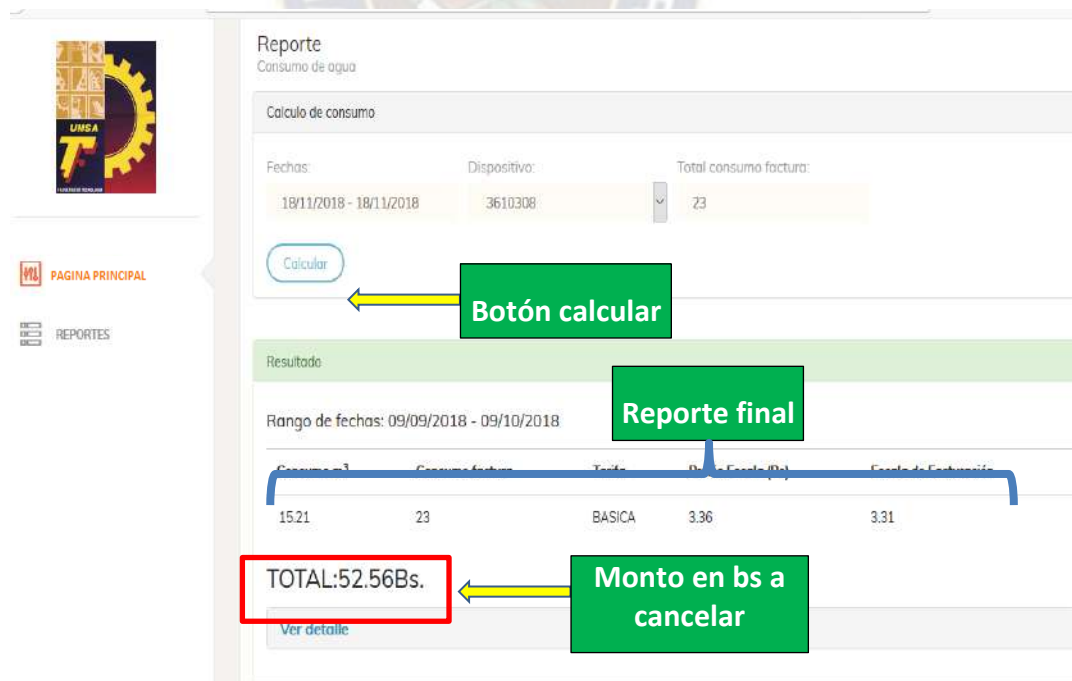


Figura N°53. Reporte consumo de agua
Fuente: Elaboración propia

3.5.5 CONSIDERACIONES PARA EL CÁLCULO DEL MONTO A PAGAR SEGÚN EL CONSUMO REALIZADO

Al momento de realizar las consultas en PHP para calcular el monto a cancelar en Bs. Se tomó en cuenta las escalas de cobro que tienen en consideración la empresa encargada de suministrar el agua potable en el departamento de La Paz la cual es Empresa Pública Social de Agua y Saneamiento **EPSAS**, dicha escala es la siguiente:

<i>Tarifa</i>	<i>Consumo(m3)</i>	<i>precio x m3(Bs)</i>
SOLIDARIA	0 - 15	1.78
BASICA	16 - 30	3.40
MEDIA	31 - 150	6.26
ALTA	151 - 300	10.10
E	301 - en adelante	18.05

Tabla N°4. Escala de precios según consumo
Fuente: EPSAS

Adicionalmente a esta escala se tiene un cobro de facturación que varía según a la ciudad en la que el usuario se encuentre, ya sea en la ciudad de La Paz o la ciudad de El Alto.

A cada escala de facturación se suma el monto por facturación, la cual es:

- **3,31 Bs** por facturación (**El Alto**)
- **5,20 Bs** por facturación (**La Paz**)

A continuación se realizaran tres ejemplos sobre el cálculo de consumo.

Suponiendo que el usuario consumió 13 M3 el cálculo sobre el monto a cancelar es el siguiente.

13 M3 se ubica en la primera escala que es la Solidaria, por cada M3 se multiplicará 1,78 Bs. Adicionalmente se sumara el monto por facturación 5.20 Bs, si tomamos en cuenta que el usuario está ubicado en la ciudad de La Paz

$$13 * 1.78 = 23.14 \quad \Rightarrow \quad 23.14 + 5.20 = 28.34$$

28.34 Bs será el monto a cancelar por el consumo de **13 M3**

Si el usuario consumió 25 M3 el cálculo será el siguiente.

$$25 * 3.40 = 85 \quad \Rightarrow \quad 85 + 5.20 = 90.20$$

90.20 Bs será el monto a cancelar por el consumo de **25 M3**

Ahora si el usuario realizo un consumo de 50 M3 que está por encima de los 30 M3 los cálculos serán de la siguiente forma.

De los 50 M3 consumidos los primeros 30 M3 se multiplicaran con el precio de la tarifa básica, los restantes 20 M3 se harán con la respectiva escala a la cual pertenece 50 M3 la cual es 6.26 Bs por metro cubico.

50 M3	{	30 M3	=>	30 * 3.40	=	102 Bs
		20 M3	=>	20 * 6.26	=	125.2 Bs
total		50 M3				227.2 Bs

$$227.2 \text{ Bs} + 5.20 \text{ Bs} = 232.40 \text{ Bs}$$

232.40 Bs será el monto a cancelar por el consumo de **50 M3**

De esa manera es como se realizaran los cálculos si el consumo sobre pasa los 30 M3

3.6 INSTALACIÓN DEL SERVIDOR WEB EN DIGITALOCEAN

Después de haber verificado que nuestro prototipo electrónico de medición y consumo de agua y la interfaz web para el usuario funcionan de acuerdo a lo esperado, es necesario que nuestra interfaz web y la base de datos estén alojadas en un servidor web para que el prototipo envíe los datos de consumo a través de una conexión Wi-Fi y la interfaz web pueda ser consultada desde un dispositivo móvil o computador desde cualquier punto remoto con conexión a internet.

Para ello seleccionamos los servicios de DigitalOcean que nos proveen de máquinas virtuales para la instalación de nuestro propio servidor, dichas máquinas virtuales son denominadas Droplets (gotitas).

3.6.1 CREACIÓN DEL DROPELT

Lo primero que se hizo fue crear una cuenta en **Digital Ocean** e ingresar a www.digitalocean.com con el nombre de usuario y contraseña que escogimos.

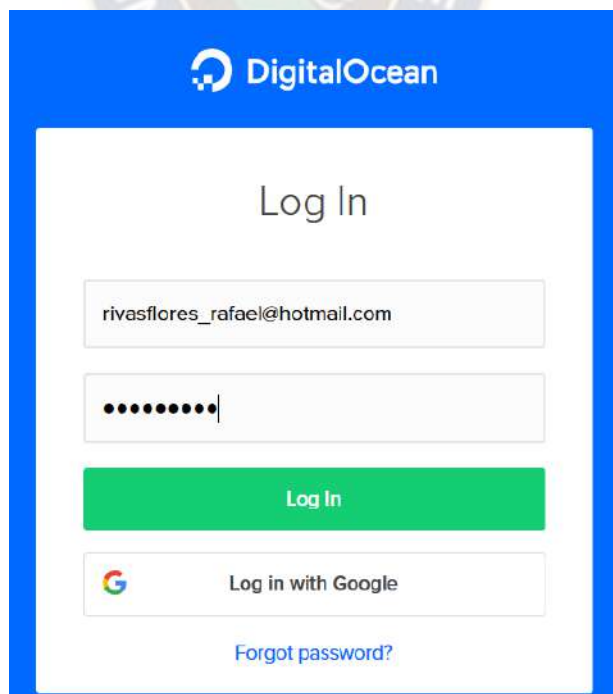


Figura N°55. Inicio de cesión en DigitalOcean
Fuente: Elaboración propia

El siguiente paso es seleccionar la opción crear Droplet como se observa en la Figura N°55

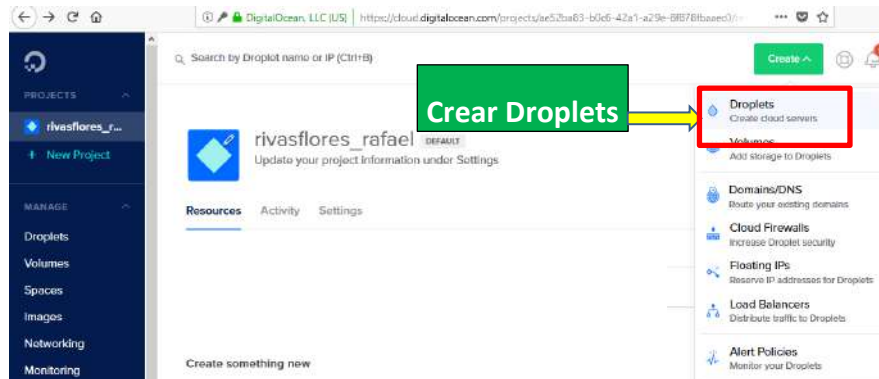


Figura N°56. Creación nuevo Droplet
Fuente: Elaboración propia

Ahora vamos a elegir la imagen de la distribución de Linux a instalar, elegimos la primera opción que es Ubuntu 16.04 de 64 bits, adicionalmente debemos elegir el plan que se requiera, para el proyecto se eligió el de \$ 5,00 dólares mensuales con las siguientes características:

- 1 GB de RAM
- 25 GB de disco duro tipo SSD
- 1 TB de transferencia

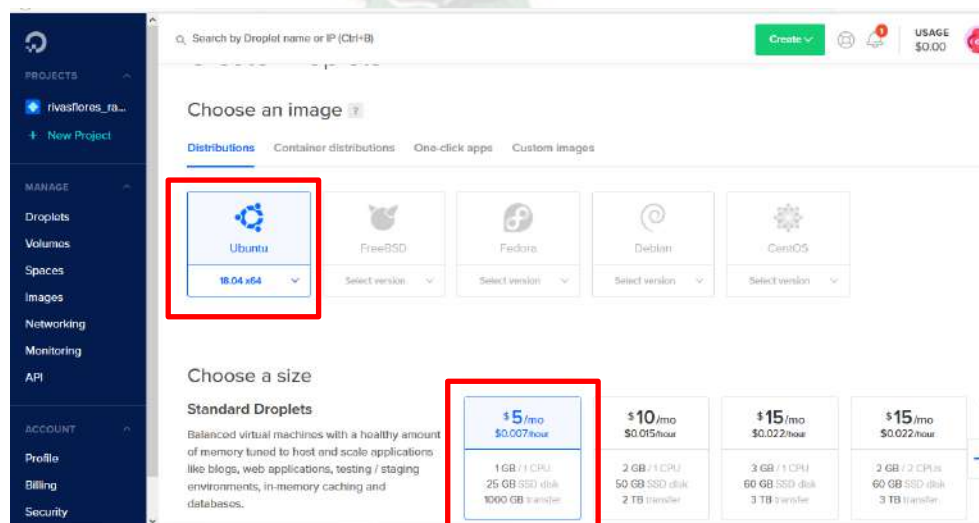


Figura N°57. Características nuevo Droplet
Fuente: Elaboración propia

En el siguiente paso seleccionamos el lugar donde estará ubicado el centro de datos donde va estar nuestro servidor, por lo general aparece por defecto el que está más cercano de acuerdo a nuestra ubicación actual. En nuestro caso Nueva York.

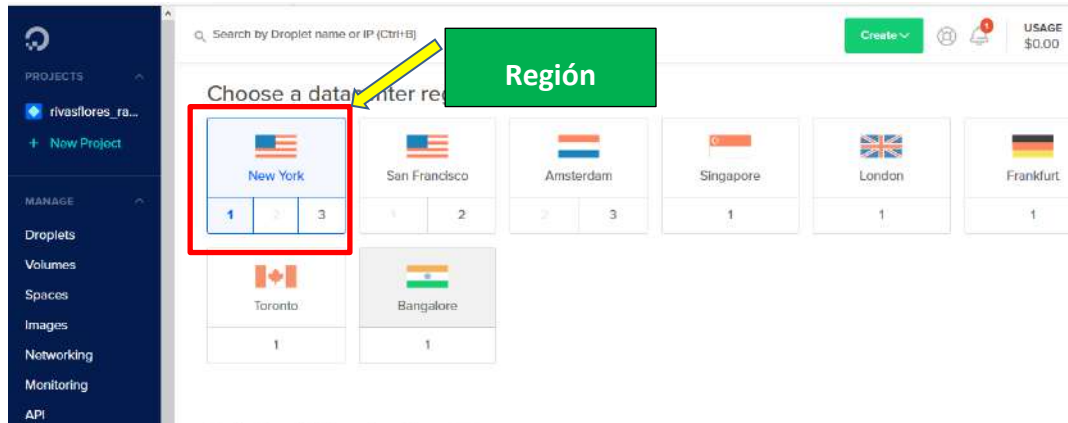


Figura N°58. Selección de la región
Fuente: Elaboración propia

Finalmente seleccionamos el nombre que le daremos nuestro nuevo Droplet y presionamos crear.

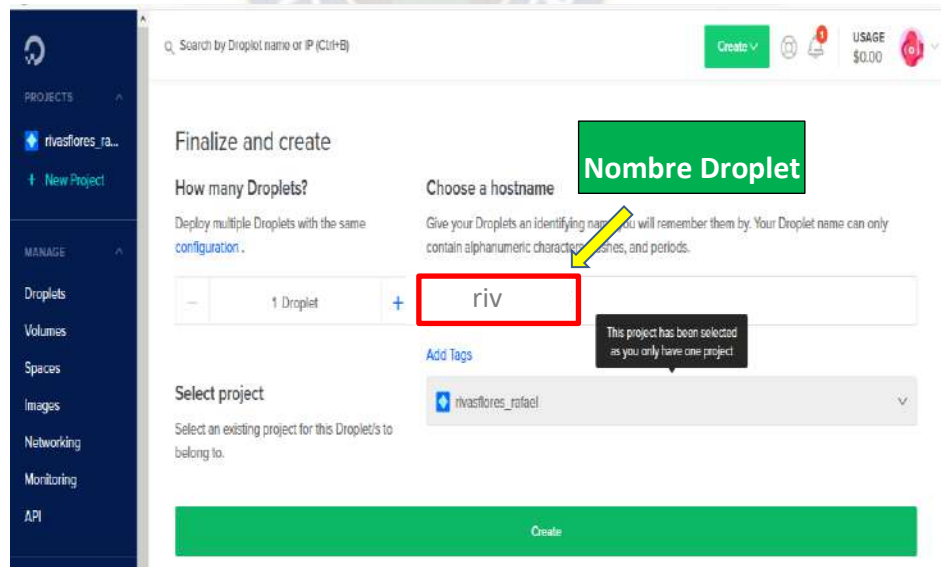


Figura N°59. Nombre de Droplet
Fuente: Elaboración propia

Una vez terminado el proceso, en el menú **Droplets** nos aparecerá lo que acabamos de crear, junto a la IP asignada para su acceso, la cual es 104.248.114.61.



Figura N°60. IP asignada de nuevo Droplet
Fuente: Elaboración propia

En Figura N°61 se observa cómo podemos encender y apagar nuestra máquina virtual Droplet.



Figura N°61. Encender y apagar el Droplet
Fuente: Elaboración propia

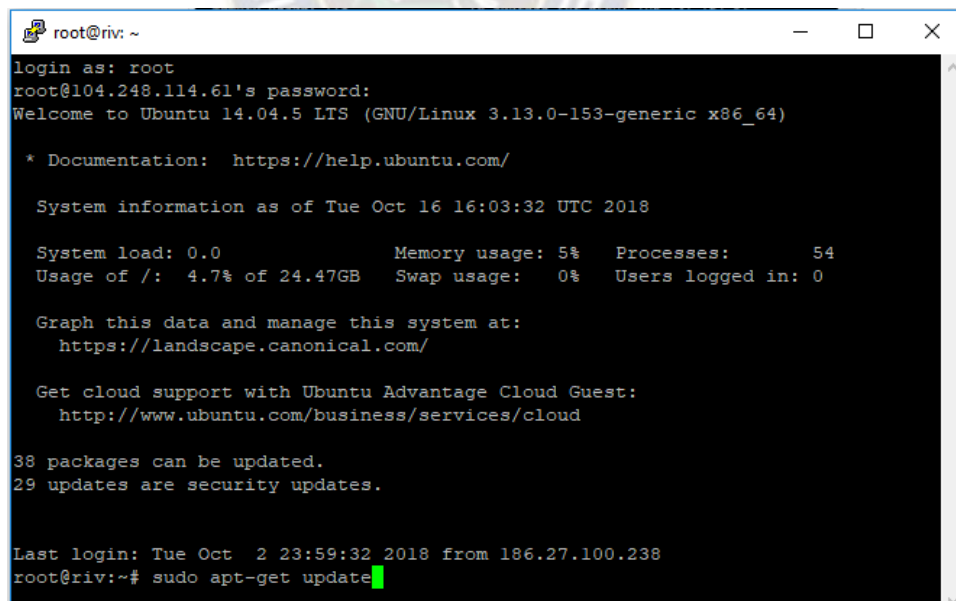
3.6.2 INSTALACIÓN DE LA PILA LAMP EN EL DROPLET

Para subir nuestros archivos a nuestro servidor web es necesario instalar LAMP, este término en realidad es un acrónimo que representa un sistema operativo **L**inux con un servidor **A**pache, los datos del sitio son almacenados en base de datos **M**ySQL y el contenido dinámico es procesado con **P**HP.

Por lo que procederemos a instalar Apache, MySQL y PHP en la máquina virtual creada en DigitalOcean denominada Droplet, todas las aplicaciones tienen que ser instaladas desde el SSH (Secure Shell), que es un protocolo de administración remota que permite a los usuarios controlar y modificar sus servidores remotos a través de Internet, por lo que el primer paso será actualizar los repositorios que son todos los servidores desde donde se descargarán las aplicaciones a instalar.

Ejecutamos el siguiente comando en la SSH del Droplet creado:

```
sudo apt-get update
```



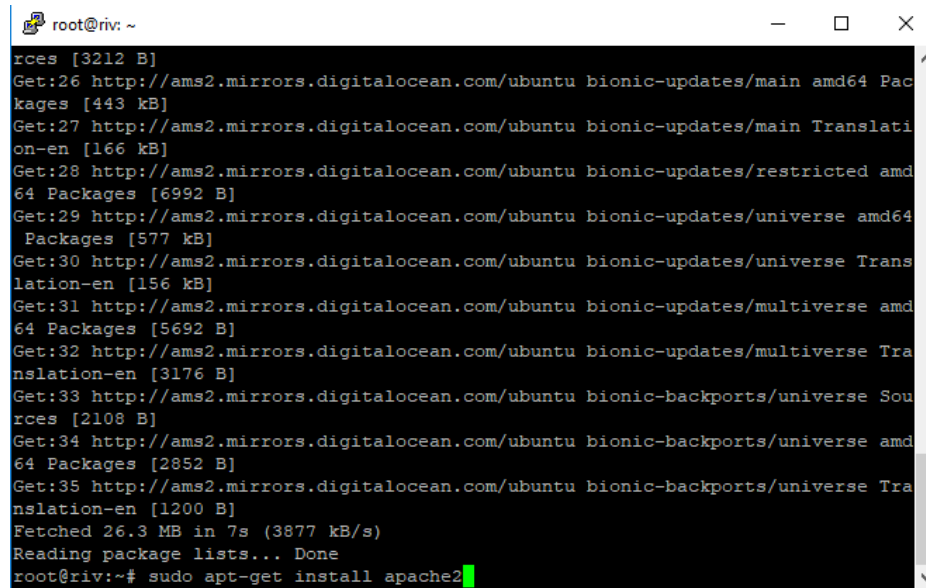
```
root@riv: ~  
login as: root  
root@104.248.114.61's password:  
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.13.0-153-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/  
  
System information as of Tue Oct 16 16:03:32 UTC 2018  
  
System load: 0.0          Memory usage: 5%    Processes:      54  
Usage of /:  4.7% of 24.47GB  Swap usage:  0%    Users logged in: 0  
  
Graph this data and manage this system at:  
  https://landscape.canonical.com/  
  
Get cloud support with Ubuntu Advantage Cloud Guest:  
  http://www.ubuntu.com/business/services/cloud  
  
38 packages can be updated.  
29 updates are security updates.  
  
Last login: Tue Oct  2 23:59:32 2018 from 186.27.100.238  
root@riv:~# sudo apt-get update
```

Figura N°62. Actualización de repositorio
Fuente: Elaboración propia

3.6.2.1 INSTALACIÓN DE APACHE

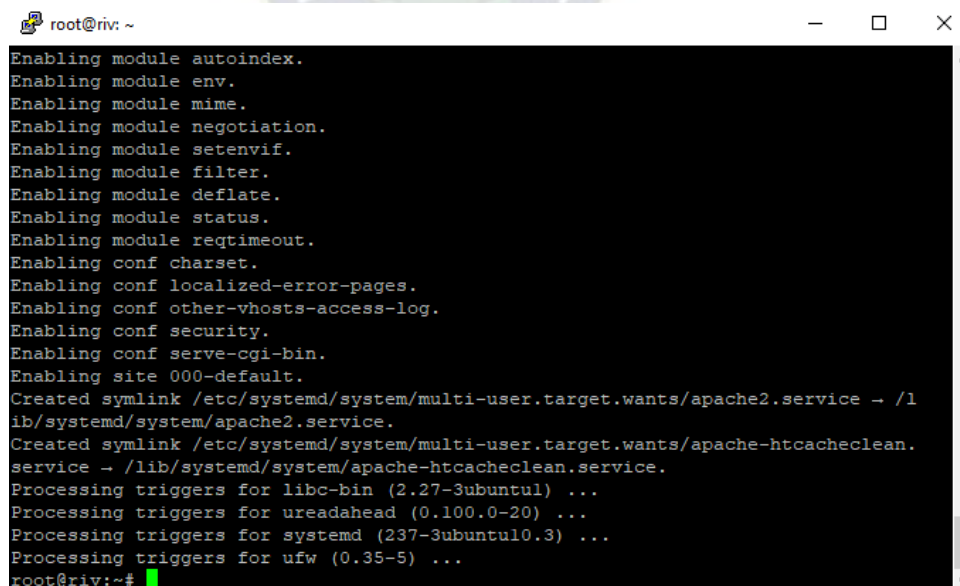
Ejecutamos el siguiente comando para la instalación de Apache en el Droplet.

```
sudo apt install apache2
```



```
root@riv: ~  
rces [3212 B]  
Get:26 http://ams2.mirrors.digitalocean.com/ubuntu bionic-updates/main amd64 Pac  
kages [443 kB]  
Get:27 http://ams2.mirrors.digitalocean.com/ubuntu bionic-updates/main Translati  
on-en [166 kB]  
Get:28 http://ams2.mirrors.digitalocean.com/ubuntu bionic-updates/restricted amd  
64 Packages [6992 B]  
Get:29 http://ams2.mirrors.digitalocean.com/ubuntu bionic-updates/universe amd64  
 Packages [577 kB]  
Get:30 http://ams2.mirrors.digitalocean.com/ubuntu bionic-updates/universe Trans  
lation-en [156 kB]  
Get:31 http://ams2.mirrors.digitalocean.com/ubuntu bionic-updates/multiverse amd  
64 Packages [5692 B]  
Get:32 http://ams2.mirrors.digitalocean.com/ubuntu bionic-updates/multiverse Tra  
nslation-en [3176 B]  
Get:33 http://ams2.mirrors.digitalocean.com/ubuntu bionic-backports/universe Sou  
rces [2108 B]  
Get:34 http://ams2.mirrors.digitalocean.com/ubuntu bionic-backports/universe amd  
64 Packages [2852 B]  
Get:35 http://ams2.mirrors.digitalocean.com/ubuntu bionic-backports/universe Tra  
nslation-en [1200 B]  
Fetched 26.3 MB in 7s (3877 kB/s)  
Reading package lists... Done  
root@riv:~# sudo apt-get install apache2
```


Figura N°63. Comando instalación de Apache
Fuente: Elaboración propia



```
root@riv: ~  
Enabling module autoindex.  
Enabling module env.  
Enabling module mime.  
Enabling module negotiation.  
Enabling module setenvif.  
Enabling module filter.  
Enabling module deflate.  
Enabling module status.  
Enabling module reqtimeout.  
Enabling conf charset.  
Enabling conf localized-error-pages.  
Enabling conf other-vhosts-access-log.  
Enabling conf security.  
Enabling conf serve-cgi-bin.  
Enabling site 000-default.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service -> /l  
ib/systemd/system/apache2.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.  
service -> /lib/systemd/system/apache-htcacheclean.service.  
Processing triggers for libc-bin (2.27-3ubuntu1) ...  
Processing triggers for ureadahead (0.100.0-20) ...  
Processing triggers for systemd (237-3ubuntu10.3) ...  
Processing triggers for ufw (0.35-5) ...  
root@riv:~#
```

Figura N°64. Apache instalado
Fuente: Elaboración propia

Para una comprobación de que Apache ha sido instalado correctamente visitamos la dirección IP pública de nuestro servidor en un navegador y veremos la página web predeterminada de Apache para Ubuntu, la cual tiene propósitos informativos y de prueba.

 **Apache2 Ubuntu Default Page**

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

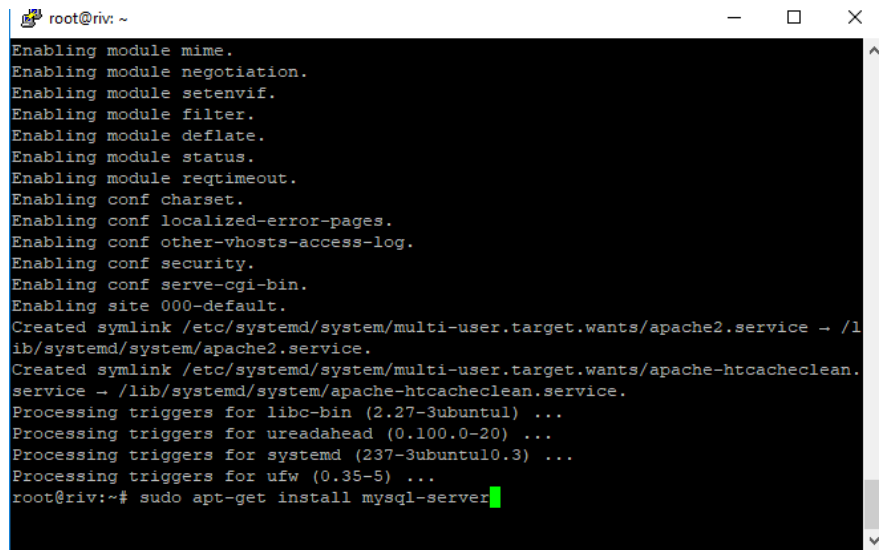
Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

Figura N°65. Página web de apache de instalación satisfactoria
Fuente: Elaboración propia

3.6.2.2 INSTALCION MySQL

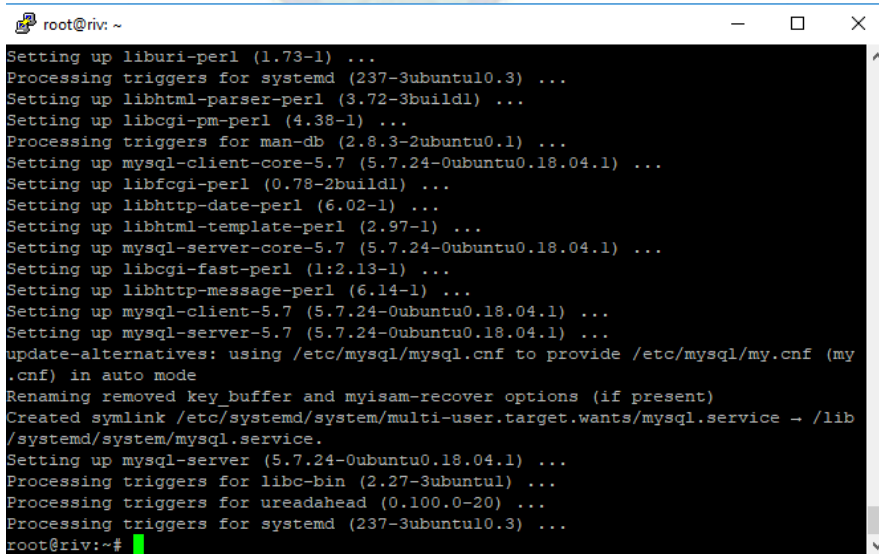
Ejecutamos el siguiente comando para la instalación de MySQL en el Droplet.

```
sudo apt install mysql-server
```



```
root@riv: ~
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service -> /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service -> /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for ureadahead (0.100.0-20) ...
Processing triggers for systemd (237-3ubuntu10.3) ...
Processing triggers for ufw (0.35-5) ...
root@riv:~# sudo apt-get install mysql-server
```

Figura N°66. Comando instalación de Apache
Fuente: Elaboración propia



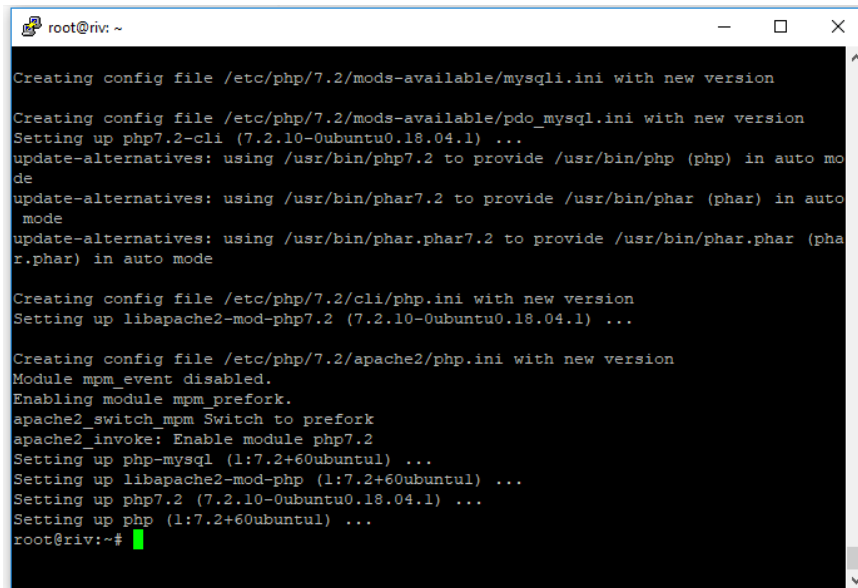
```
root@riv: ~
Setting up liburi-perl (1.73-1) ...
Processing triggers for systemd (237-3ubuntu10.3) ...
Setting up libhtml-parser-perl (3.72-3build1) ...
Setting up libcgi-pm-perl (4.38-1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Setting up mysql-client-core-5.7 (5.7.24-0ubuntu0.18.04.1) ...
Setting up libfcgi-perl (0.78-2build1) ...
Setting up libhttp-date-perl (6.02-1) ...
Setting up libhtml-template-perl (2.97-1) ...
Setting up mysql-server-core-5.7 (5.7.24-0ubuntu0.18.04.1) ...
Setting up libcgi-fast-perl (1:2.13-1) ...
Setting up libhttp-message-perl (6.14-1) ...
Setting up mysql-client-5.7 (5.7.24-0ubuntu0.18.04.1) ...
Setting up mysql-server-5.7 (5.7.24-0ubuntu0.18.04.1) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Renaming removed key_buffer and myisam-recover options (if present)
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service -> /lib/systemd/system/mysql.service.
Setting up mysql-server (5.7.24-0ubuntu0.18.04.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for ureadahead (0.100.0-20) ...
Processing triggers for systemd (237-3ubuntu10.3) ...
root@riv:~#
```

Figura N°67. MySQL instalado
Fuente: Elaboración propia

3.6.2.3 INSTALACIÓN PHP

Ejecutamos el siguiente comando para la instalación de PHP en el Droplet y se pueda ejecutar sobre el servidor apache y también para que se comunique con la base de datos MySQL

```
sudo apt install php libapache2-mod-php php-mysql
```



```
root@riv: ~  
Creating config file /etc/php/7.2/mods-available/mysqli.ini with new version  
Creating config file /etc/php/7.2/mods-available/pdo_mysql.ini with new version  
Setting up php7.2-cli (7.2.10-0ubuntu0.18.04.1) ...  
update-alternatives: using /usr/bin/php7.2 to provide /usr/bin/php (php) in auto mode  
update-alternatives: using /usr/bin/phar7.2 to provide /usr/bin/phar (phar) in auto mode  
update-alternatives: using /usr/bin/phar.phar7.2 to provide /usr/bin/phar.phar (phar.phar) in auto mode  
Creating config file /etc/php/7.2/cli/php.ini with new version  
Setting up libapache2-mod-php7.2 (7.2.10-0ubuntu0.18.04.1) ...  
Creating config file /etc/php/7.2/apache2/php.ini with new version  
Module mpm_event disabled.  
Enabling module mpm_prefork.  
apache2_switch_mpm Switch to prefork  
apache2_invoke: Enable module php7.2  
Setting up php-mysql (1:7.2+60ubuntu1) ...  
Setting up libapache2-mod-php (1:7.2+60ubuntu1) ...  
Setting up php7.2 (7.2.10-0ubuntu0.18.04.1) ...  
Setting up php (1:7.2+60ubuntu1) ...  
root@riv:~#
```

Figura N°68. PHP instalado
Fuente: Elaboración propia

3.6.2.4 INSTALACION DE PHPMYADMIN

Ejecutamos el siguiente comando para la instalación de phpMyAdmin en el Droplet.

```
sudo apt-get install phpmyadmin php-mbstring php-gettext
```

Luego de ejecutar el comando se harán algunas consultas y peticiones.

- Selección del servidor, elegimos **apache2**.

```
root@riv: ~  
Creating config file /etc/php/7.2/mods-available/mysql.ini with new version  
Creating config file /etc/php/7.2/mods-available/pdo_mysql.ini with new version  
Setting up php7.2-cli (7.2.10-0ubuntu0.18.04.1) ...  
update-alternatives: using /usr/bin/php7.2 to provide /usr/bin/php (php) in auto mode  
update-alternatives: using /usr/bin/phar7.2 to provide /usr/bin/phar (phar) in auto mode  
update-alternatives: using /usr/bin/phar.phar7.2 to provide /usr/bin/phar.phar (phar.phar) in auto mode  
Creating config file /etc/php/7.2/cli/php.ini with new version  
Setting up libapache2-mod-php7.2 (7.2.10-0ubuntu0.18.04.1) ...  
Creating config file /etc/php/7.2/apache2/php.ini with new version  
Module mpm_event disabled.  
Enabling module mpm_prefork.  
apache2_switch_mpm Switch to prefork  
apache2_invoke: Enable module php7.2  
Setting up php-mysql (1:7.2+60ubuntu1) ...  
Setting up libapache2-mod-php (1:7.2+60ubuntu1) ...  
Setting up php7.2 (7.2.10-0ubuntu0.18.04.1) ...  
Setting up php (1:7.2+60ubuntu1) ...  
root@riv:~# sudo apt-get install phpmyadmin php-mbstring php-gettext
```

Figura N°69. Comando instalación de phpMyAdmin
Fuente: Elaboración propia



Figura N°70. Selección de servidor apache2
Fuente: Elaboración propia

Ingresamos la contraseña para el administrador de la base de datos

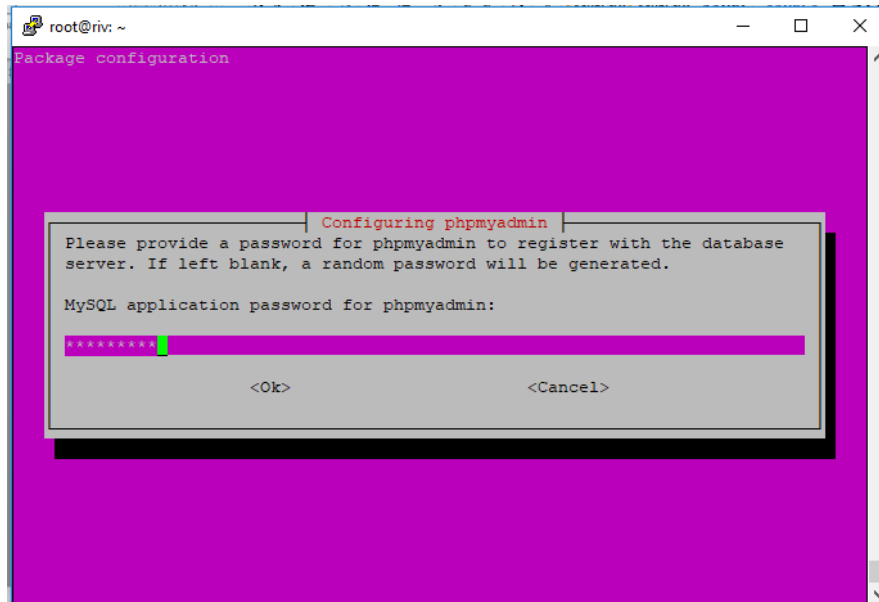


Figura N°71. Selección de servidor apache2
Fuente: Elaboración propia

Confirmación de la contraseña

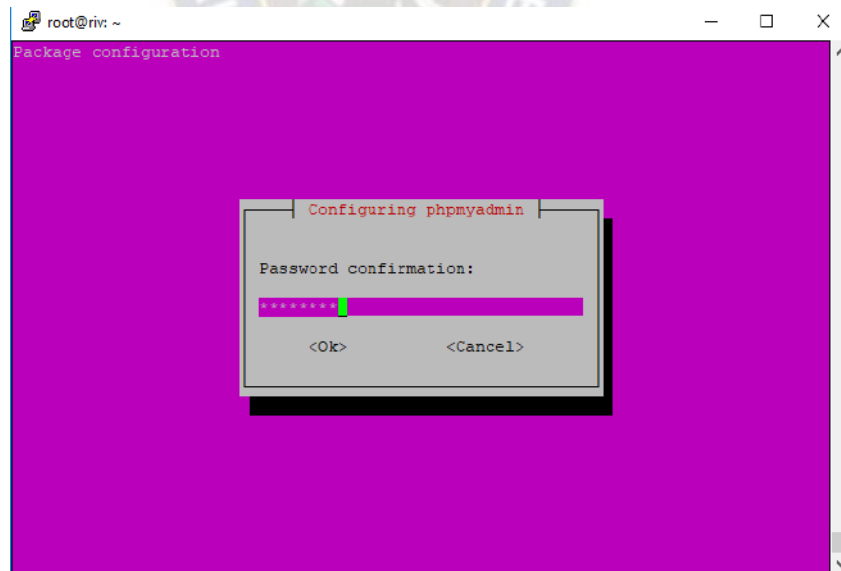


Figura N°72. Confirmación contraseña
Fuente: Elaboración propia

Ahora podemos acceder a la interfaz web de phpMyAdmin visitando la dirección IP pública seguida de /phpmyadmin:

http://104.248.114.61/phpmyadmin

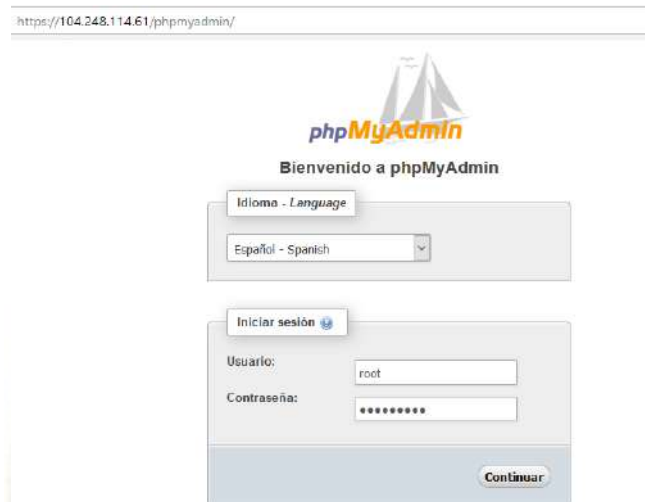


Figura N°73. Acceso a phpMyAdmin
Fuente: Elaboración propia

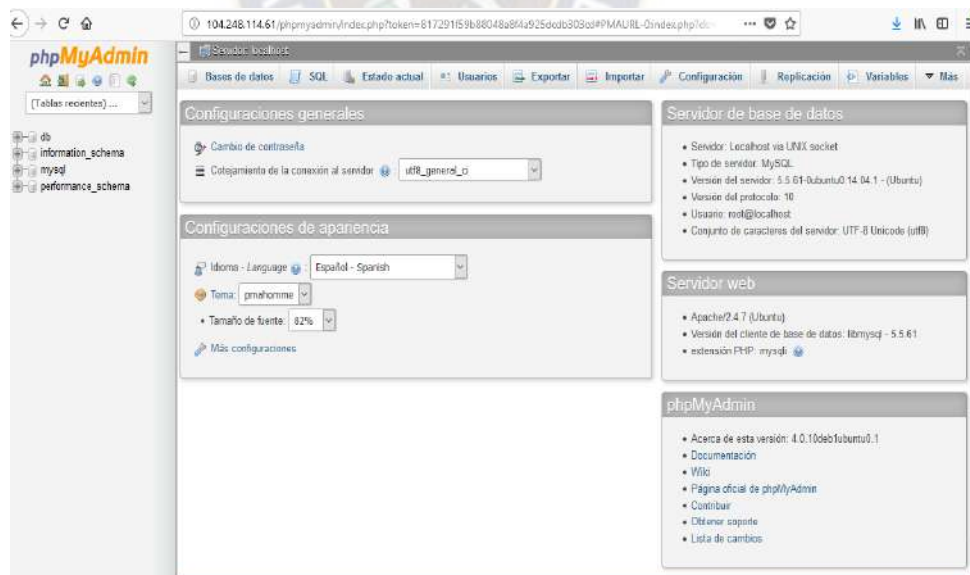


Figura N°74. Interfaz web phpMyAdmin
Fuente: Elaboración propia

3.6.3 SUBIENDO LOS ARCHIVOS AL SERVIDOR

Una vez instalado todo lo necesario en la máquina virtual (Droplet) para que nuestro servidor web funcione de manera correcta, ahora es necesario subir nuestros archivos a dicho servidor para que se pueda acceder vía internet de manera remota desde cualquier punto con conexión a internet con la IP pública asignada.

Para ello usamos una aplicación llamada WinSCP que no permitirá llevar todos los archivos necesarios a la dirección raíz de ubuntu que es `/var/www/html/`

Iniciamos la aplicación WinSCP e incluimos la ip asignada.

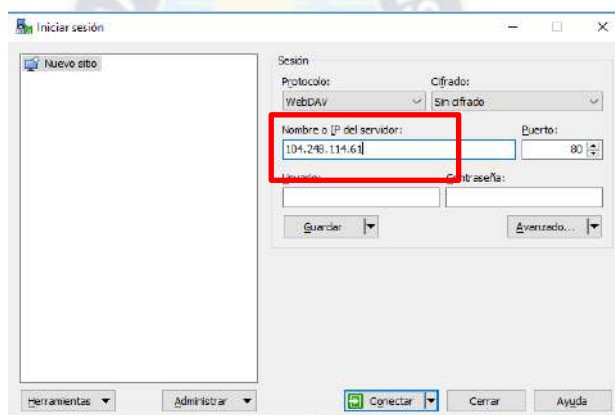


Figura N°75. Inicio de WinSCP con la IP asignada
Fuente: Elaboración propia

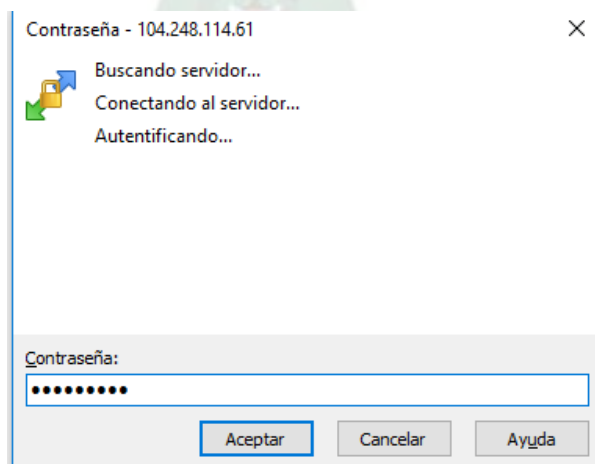


Figura N°76. Conexión al servidor
Fuente: Elaboración propia

Ingreamos al panel de WinSCP, pasamos los archivos necesarios de la columna izquierda hacia la columna derecha donde está el directorio raíz de ubuntu.

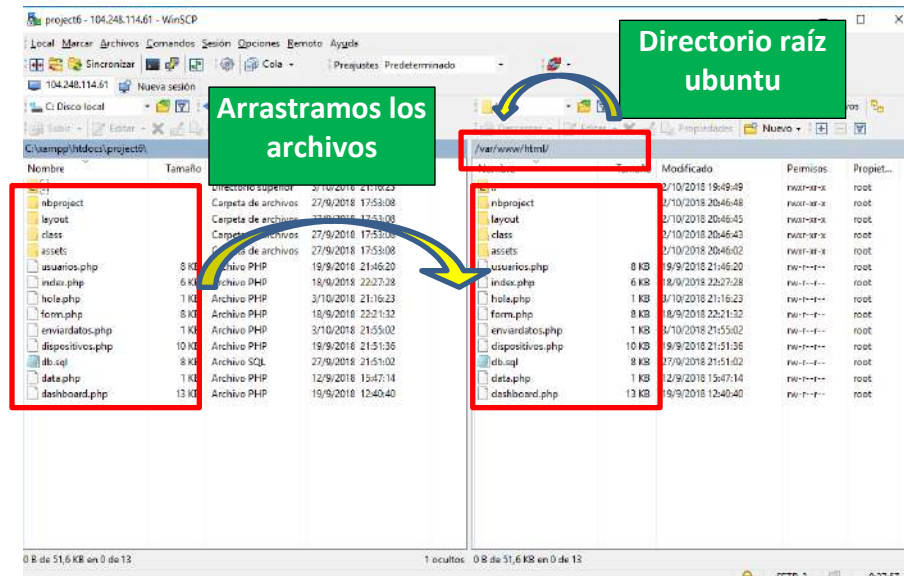


Figura N°77. Archivos cargados en el servidor
Fuente: Elaboración propia

A continuación cargamos nuestra base de datos y sus correspondientes tablas en phpMyAdmin instalada en nuestra maquina virtual (Droplet).

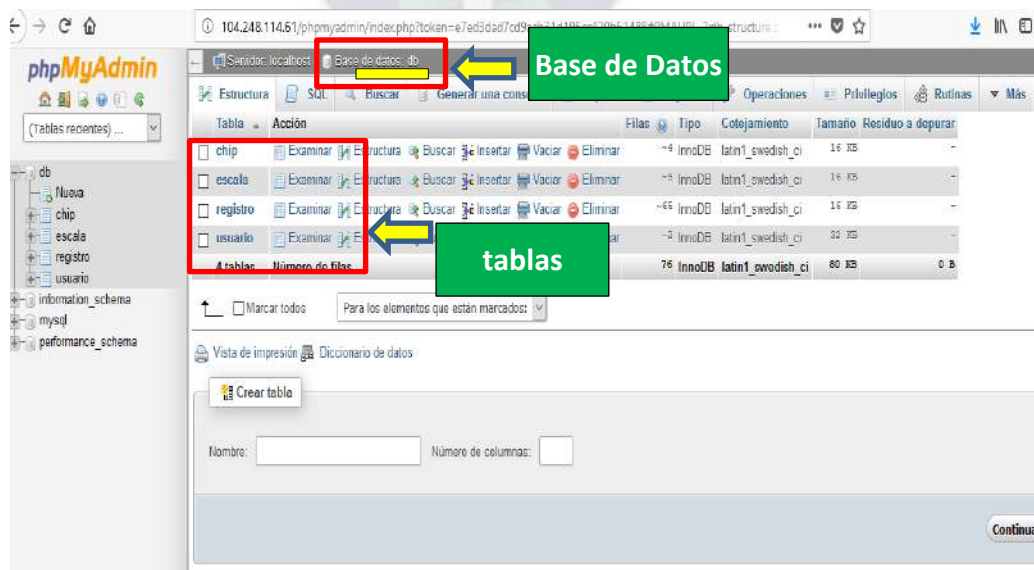


Figura N°78. Base de datos cargada en phpMyAdmin
Fuente: Elaboración propia

3.7 PRUEBA DE FUNCIONAMIENTO HARDWARE Y SOFTWARE

3.7.1 DIAGRAMA DE CONEXIÓN HARDWARE

En la Figura N°79 se observa el diagrama de conexión.

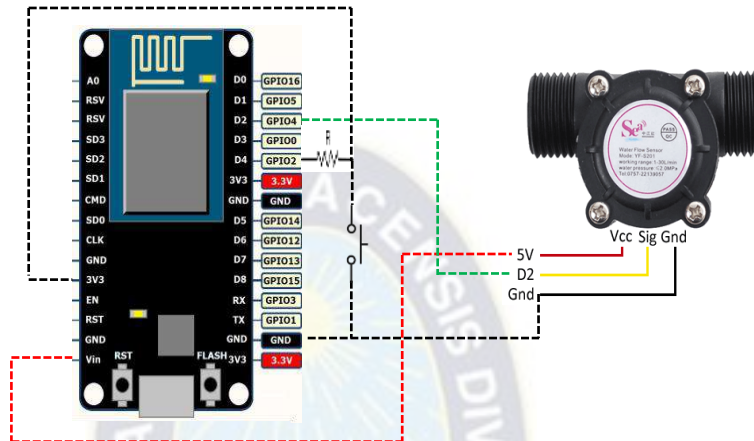


Figura N°79. Diagrama de conexión del circuito
Fuente: Elaboración propia

3.7.1.1 IMPLEMENTACIÓN DE CIRCUITO EN PLACA PCB

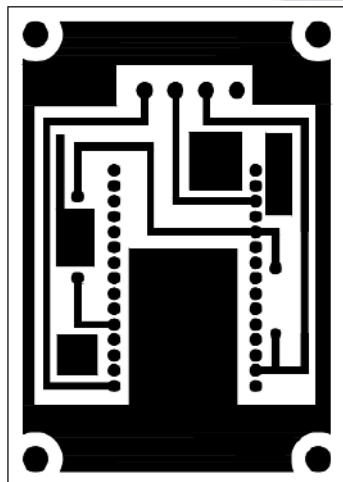


Figura N°80. Diseño para placa PCB
Fuente: Elaboración propia



Figura N°81. Circuito impreso PCB
Fuente: Elaboración propia



Figura N°82. Circuito implementado en PCB
cara inferior
Fuente: Elaboración propia

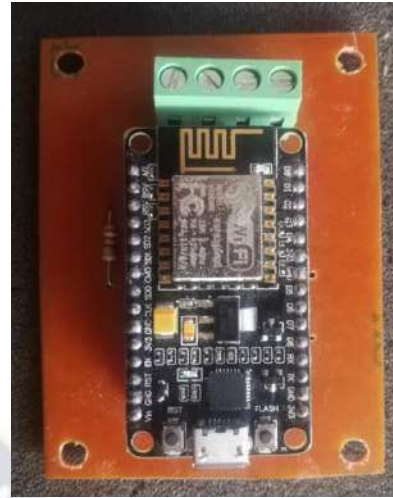


Figura N°83. Circuito implementado en PCB
cara superior
Fuente: Elaboración propia



Figura N°84. Diseño final del circuito en PCB
Fuente: Elaboración propia



Figura N°85. Diseño final del circuito en PCB
Fuente: Elaboración propia

3.7.2 INSTALACIÓN DEL HARDWARE

La primera instalación del prototipo se la realizó en tuberías de 1/2" estándar de agua potable.



Figura N°86. Caudalímetro de 1/2" instalado en tubería
Fuente: Elaboración propia

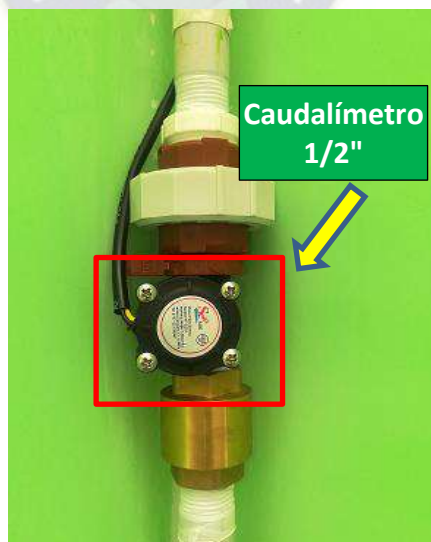


Figura N°87. Caudalímetro de 1/2" instalado en tubería
Fuente: Elaboración propia

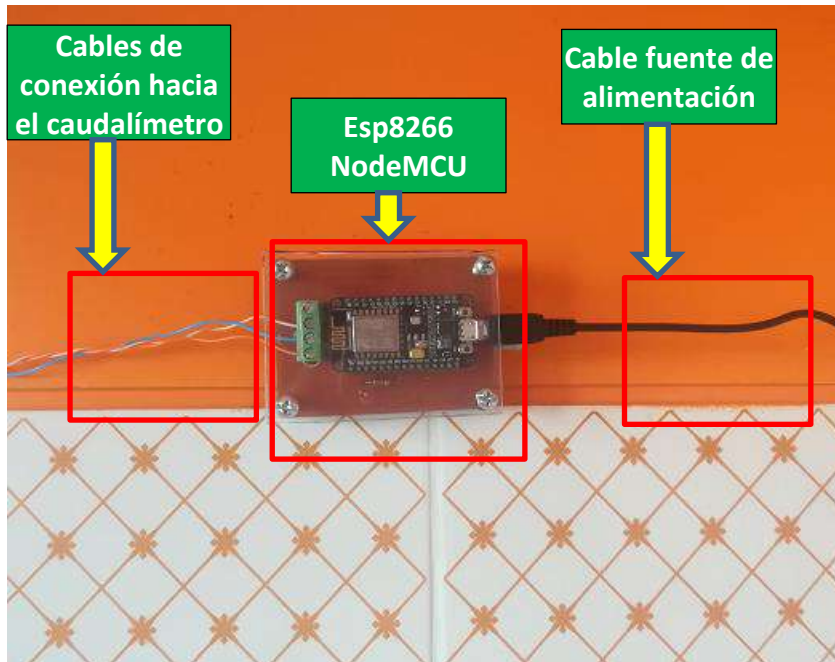


Figura N°88. Conexión tarjeta de desarrollo Esp8266NodeMCU
Fuente: Elaboración propia

La segunda instalación del prototipo se la realizó en tuberías de 3/4" estándar de agua potable.



Figura N°89. Caudalímetro de 3/4" instalado en tubería
Fuente: Elaboración propia

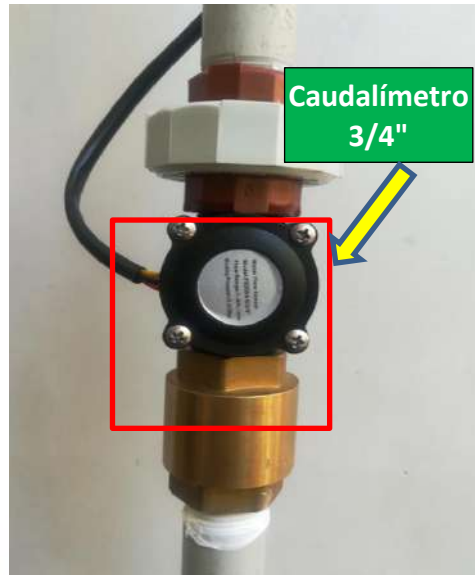


Figura N°90. Caudalímetro de 3/4" instalado en tubería
Fuente: Elaboración propia

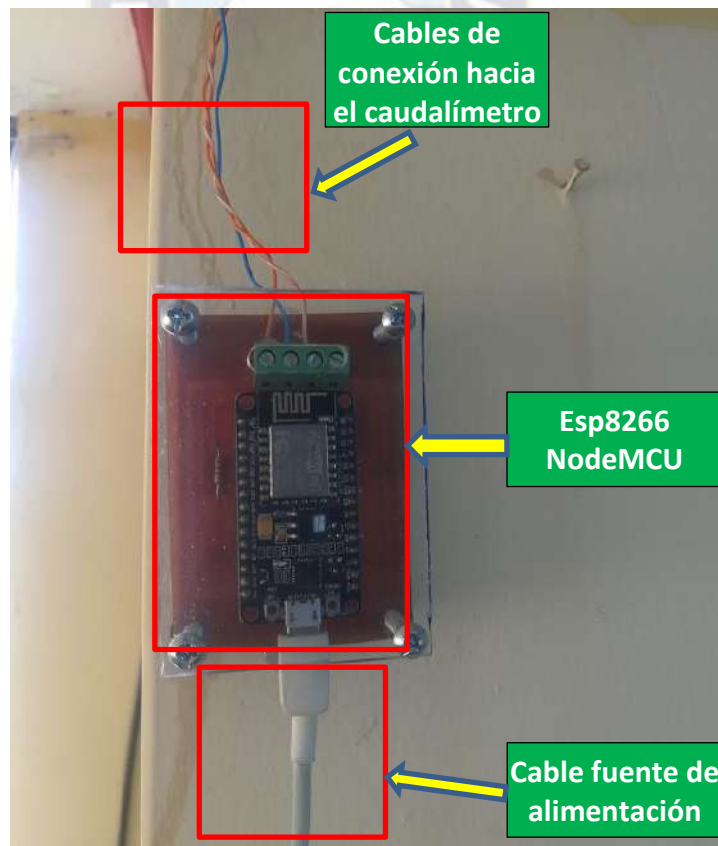


Figura N° 91. Conexión tarjeta de desarrollo Esp8266NodeMCU
Fuente: Elaboración propia

3.7.3 ACCESO Y PRUEBA DE LA INTERFAZ WEB DEL USUARIO

Como todos nuestros archivos ya están subidos en nuestro servidor, ya es posible acceder a la interfaz web desde cualquier dispositivo móvil o computador que tengan conexión a internet.

Para acceder a la interfaz web solo debemos escribir la Url que es la siguiente dirección IP: **http://104.248.114.61**. Y acceder a ella como usuario o administrador con el nombre y contraseña asignados.

Accedemos como administrador. Con el nombre de usuario XXXXX y la contraseña XXXXX

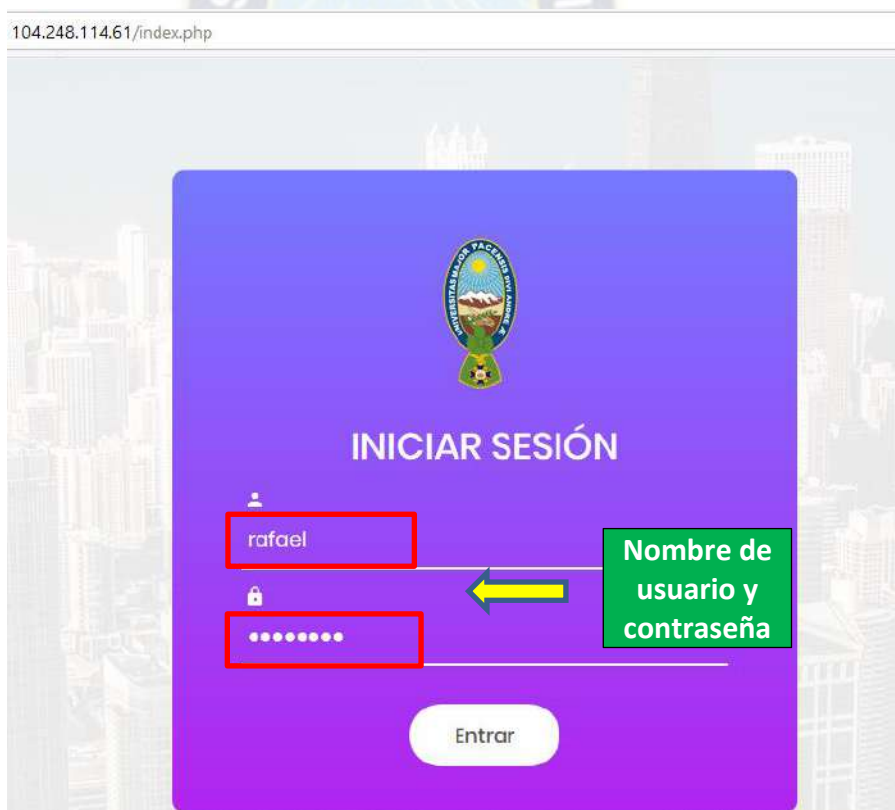


Figura N°92. Acceso a la interfaz web como usuario
Fuente: Elaboración propia

Como anteriormente agregamos dos nuevos usuarios y dos dispositivos, en la interfaz se puede apreciar ese dato. También se aprecia el menú principal.

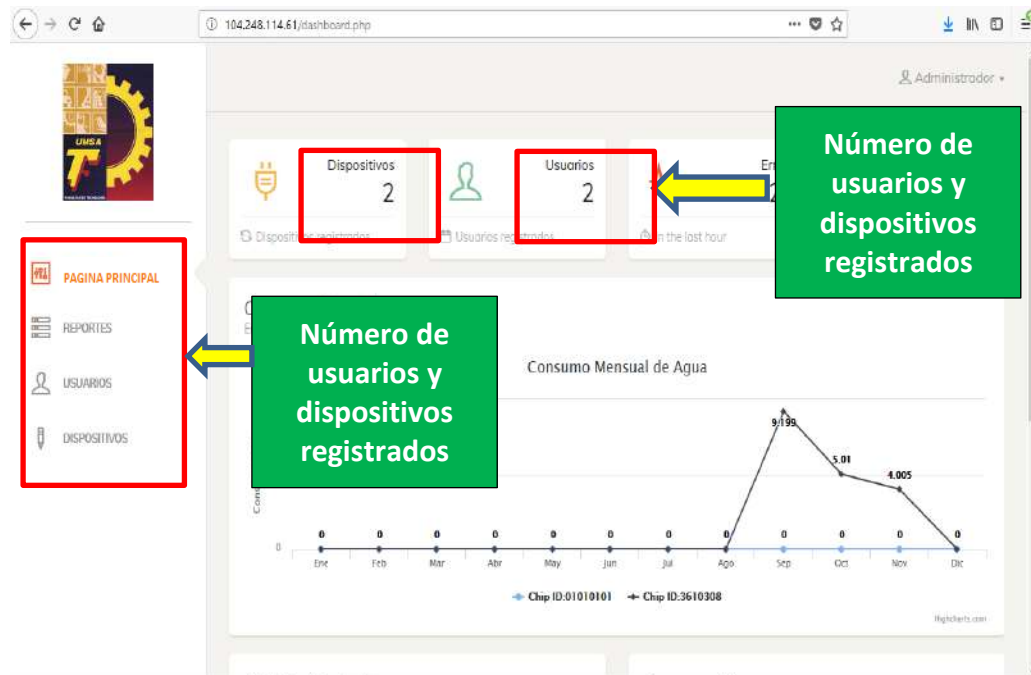


Figura N°93. Pantalla principal de la interfaz web
Fuente: Elaboración propia

Para realizar un reporte de consumo de manera correcta debemos seguir los siguientes pasos:

Paso 1.- Ingresamos a la opción reportes.

Paso 2.- seleccionamos el rango de fechas en que se realizara el cálculo de reporte de consumo, las cuales obtenemos de la factura de agua de la cual queremos saber nuestro consumo correspondiente.

Paso 3.- seleccionamos el dispositivo que es la id de la tarjeta de desarrollo.

Paso 4.- copiamos el total del consumo en M3(metros cúbicos) de la factura de agua de la cual queremos saber nuestro consumo correspondiente.

Paso5.- Presionamos calcular.

Realizaremos las pruebas de los dos prototipos de consumo, cabe resaltar que uno está instalado en una zona determinada de La Paz y el otro en El Alto.

El primer reporte es del que está instalado en la ciudad de El Alto.

Paso 1.- Ingresamos a la opción reportes.

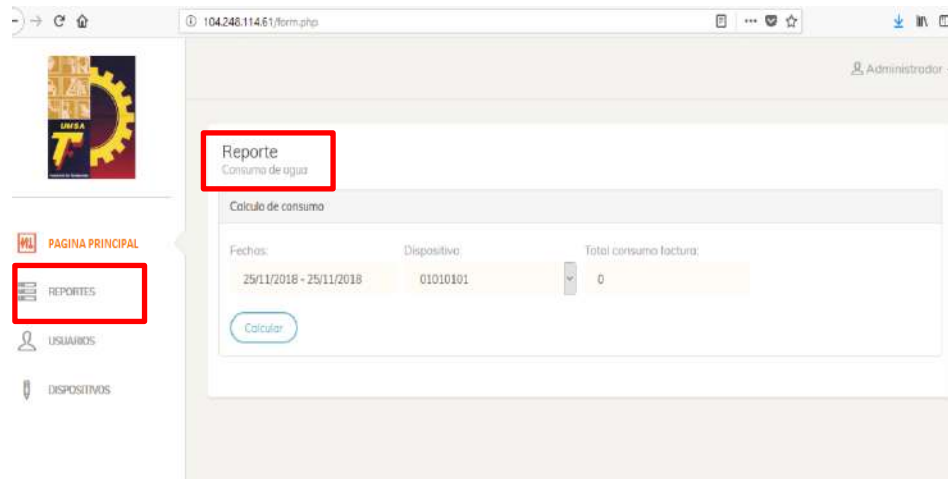


Figura N°94. Reporte de consumo

Fuente: Elaboración propia

De la Figura N°95 tomamos los datos de las fechas y consumo total M3 para realizar los Pasos 2, 3 y 4.

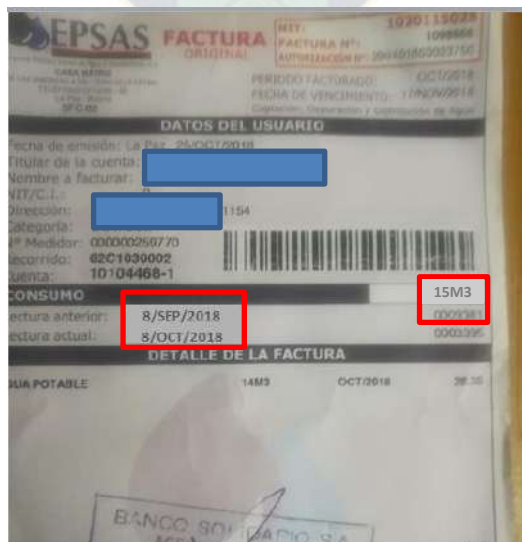


Figura N°95. Datos de factura de agua

Fuente: Elaboración propia

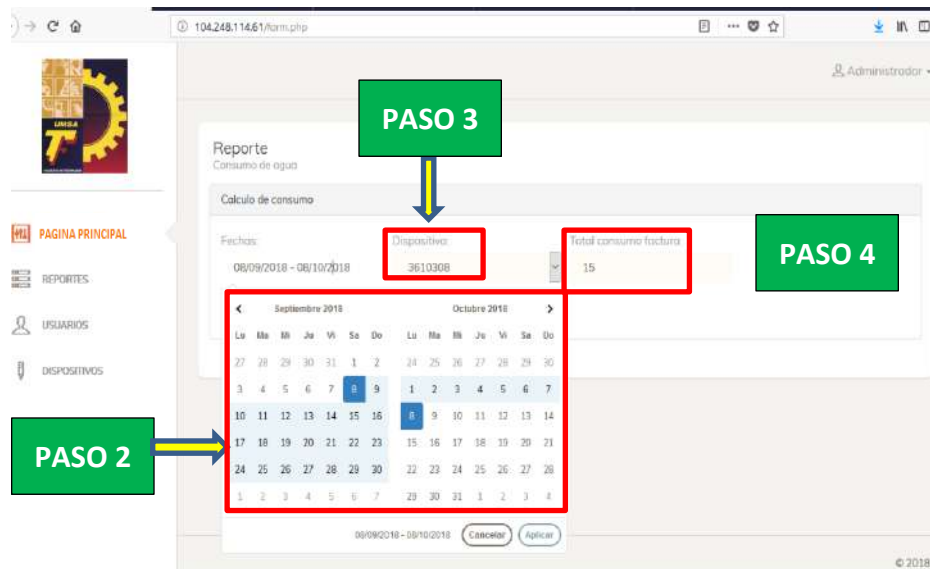


Figura N°96. Reporte de consumo

Fuente: Elaboración propia

Por ultimo presionamos calcular y podremos observar en la pantalla el reporte final donde se muestra el total a pagar en Bs por el consumo realizado

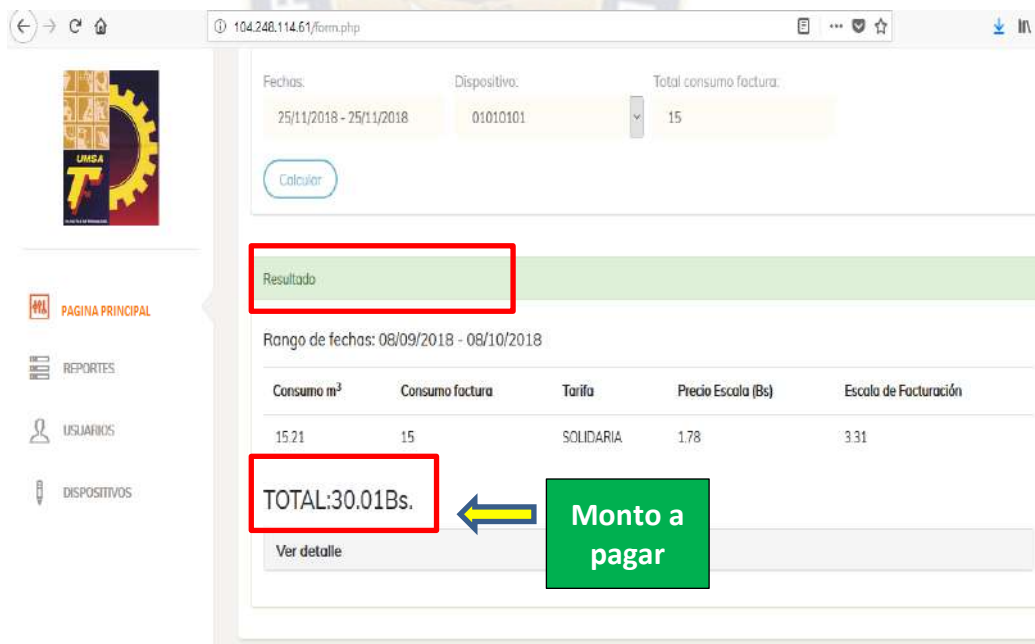


Figura N°97. Reporte final de consumo

Fuente: Elaboración propia

Para el segundo reporte seguiremos los mismos pasos para esta vez es del que está instalado en un domicilio en la ciudad de La Paz el cual cuenta con cuatro departamentos, el prototipo de medición de consumo hará el reporte de forma individual de uno de los departamentos donde está instalado.

De la Figura N°98 que es la factura de cobro del domicilio, tomamos los datos de las fechas y consumo total M3 para realizar los Pasos 2, 3 y 4

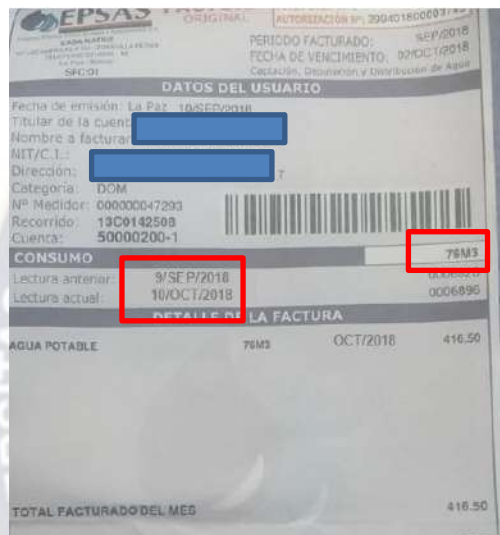


Figura N°98. Datos de factura de agua
Fuente: Elaboración propia

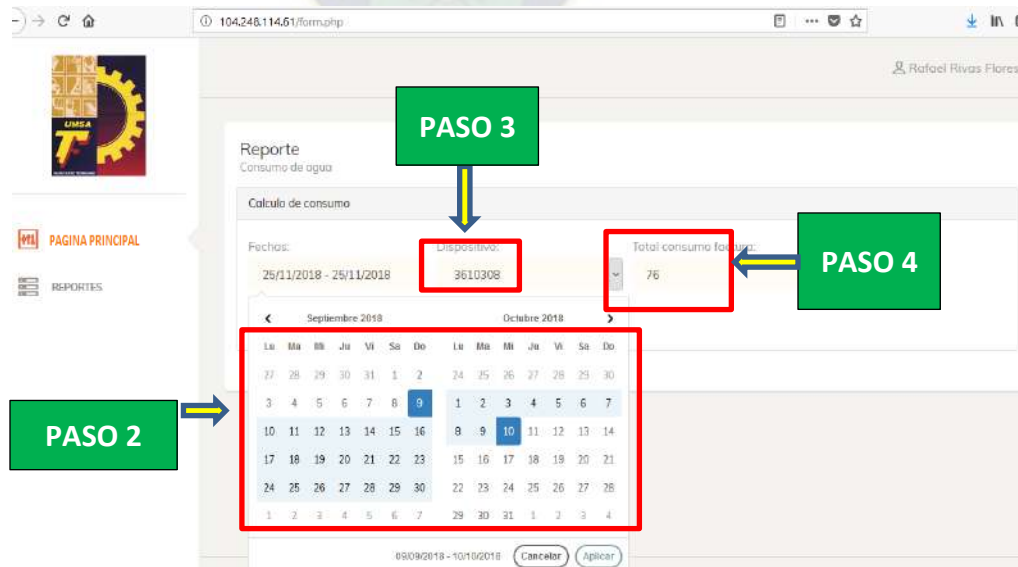


Figura N°99. Reporte de consumo
Fuente: Elaboración propia

Por ultimo presionamos calcular y podremos observar en la pantalla el reporte final donde se muestra el total a pagar en Bs por el consumo realizado.

104.248.114.61/form.php

Fechas: 25/11/2018 - 25/11/2018 Dispositivo: 3610308 Total consumo factura: 76

Calcular

Resultado

Rango de fechas: 09/09/2018 - 10/10/2018

Consumo m ³	Consumo factura	Tarifa	Precio Escala (Bs)	Escala de Facturación
15.21	76	MEDIA	6.68	5.20

TOTAL:81.57Bs.

Monto a pagar

Figura N°100. Reporte final de consumo
Fuente: Elaboración propia

En la Figura N°100 podemos observar que el consumo realizado en M3 es de 15 M3 de los 76 M3 totales consumido y el monto a pagar es de 81.57 Bs. De un total de 416.50 Bs.



CAPITULO IV

ANÁLISIS DE COSTOS

4.1 INTRODUCCIÓN

En esta sección se realizará el análisis sobre el costo del proyecto, que implica los costos fijos y variables, que conciernen en el diseño y recursos empleados para la realización del prototipo.

4.2 COSTOS FIJOS

Respecto al costo del hardware que se tiene en el desarrollo del prototipo, el detalle se explica en la tabla N° 4.

ITEM	DESCRIPCIÓN	CANTIDAD	COSTO UNITARIO (Bs)	COSTO TOTAL (Bs)
1	Caudalímetro (1/2" - 3/4")	1	60	60
2	Fuente de alimentación 5V	1	35	35
3	Esp8266NoceMCU	1	50	50
4	Zócalo 30 pines	1	10	15
5	Bornes de conexión	1	7	7
6	Resistencia	1	0.50	0.50
7	Placa virgen	1	15	15
8	Accesorios	1	50	50
TOTAL COSTOS				232.50 Bs

Tabla N°5. Detalle costo de componentes electrónicos
Fuente: Elaboración propia

4.3 COSTOS VARIABLES

La tabla del costo variable considera solo el pago por el servicio de la máquina virtual (Droplet) en DigitalOcean que dependerá del requerimiento que nuestro servidor web necesite.

ITEM	DESCRIPCIÓN	COSTO(Bs)
1	Máquina virtual (Droplet)	35
COSTO TOTAL		35 BS

Tabla N°6. Servicios de VPN
Fuente: Elaboración propia

4.4 COSTO TOTAL

El costo total incurrido para la implementación del prototipo no es más que la suma de los costos fijos y variables analizados anteriormente. La tabla siguiente muestra el costo total.

ITEM	DESCRIPCIÓN	COSTO TOTAL(Bs)
1	Componentes electrónicos	232,5
2	Servicios VPN	35
COSTO TOTAL		267.5 Bs

Tabla N°7. Costo total de la implementación del Prototipo
Fuente: Elaboración propia

Aun tomando el costo total, la implementación del presente proyecto como prototipo para su uso comercial es factible.



CAPITULO V

CONCLUSIONES Y

RECOMENDACIONES

5.1 CONCLUSIONES

Concluido la implementación del proyecto, en base a las experiencias obtenidas, información expuesta, pruebas realizadas y a los resultados obtenidos durante el desarrollo del prototipo se ha llegado a las siguientes conclusiones.

- Con el diseño y la implementación de un prototipo electrónico de medición, control y registro de consumo de agua potable para aplicación doméstica, se ha conseguido realizar un prototipo cuyo funcionamiento es aceptable debido a que las medidas tienen buena precisión.
- El diseño de la interfaz web ofrece un entorno amigable para el usuario, donde interactua de forma sencilla y fácil, teniendo en cuenta que la interfaz web cuenta con la seguridad correspondiente.
- Las pruebas realizadas del funcionamiento del prototipo fueron satisfactorias. Según se esperaba al comienzo del proyecto.

5.2 RECOMENDACIONES

De acuerdo a los resultados obtenidos en el diseño e implementación, así como en las pruebas de funcionamiento del proyecto se realizan las respectivas recomendaciones.

- Es recomendable que este prototipo sea considerado a la hora de las instalaciones de las tuberías de agua potable para su implementación.
- Para un proyecto futuro se podría adicionar la posibilidad de efectuar el corte de suministro del servicio via remota en caso de fugas de agua.

- Al tener instalado un servidor web y una base de datos en nuestra máquina virtual es recomendable no sobrecargar de aplicaciones que no sean necesarias que consuman recursos y que puedan afectar el rendimiento del mismo.



REFERENCIAS BIBLIOGRÁFICAS

1. L. Flóyd, T. (2008). *Dispositivos electrónicos*. México: Editorial Pearson Educación.
2. Ogáta, K. (2010). *Ingeniería de control moderna*. España: Editorial Prentice Hall.
3. Sánchez Reyes, C. (2005). *Sistemas de riego uso, manejo e instalación*. Perú: Editorial Ripalme.
4. Sampieri, C., y Fernández, C. (1997). *Metodología de investigación. Colombia*. Editorial McGraw-Hill.
5. Pomares Baeza Jorge. (2009). Manual de Arduino.

INTERNET

1. Esp8266NodeMCU. Recuperado de:
<https://www.luisllamas.es/esp8266-nodemcu/>
2. Modelos ESP8266. Recuperado de:
<https://www.prometec.net/modelos-esp8266/>
3. Esp8266NodeMCU cliente. Recuperado de:
<http://www.esploradores.com/conexion-del-procesador-esp8266-como-cliente-de-un-servidor-web/>
4. Estructura de la IDE arduino. Recuperado de:
<https://openwebinars.net/blog/tutorial-arduino-ide-arduino/>
5. Domótica con Esp8266. Recuperado de:
<https://secmotic.com/encender-bombilla-con-un-modulo-wifi-esp8266-arduino/>
6. Tipos de medidores. Recuperado de:
<https://www.quiminet.com/articulos/tipos-de-medidores-de-flujo-2561475.htm>

7. Medidores de flujo. Recuperado de:
<http://www.monografias.com/trabajos31/medidores-flujo/medidores-flujo.shtml>
8. Caudalímetros. Recuperado de:
<https://electronica.mercadolibre.com.ar/componentes-electronicos/caudalimetro-para-arduino#>
9. Servidor web LAMP. Recuperado de:
<https://www.ochobitshacenunbyte.com/2018/06/20/instalar-y-configurar-un-servidor-web-lamp-en-ubuntu-18-04/>
10. Servidores web. Recuperado de:
<https://duplika.com/blog/que-son-los-servidores-web-y-por-que-son-necesarios/>
11. Base de datos MySQL. Recuperado de:
<http://www.oscarabadfolgueira.com/crear-una-base-datos-mysql-desde-consola/>
12. Instalación PHP. Recuperado de:
<https://www.redeszone.net/2017/08/26/instalar-php-7-0-php-7-1-ubuntu/>
13. Instalar phpMyAdmin. Recuperado de:
<https://www.digitalocean.com/community/tutorials/como-instalar-y-proteger-phpmyadmin-en-ubuntu-16-04-es>
14. Servicios VPN en DigitalOcean. Recuperado de :
<https://www.digitalocean.com/community/tutorials/como-configurar-un-servidor-openvpn-en-ubuntu-16-04-es>

GLOSARIO

FTP protocolo (“File Transfer Protocol”, o “Protocolo de Transferencia de Archivos”) se usa para intercambiar archivos entre el cliente y el servidor. Para que dicha transferencia de archivos sea posible es necesario un cliente FTP y un servidor

GPIO por sus siglas en ingles (*General Purpose Input/Output*, Entrada/Salida de Propósito General) es un pin genérico en un chip, cuyo comportamiento (incluyendo si es un pin de entrada o salida) se puede controlar (programar) por el usuario en tiempo de ejecución.

HTTP son las siglas de “Hypertext Transfer Protocol” es un protocolo de transferencia donde se utiliza un sistema mediante el cual se permite la transferencia de información entre diferentes servicios y los clientes que utilizan páginas web.

MCU por sus siglas en inglés de microcontroller unit, un chip que contiene procesador, RAM, ROM, reloj y la unidad de control de E/S en un único encapsulado, un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria

PCB de sus siglas en ingles *PCB (Printed Circuit Board)* es la superficie constituida por caminos, pistas o buses de material conductor laminadas sobre una base no conductora. El circuito impreso se utiliza para conectar eléctricamente a través de las pistas conductoras, y sostener mecánicamente, por medio de la base, un conjunto de componentes electrónicos

P2P son las siglas en inglés de *Peer-to-Peer*, que se puede traducir como “comunicación entre iguales” las redes P2P son una red de computadoras que funciona sin necesidad de contar ni con clientes ni con servidores fijos.

SERVIDOR Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente

TCP significa Transmission Control Protocol, proporciona un servicio orientado a conexión y fiable.

UDP significa User Datagram Protocol, proporciona un servicio no orientado a conexión y no fiable, esto quiere decir que se va a intentar por todos los medios que los datos lleguen, pero no lo garantiza.

WI-FI es un mecanismo que permite, de forma inalámbrica, el acceso a Internet de distintos dispositivos al conectarse a una red determinada. Esta tecnología, al tiempo que ofrece la entrada a la gran red de redes, vincula diferentes equipos entre sí sin la necesidad de cables.



ANEXOS

Código tarjeta de desarrollo

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>

int contconexion = 0;
const char *ssid = "*****";
const char *password = "*****";

char host[48];
String strhost = "xx.yy.xx.yy" ; //
String strurl = "../enviardatos.php"; //
String chipid = "";

volatile int NumPulsos;
int PinSensor = 4;
float factor_conversion = 6;
float volumen = 0;
float m3 = 0;
long dt = 0;
long t0 = 0;
float ac = 0;
unsigned long anterior_Millis = 0;

String enviardatos(String datos) {
  String linea = "error";
  WiFiClient client;
  strhost.toCharArray(host, 49);
  if (!client.connect(host, 80)) {
    Serial.println("Fallo de conexion");
    return linea;
  }
}

client.print(String("POST ") + strurl + " HTTP/1.1" + "\r\n" +
  "Host: " + strhost + "\r\n" +
  "Accept: */*" + "\r\n" +
  "Content-Length: " + datos.length() + "\r\n" +
  "\r\n" + datos);
delay(10);

Serial.print("Enviando datos a SQL...");
unsigned long timeout = millis();
while (client.available() == 0) {
  if (millis() - timeout > 5000) {
    Serial.println("Cliente fuera de tiempo!");
  }
}
```

```

        client.stop();
        return linea;
    }
}

while (client.available()) {
    linea = client.readStringUntil('\r');
}
Serial.println(linea);
return linea;
}

void ContarPulsos ()
{
    NumPulsos++;
}

int ObtenerFrecuecia()
{
    int frecuencia;
    NumPulsos = 0
    interrupts();
    delay(1000);
    noInterrupts();
    frecuencia = NumPulsos;
    return frecuencia;
}

void setup()
{
    Serial.begin(9600);
    Serial.println("");

    Serial.print("chipId: ");
    chipid = String(ESP.getChipId());
    Serial.println(chipid);
    pinMode(PinSensor, INPUT);
    attachInterrupt(digitalPinToInterrupt(PinSensor), ContarPulsos, RISING);
    Serial.println ("restablecer el volumen a 0 Litros");
    t0 = millis();
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED and contconexion < 50) {    ++contconexion;
        delay(500);
        Serial.print(".");
    }
    if (contconexion < 50) {
        //para usar con ip fija
        IPAddress ip(192, 168, 1, 147);
        IPAddress gateway(192, 168, 1, 1);
        IPAddress subnet(255, 255, 255, 0);
        WiFi.config(ip, gateway, subnet);

        Serial.println("");
        Serial.println("WiFi conectado");
        Serial.println(WiFi.localIP());
    }
}

```

```

}
else {
  Serial.println("");
  Serial.println("Error de conexion");
}
}
}

void loop ()
{
  unsigned long actual_Millis = millis();
  if (Serial.available()) {
    if (Serial.read() == 'r')volumen = 0;
  }

  float frecuencia = ObtenerFrecuecia();
  float caudal_L_m = frecuencia / factor_conversion;
  dt = millis() - t0;
  //if (dt >= 5000) {
  t0 = millis();
  volumen = volumen + (caudal_L_m / 60) * (dt / 1000); // volumen(L)=caudal(L/s)*tiempo(s)
  ac = (volumen / 1000);
  if (volumen >= 1000) {
    m3 = m3 + 1;
    volumen = 0;
    ac = 0;
  }
  if (caudal_L_m != 0) {
    if (actual_Millis - anterior_Millis >= 300000) {
      //-----Enviamos por el puerto serie-----
      Serial.print ("Caudal: ");
      Serial.print (caudal_L_m, 3);
      Serial.print ("L/min\tVolumen: ");
      Serial.print (volumen, 0);
      Serial.print (" L\tConsumo:");
      Serial.print (m3 + ac, 3);
      Serial.println (" M3");

      enviardatos("chipid=" + chipid + "caudal=" + String(caudal_L_m, 3) + "litros=" +
String(volumen, 0) + "consumo_M3=" + String(m3 + ac, 3));
      delay(50);
    }
  }
}
}
}

```

Código fuente enviardatos.php

```

<?php
$conexion = mysql_connect('localhost', 'XXXX', 'XXXXXX');
//$conexion = mysql_connect('xxxxt', 'xxx', 'xxxx');
mysql_select_db('db', $conexion);
mysql_query("SET NAMES 'utf8'");

$chipid = $_POST ['chipid'];
$caudal = $_POST ['caudal'];
$litros = $_POST ['litros'];
$consumo_M3 = $_POST ['consumo_M3'];

```

```
mysql_query("INSERT INTO `db`.`registro` (`id`, `chipld`, `fecha`, `caudal`, `litros`, `consumo_M3`)
VALUES (NULL, '$chipld', CURRENT_TIMESTAMP, '$caudal', '$litros', '$consumo_M3');");

mysql_close();

echo "Datos ingresados correctamente.";
?>
```

Código fuente para el login

```
<?php
include 'class/connection.php';
include 'class/usuario.php';
$usuario=new Usuario();
session_start();
if($_GET && isset($_GET['logout'])){
    $_SESSION['user']=NULL;
}
if($_POST){
    $u=$_POST['usuario'];
    $p=$_POST['password'];
    $d=$usuario->check($u, $p);
    if($d!=NULL){
        $_SESSION['user'] = array('id'=>$d->getId(), 'nombre'=>$d->getNombre(), 'rol'=>$d->getRol());
        header('location:dashboard.php');
    }else{
    }
}
}

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Agua</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
<!--
=====
----->
    <link rel="icon" type="image/png" href="assets/images/icons/favicon.ico"/>
<!--
=====
----->
    <link rel="stylesheet" type="text/css" href="assets/vendor/bootstrap/css/bootstrap.min.css">
<!--
=====
----->
    <link rel="stylesheet" type="text/css" href="assets/fonts/font-awesome-4.7.0/css/font-
awesome.min.css">
```

```

<!--
=====
----->
    <link rel="stylesheet" type="text/css" href="assets/fonts/iconic/css/material-design-iconic-
font.min.css">
<!--
=====
----->
    <link rel="stylesheet" type="text/css" href="assets/vendor/animate/animate.css">
<!--
=====
----->
    <link rel="stylesheet" type="text/css" href="assets/vendor/css-hamburgers/hamburgers.min.css">
<!--
=====
----->
    <link rel="stylesheet" type="text/css" href="assets/vendor/animation/css/animation.min.css">
<!--
=====
----->
    <link rel="stylesheet" type="text/css" href="assets/vendor/select2/select2.min.css">
<!--
=====
----->
    <link rel="stylesheet" type="text/css" href="assets/vendor/daterangepicker/daterangepicker.css">
<!--
=====
----->
    <link rel="stylesheet" type="text/css" href="assets/css/util.css">
    <link rel="stylesheet" type="text/css" href="assets/css/main.css">
<!--
=====
----->
</head>
<body>
    <div class="limiter">
        <div class="container-login100" style="background-image: url('assets/images/bg-
01.jpg');">
            <div class="wrap-login100">
                <form class="login100-form validate-form" method="post">
                    <span style="background-color: transparent" class="login100-
form-logo">
                        
                    </span>

                    <span class="login100-form-title p-b-34 p-t-27">
                        Iniciar Sesión
                    </span>

                    <div class="wrap-input100 validate-input" data-validate =
"Enter username">
                        <input class="input100" type="text" name="usuario"
placeholder="usuario">

```

```

placeholder="&#xf207;"></span>
</div>
<div class="wrap-input100 validate-input" data-validate="Enter
password">
<input class="input100" type="password"
name="password" placeholder="Contraseña">
<span class="focus-input100" data-
placeholder="&#xf191;"></span>
</div>
<div class="container-login100-form-btn">
<button class="login100-form-btn">
Entrar
</button>
</div>
</form>
</div>
</div>
</div>
<div id="dropDownSelect1"></div>
<!--
=====
----->
<script src="assets/vendor/jquery/jquery-3.2.1.min.js"></script>
<!--
=====
----->
<script src="assets/vendor/animstion/js/animstion.min.js"></script>
<!--
=====
----->
<script src="assets/vendor/bootstrap/js/popper.js"></script>
<script src="assets/vendor/bootstrap/js/bootstrap.min.js"></script>
<!--
=====
----->
<script src="assets/vendor/select2/select2.min.js"></script>
<!--
=====
----->
<script src="assets/vendor/daterangepicker/moment.min.js"></script>
<script src="assets/vendor/daterangepicker/daterangepicker.js"></script>
<!--
=====
----->
<script src="assets/vendor/countdowntime/countdowntime.js"></script>
<!--
=====
----->
<script src="assets/js/main.js"></script>

```

```
</body>
</html>
```

Código fuente usuarios.php

```
<?php
require_once 'layout/header.php';
include 'class/connection.php';
include 'class/Registro.php';
include 'class/Chip.php';
include 'class/escala.php';
include 'class/usuario.php';
if ($_POST) {
    $r = FALSE;
    if ($_POST["id"]=="") {
        if (isset($_POST["usuario"]) && isset($_POST["password"])) {
            $usuario = new Usuario();
            $usuario->setUsuario($_POST['usuario']);
            $usuario->setNombre($_POST['nombre']);
            $usuario->setPassword($_POST['password']);
            $usuario->setRol($_POST['rol']);
            $usuario->add();
            $r = true;
        }
    } else {
        $usuario = new Usuario();
        $usuario = $usuario->_getUsuario($_GET['id']);
        $usuario->setUsuario($_POST['usuario']);
        $usuario->setNombre($_POST['nombre']);
        $usuario->setPassword($_POST['password']);
        $usuario->setRol($_POST['rol']);
        $usuario->update();
    }
}
$usuario = new Usuario();
if ($_GET && isset($_GET['id'])) {
    $r = FALSE;
    $usuario = $usuario->_getUsuario($_GET['id']);
}

if ($_GET && isset($_GET['del'])) {
    $r = FALSE;
    $usuario = $usuario->_getUsuario($_GET['del']);
    $usuario->delete();
}
$usuarios = new Usuario();
$usuarios = $usuarios->getUsuarios();
```



```

?>
<div class="main-panel">
  <?php include_once 'layout/top-nav.php'; ?>

  <div class="content">
    <div class="container-fluid">
      <div class="row">
        <div class="col-md-12">
          <div class="card ">
            <div class="header">
              <h4 class="title">Usuarios</h4>
              <p class="category">Usuarios</p>
            </div>
            <div class="content">
              <div class="panel panel-default">
                <div class="panel-heading">Nuevo</div>
                <div class="panel-body">
                  <form method="post" action="">
                    <div class="row">
                      <div class="col-md-6">
                        <div class="form-group">
                          <label for="exampleInputEmail1">Usuario:</label>
                          <input required name="usuario" value="<?php echo $usuario->getUsuario(); ?>" type="text" class="form-control" placeholder="ej:juan.perez">
                        </div>
                      </div>
                      <div class="col-md-6">
                        <div class="form-group">
                          <label for="exampleInputEmail1">Password:</label>
                          <input required name="password" value="<?php echo $usuario->getPassword(); ?>" type="password" class="form-control" placeholder="">
                        </div>
                      </div>
                      <div class="col-md-6">
                        <div class="form-group">
                          <label for="exampleInputPassword1">Nombre:</label>
                          <input required type="text" name="nombre" value="<?php echo $usuario->getNombre(); ?>" class="form-control" placeholder="ej: Juan Perez">
                          <input type="hidden" name="id" value="<?php echo $usuario->getId(); ?>">
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
                <div class="col-md-6">
                  <div class="form-group">
                    <label for="exampleInputPassword1">Rol:</label>
                    <select class="form-control" name="rol">
                      <?php
                      if ($_GET) {
                        if ($usuario->getRol() == 'ROLE_ADMIN') {
                          ?>
                          <option selected value="ROLE_ADMIN">Administrador</option>
                          <option value="ROLE_USER">Usuario</option>
                        } else { ?>
                          <option value="ROLE_ADMIN">Administrador</option>

```

```

        <option selected value="ROLE_USER">Usuario</option>
    <?php
    }
    } else {
    ?>
        <option value="ROLE_ADMIN">Administrador</option>
        <option value="ROLE_USER">Usuario</option>
    <?php } ?>
    </select>
</div>
</div>
</div>
</div>
<button type="submit" class="btn btn-info">Guardar</button>
</form>
</div>
</div>
<table class="table">
<tr><th>Usuario</th><th>Password</th><th>Nombre</th><th>Rol</th><th>Acciones</th></tr>
<?php
    foreach ($usuarios as $usuario) {
        echo '<tr><td>' . $usuario->getUsuario() . '</td><td>' . $usuario->getPassword() .
'</td><td>' . $usuario->getNombre() . '</td><td>' . str_replace('ROLE_', "", $usuario->getRol()) . '</td>'
        . '<td><a href="usuarios.php?id=' . $usuario->getId() . '" class="btn btn-info"><i
class="ti-pencil-alt2"></i></a><a href="javascript:deleteChip(' . $usuario->getId() . ');" class="btn btn-
info"><i class="ti-close"></i></a></td></tr>';
    }
    ?>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script>
    function deleteChip(id) {
        console.log(id);
        bootbox.confirm("Confirme...!", function (result) {
            if (result) {
                document.location.href = 'usuarios.php?del=' + id;
            }
        });
    }
</script>
<script>
<?php if ($r) { ?>
    $.notify({
        icon: "ti-gift",
        message: "Dispositivo agregado correctamente"
    }, {

```

```

        type: 'success',
        timer: 4000
    });
<?php } else { ?>
    $.notify({
        icon: "ti-gift",
        message: "Sucedió un error al agregar el dispositivo"

    }, {
        type: 'danger',
        timer: 4000
    });
<?php } ?>
</script>

<?php
require_once 'layout/footer.php';
?>

```

Código fuente dispositivos.php

```

<?php
require_once 'layout/header.php';
include 'class/connection.php';
include 'class/Registro.php';
include 'class/Chip.php';
include 'class/escala.php';
include 'class/usuario.php';
$msg = "";
if ($_POST) {
    $r = FALSE;
    $msg = "";
    if (isset($_POST["usuario"]) && isset($_POST['chipid'])) {
        $chip = new Chip();
        $usuario = new Usuario();
        $usuario->setId($_POST['usuario']);
        $chip->setChipid($_POST['chipid']);
        $chip->setDescripcion(isset($_POST['descripcion']) ? $_POST['descripcion'] : "");
        $chip->setUbicacion(isset($_POST['ubicacion']) ? $_POST['ubicacion'] : "");
        $chip->setTarifa($_POST['tarifa']);
        $chip->setUsuario($usuario);
        if (isset($_GET["id"])) {
            $r = $chip->update();
        }else{
            $r = $chip->add();
        }
    }

    $_POST = NULL;
    if ($r) {

```

```

        $msg = 'Dispositivo agregado correctamente';
    } else {
        $msg = 'Sucedio un error al agregar el dispositivo';
    }
}
}
}
if ($_GET) {
    $r = FALSE;
    if (isset($_GET["dev"])) {
        $chip = new Chip();
        $chip->setChipid($_GET["dev"]);
        $r = $chip->delete();
        if ($r) {
            $msg = 'Dispositivo eliminado';
        } else {
            $msg = 'Sucedio un error al eliminar el dispositivo';
        }
    }

    if (isset($_GET["id"])) {
        $chip = new Chip();
        $chip = $chip->getDispositivo($_GET["id"]);
    }
} else {
    $chip = new Chip();
}
$chips = new Chip();
$usuarios = new Usuario();
$usuarios = $usuarios->getUsuariosLibres();
$chips = $chips->getDispositivos();
?>
<div class="main-panel">
    <?php include_once 'layout/top-nav.php'; ?>

    <div class="content">
        <div class="container-fluid">
            <div class="row">
                <div class="col-md-12">
                    <div class="card ">
                        <div class="header">
                            <h4 class="title">Dispositivos</h4>
                            <p class="category">Dispositivos </p>
                        </div>
                        <div class="content">
                            <div class="panel panel-default">
                                <div class="panel-heading">Nuevo</div>
                                <div class="panel-body">
                                    <form method="post" action="">
                                        <div class="row">
                                            <div class="col-md-6">
                                                <div class="form-group">
                                                    <label for="exampleInputEmail1">Chip ID:</label>
                                                    <input required name="chipid" type="text" value="<?php echo $chip->getChipid(); ?>" class="form-control" placeholder="ej:454545">

```

```

        </div>
    </div>
    <div class="col-md-6">
        <div class="form-group">
            <label for="exampleInputEmail1">Descripción:</label>
            <input name="descripcion" value="<?php echo $chip->getDescripcion();
?>" type="text" class="form-control" placeholder="">
        </div>
    </div>
    <div class="col-md-6">
        <div class="form-group">
            <label for="exampleInputPassword1">Ubicación:</label>
            <input type="text" name="ubicacion" value="<?php echo $chip-
>getUbicacion(); ?>" class="form-control" placeholder="">
        </div>
    </div>
    <div class="col-md-6">
        <div class="form-group">
            <label for="exampleInputPassword1">Usuario:</label>
            <select required class="form-control" name="usuario">
                <?php
                if (isset($_GET['id'])) {
                    echo '<option selected value="" . $chip->getUsuario()->getId() . "">' .
$chip->getUsuario()->getNombre() . '</option>';
                }
                foreach ($usuarios as $usuario) {
                    echo '<option value="" . $usuario->getId() . "">' . $usuario-
>getNombre() . '</option>';
                }
                ?>
            </select>
        </div>
    </div>
    <div class="col-md-6">
        <div class="form-group">
            <label for="exampleInputPassword1">Tarifa:</label>
            <select required class="form-control" name="tarifa">
                <?php
                if ($chip->getTarifa() == '3.31') {
                    echo '<option selected value="3.31">El Alto</option>';
                    echo '<option value="5.20">La Paz</option>';
                } else {
                    echo '<option value="3.31">El Alto</option>';
                    echo '<option selected value="5.20">La Paz</option>';
                }
                ?>
            </select>
        </div>
    </div>
    <div class="col-md-6">
        <div class="form-group">
            <button type="submit" class="btn btn-info">Guardar</button>
            <button type="reset" class="btn btn-info">Cancelar</button>
        </div>
    </div>

```

```

        </form>
    </div>
</div>
</div>
<table class="table">
    <tr><th>Chip
ID</th><th>Descripción</th><th>Ubicación</th><th>Tarifa</th><th>Usuario</th><th>Acciones</th></tr>
    <?php
        foreach ($chips as $chip) {
            echo ' <tr><td>' . $chip->getChipid() . '</td><td>' . $chip->getDescripcion() .
'</td><td>' . $chip->getUbicacion() . '</td><td>' . $chip->getTarifa() . '</td><td>' . $chip->getUsuario()-
>getNombre() . '</td>'
                . '<td><button onclick="editChip(\'' . $chip->getChipid() . '\');" class="btn btn-info"><i
class="ti-pencil-alt2"></i></button><button onclick="deleteChip(\'' . $chip->getChipid() . '\');" class="btn btn-
info"><i class="ti-close"></i></button></td></tr>';
            }
        }?>
    </table>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script>
function deleteChip(id) {
    console.log(id);
    bootbox.confirm("Confirme...!", function (result) {
        if (result) {
            document.location.href = 'dispositivos.php?dev=' + id;
        }
    });
}
function editChip(id) {
    console.log(id);
    document.location.href = 'dispositivos.php?id=' + id;
}
function updateChip(id) {
    console.log(id);
    bootbox.confirm("Confirme...!", function (result) {
        if (result) {
            document.location.href = 'dispositivos.php?edt=' + id;
        }
    });
}
}
</script>
<script>
<?php if (isset($r) && $msg != "") { ?>
    $.notify({
        icon: "ti-gift",
        message: "<?php echo $msg; ?>"
    }, {
}

```

```

        type: 'success',
        timer: 4000
    });
    <?php } elseif (isset($r) && $msg != "") { ?>
        $.notify({
            icon: "ti-gift",
            message: "<?php echo $msg; ?>"

        }, {
            type: 'danger',
            timer: 4000
        });
    <?php } ?>
</script>

<?php
require_once 'layout/footer.php';
?>

```

Código fuente formulario reporte

```

<?php
require_once 'layout/header.php';
include 'class/connection.php';
include 'class/Registro.php';
include 'class/Chip.php';
include 'class/escala.php';
include 'class/usuario.php';

$chips = new Chip();

if($u['rol']=='ROLE_ADMIN'){
    $chips = $chips->getDispositivos();
}else{
    $chips = array($chips->getDispositivoPorUsuario($u['id']));
}

```

```

$consumo = 0;
if ($_POST) {
    $fechas = $_POST['fechas'];
    $_fechas = $fechas;
    $chipid = $_POST['chipid'];
    $_chip=new Chip();
    $_chip=$_chip->getDispositivo($chipid);
    $fechas = explode(' - ', $fechas);
    $f1 = explode('/', $fechas[0]);
    $f2 = explode('/', $fechas[1]);
    $fecha1 = $f1[2] . '-' . $f1[1] . '-' . $f1[0];
    $fecha2 = $f2[2] . '-' . $f2[1] . '-' . $f2[0];
    $registro = new Registro();
    $r = $registro->getRegistroEntreFechas($chipid, $fecha1, $fecha2);
    if (!$r['err']) {
        $consumo = 0;
        if (is_numeric($_POST['consumo'])) {
            $consumo = $_POST['consumo'];
        }
        $escala = $registro->getEscala($consumo);
        $registro = null;
    }
}
?>
<div class="main-panel">
<?php include_once 'layout/top-nav.php'; ?>
<div class="content">
    <div class="container-fluid">
        <div class="row">

```



```

<div class="col-md-12">
  <div class="card ">
    <div class="header">
      <h4 class="title">Reporte</h4>
      <p class="category">Consumo de agua</p>
    </div>
    <div class="content">
      <div class="panel panel-default">
        <div class="panel-heading">Calculo de consumo</div>
        <div class="panel-body">
          <form method="post" action="">
            <div class="row">
              <div class="col-md-3">
                <div class="form-group">
                  <label for="exampleInputEmail1">Fechas:</label>
                  <input name="fechas" type="text" class="form-control"
placeholder="Email">
                </div>
              </div>
              <div class="col-md-3">
                <div class="form-group">
                  <label for="exampleInputEmail1">Dispositivo:</label>
                  <select class="form-control" name="chipid">
<?php
foreach ($chips as $chip) {
    echo '<option value="' . $chip->getChipid() . '">' . $chip->getChipid() . '</option>';
}
?>
                  </select>
                </div>
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>

```

```

</div>

<div class="col-md-3">

    <div class="form-group">

        <label for="exampleInputPassword1">Total consumo factura:</label>

        <input type="text" value="<?php echo $consumo; ?>" name="consumo"
class="form-control" placeholder="ej: 75">

    </div>

</div>

</div>

</div>

<button type="submit" class="btn btn-info">Calcular</button>

</form>

</div>

</div>

<br>

<?php if ($_POST && !$r['err']) { ?>

    <div class="panel panel-success">

        <div class="panel-heading">Resultado</div>

        <div class="panel-body">

            <?php
            echo '<h5>Rango de fechas: ' . $_fechas . '</h5>';
            echo '<table class="table">';
            echo '<tr>';
            echo '<th>Consumo m<sup>3</sup></th><th>Consumo factura</th><th>Tarifa</th><th>Precio Escala
(Bs)</th><th>Escala de Facturación</th>';
            echo '<tr>';
            echo '<tr>';
            echo '<td>' . $r['calculo'] . '</td><td>' . $consumo . '</td><td>' . $escala[0]->getNombre() . '</td><td>' .
$escala[0]->getPrecio() . '</td><td>' . $_chip->getTarifa().</td>';
            echo '<tr>';
            echo '</table>';

```

```

    echo '<h3>TOTAL:' . number_format(Registro::calcula($r['calculo'], $consumo, $escala[0]->getPrecio(),
    $_chip->getTarifa()),2) . 'Bs.</h3>';

```

```

?>

```

```

<div class="panel-group" id="accordion" role="tablist" aria-multiselectable="true">

```

```

    <div class="panel panel-default">

```

```

        <div class="panel-heading" role="tab" id="headingOne">

```

```

            <h4 class="panel-title">

```

```

                <a role="button" data-toggle="collapse" data-parent="#accordion"
                href="#collapseOne" aria-expanded="false" aria-controls="collapseOne">

```

```

                    Ver detalle

```

```

                </a>

```

```

            </h4>

```

```

        </div>

```

```

        <div id="collapseOne" class="panel-collapse collapse" role="tabpanel" aria-
        labelledby="headingOne">

```

```

            <div class="panel-body">

```

```

                <?php

```

```

                    echo '<table class="table">';

```

```

                    echo '<tr>';

```

```

                    echo '<th>Chip ID</th><th>Fecha</th><th>Caudal</th><th>Litros</th><th>Consumo
                    m<sup>3</sup></th>';

```

```

                    echo '</tr>';

```

```

                    foreach ($r['detalle'] as $item) {

```

```

                        echo '<tr>';

```

```

                            echo '<td>' . $item->getChipid() . '</td><td>' . $item->getFecha() . '</td><td>' . $item->getCaudal() .
                            '</td><td>' . $item->getLitros() . '</td><td>' . $item->getConsumo_M3() . '</td>';

```

```

                        echo '</tr>';

```

```

                    }

```

```

                    echo '</table>';

```

```

                ?>

```

```

            </div>

```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<?php } ?>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<script>
```

```
$(document).ready(function () {
```

```
  $('input[name="fechas"]').daterangepicker({
```

```
    "locale": {
```

```
      "format": "DD/MM/YYYY",
```

```
      "separator": " - ",
```

```
      "applyLabel": "Aplicar",
```

```
      "cancelLabel": "Cancelar",
```

```
      "fromLabel": "Desde",
```

```
      "toLabel": "Hasta",
```

```
      "customRangeLabel": "Custom",
```

```
      "weekLabel": "W",
```

```
      "daysOfWeek": [
```

```
        "Do",
```

```
"Lu",
"Ma",
"Mi",
"Ju",
"Vi",
"Sa"
],
"monthNames": [
"Enero",
"Febrero",
"Marzo",
"Abril",
"Mayo",
"Junio",
"Julio",
"Agosto",
"Septiembre",
"Octubre",
"Noviembre",
"Diciembre"
],
"firstDay": 1
}
});
});
```



```
</script>
<?php
require_once 'layout/footer.php';
?>
```

