



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA  
PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

**Diseño de una tarjeta de desarrollo con ESP8266 orientada a  
wireless y microcontroladores para IoT**

AUTOR:

Higuera Astudillo, Luis Alberto

Componente práctico del examen complejo previo a la  
obtención del grado de **INGENIERO EN  
TELECOMUNICACIONES**

REVISOR:

M. Sc. Pacheco Bohórquez, Héctor Ignacio

Guayaquil, Ecuador

16 de septiembre del 2019



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

## **CERTIFICACIÓN**

Certificamos que el presente **componente práctico del examen complejo**, fue realizado en su totalidad por **Higuera Astudillo, Luis Alberto** como requerimiento para la obtención del título de **INGENIERO EN TELECOMUNICACIONES**.

REVISOR

---

M. Sc. Pacheco Bohórquez, Héctor Ignacio

DIRECTOR DE CARRERA

---

M. Sc. Heras Sánchez, Miguel Armando

Guayaquil, 16 de septiembre del 2019



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

### **DECLARACIÓN DE RESPONSABILIDAD**

Yo, **Higuera Astudillo, Luis Alberto**

#### **DECLARÓ QUE:**

**El componente práctico del examen complejo: Diseño de una tarjeta de desarrollo con ESP8266 orientada a wireless y microcontroladores para IoT**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, 16 de septiembre del 2019

EL AUTOR

---

HIGUERA ASTUDILLO, LUIS ALBERTO



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

### **AUTORIZACIÓN**

Yo, **Higuera Astudillo, Luis Alberto**

Autorizó a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del **componente práctico del examen complejo: Diseño de una tarjeta de desarrollo con ESP8266 orientada a wireless y microcontroladores para IoT**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, 16 de septiembre del 2019

EL AUTOR

---

HIGUERA ASTUDILLO, LUIS ALBERTO

# REPORTE DE URKUND

URKUND ★ Probar v Urkund Bloques Fernando Palacios Meléndez (edwin\_palacios)

<b>Documento</b>	<a href="#">Higuera_Complexivo_FINAL_2019A.docx</a> (D55665253)
<b>Presentado</b>	2019-09-16 20:32 (-05:00)
<b>Presentado por</b>	fernandopm23@hotmail.com
<b>Recibido</b>	edwin.palacios.ucsg@analysis.orkund.com
<b>Mensaje</b>	Revisión EC Luis Higuera <a href="#">Mostrar el mensaje completo</a>

19% de estas 13 páginas, se componen de texto presente en 1 fuentes.

Categoría	Enlace/nombre de archivo
	<a href="#">naranjo_gregory_final.doc</a>
	<a href="https://www.rutanmedellin.org/imagenes/biblioteca/ob...">https://www.rutanmedellin.org/imagenes/biblioteca/ob...</a>
<b>Fuentes alternativas</b>	
<b>Fuentes no usadas</b>	

0 Advertencias. Reiniciar Exportar Compartir

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERIA EN TELECOMUNICACIONES

TEMA: Diseño de una tarjeta de desarrollo con ESP8266 orientada a wireless y microcontroladores para IoT

AUTOR: Higuera Astudillo, Luis Alberto

Componente práctico del examen complejo

previo a la obtención del grado de INGENIERO EN TELECOMUNICACIONES

REVISOR: M. Sc. Pacheco Bohórquez, Héctor Ignacio

Guayaquil, Ecuador 16 de septiembre del 2019

## **DEDICATORIA**

A la persona que me robó el corazón desde el primer día en que supe de su existencia, la que llegó a mi vida en el momento indicado para cambiármela por completo, la que guía mi camino, la que me da la fuerza y el coraje para cada día ser mejor, mi hija Luisana. Esto va para ti mi pequeña, gracias por sacar lo mejor de mí. TE AMO.

**EL AUTOR**

**HIGUERA ASTUDILLO, LUIS ALBERTO**

## **AGRADECIMIENTO**

A Dios por bendecirme con salud y permitirme haber llegado hasta aquí.

A mis padres por ser mi mayor fuente de inspiración. Gracias por estar allí y ayudarme a cumplir todos mis sueños y aspiraciones. Sin ustedes no habría logrado dar este paso tan importante en mi vida.

A mis abuelos por darme ese aliento que me impulsa en todo lo que hago. Gracias por sus oraciones en las que estoy siempre presente.

A la mujer que centró mi vida y estuvo allí cuando más lo necesitaba, Ana. Gracias por tu gran amor y apoyo incondicional, por nunca soltarme de la mano y por darme esas ganas de seguir adelante.

EL AUTOR

HIGUERA ASTUDILLO, LUIS ALBERTO



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

**TRIBUNAL DE SUSTENTACIÓN**

f. \_\_\_\_\_  
**M. Sc. ROMERO PAZ, MANUEL DE JESÚS**  
DECANO

f. \_\_\_\_\_  
**M. Sc. HERAS SÁNCHEZ, MIGUEL ARMANDO**  
DIRECTOR DE CARRERA

f. \_\_\_\_\_  
**M. Sc. PALACIOS MELÉNDEZ, EDWIN FERNANDO**  
OPONENTE

## ÍNDICE GENERAL

ÍNDICE DE FIGURAS.....	XI
Resumen .....	XIII
CAPÍTULO 1: DESCRIPCIÓN DEL COMPONENTE PRÁCTICO .....	2
1.1. Introducción.....	2
1.2. Objetivo General. ....	3
1.3. Objetivos Específicos. ....	3
CAPÍTULO 2: Fundamentación teórica.....	4
2.1. Introducción de IoT.....	4
2.2. Aplicaciones del IoT. ....	5
2.2.1. Ciudades inteligentes .....	6
2.2.2. Seguridad y emergencias. ....	7
2.2.3. Agricultura inteligente .....	7
Capítulo 3: Diseño de una tarjeta de desarrollo con ESP8266 orientada a Wireless y microcontroladores para IoT. ....	9
3.1. Elementos utilizados para la implementación de la tarjeta de desarrollo con ESP8266. ....	9
3.1.1. Dispositivo ESP8266 .....	9
3.1.2. Microcontrolador Atmel 328p.....	10
3.1.3. Modulo USB-TTL.....	12
3.1.4. Módulo Step Down 1.5Amp.....	13
3.1.5. Módulo de conectividad Ethernet W5500 .....	14
3.2. Diseño de funcionamiento del sistema.....	15
3.3. Diseño de tarjeta de desarrollo e implementación de código. ....	18
Conclusiones .....	34

Recomendaciones.....	35
Bibliografía.....	36

## ÍNDICE DE FIGURAS

### Capítulo 2:

Figura 2. 1: Despliegue de aplicaciones del internet de las cosas (IoT). .....	5
Figura 2. 2: Diagrama del desarrollo de ciudades inteligentes (Smart Cities).6	
Figura 2. 3: Aplicación de IoT en agricultura de precisión mediante el uso de sensores. ....	8

### Capítulo 3:

Figura 3. 1: ESP8266.....	9
Figura 3. 2: Pin Planner. ....	10
Figura 3. 3: Microcontrolador ATMEGA 328p. ....	11
Figura 3. 4: Pin Planner Atmega 328p. ....	11
Figura 3. 5: Modulo conversor USB - TTL. ....	12
Figura 3. 6: Diagrama de bloques del sistema conversor USB - TTL .....	13
Figura 3. 7: Regulador de voltaje conmutada. ....	14
Figura 3. 8: Módulo Ethernet W5500. ....	14
Figura 3. 9: Pines y tamaño del módulo ethernet.....	15
Figura 3. 10: Diagrama de bloques del sistema propuesto.....	16
Figura 3. 11: Diagrama de flujo de programa de adquisición de datos y transmisión.....	17
Figura 3. 12: Diagrama de flujo de adquisición de datos desde ESP8266 y envío por Ethernet. ....	17
Figura 3. 13: Diagrama de flujo de adquisición de datos desde ESP8266 y envío por Ethernet. ....	18
Figura 3. 14: Nuevo proyecto en Proteus. ....	19
Figura 3. 15: Dimensiones para el módulo ESP8266. ....	20
Figura 3. 16: Dibujo realizado en Proteus para el módulo ESP8266. ....	20
Figura 3. 17: Creación de Package ESP8266 en Proteus. ....	21
Figura 3. 18: Creación del símbolo del módulo ESP8266.....	21
Figura 3. 19: Resultado obtenido a partir de la vinculación de símbolo y pcb del módulo. ....	22
Figura 3. 20: Esquemático del circuito para la tarjeta de desarrollo.....	23
Figura 3. 21: Administrador de diseño. ....	24

Figura 3. 22: Diseño PCB del prototipo.....	25
Figura 3. 23: Vista superior frontal 3D. ....	25
Figura 3. 24: Vista inferior 3D. ....	26
Figura 3. 25: Vista lateral 3D. ....	26
Figura 3. 26: Instalación de librería ESP8266.....	27
Figura 3. 27: Encabezado del código.....	27
Figura 3. 28: Función Setup.....	28
Figura 3. 29: Función principal del código. ....	29
Figura 3. 30: Sección de código para activación de leds. ....	29
Figura 3. 31: Diseño HTML de visualización.....	30
Figura 3. 32: Última sección del código. ....	30
Figura 3. 33: Código para envío de datos analógicos desde el microcontrolador ATMega al ESP8266. ....	31
Figura 3. 34: Encabezado del código del ESP8266.....	31
Figura 3. 35: Función setup y principal del ESP8266 .....	32

## Resumen

El presente documento presente el desarrollo del componente práctico del examen complejo, denominado "Diseño de una tarjeta de desarrollo con ESP8266 orientada a wireless y microcontroladores para IoT". En general, estamos entrando en una nueva era de tecnología informática, es decir, Internet de las cosas (IoT). IoT es una especie de "red neuronal global universal" en la nube que conecta varias cosas. IoT es un dispositivo y un sistema inteligentemente conectado que comprende máquinas inteligentes que interactúan y se comunican con otras máquinas, entornos, objetos e infraestructuras y la identificación por radiofrecuencia (RFID) y las tecnologías de red de sensores se enfrentarán para enfrentar este nuevo desafío. En el capítulo 1, se describe una introducción de IoT, objetivo general y específicos del componente práctico. En el capítulo 2, se describe brevemente la parte fundamental de IoT. Finalmente, en el capítulo 3, se muestra el propósito del presente trabajo que fue diseñar una tarjeta electrónica que utiliza el dispositivo ESP8266 para realizar la comunicación inalámbrica orientada al internet de las cosas.

**Palabras claves:** COMUNICACIONES, INALÁMBRICO, INTERNET, IOT, DISPOSITIVOS, SISTEMAS

## CAPÍTULO 1: DESCRIPCIÓN DEL COMPONENTE PRÁCTICO

### 1.1. Introducción.

En los últimos años se ha visto una explosión en los dispositivos de Internet de las cosas (*Internet of Things, IoT*) y en productos conectados como sensores inalámbricos, medidores inteligentes, sistemas de automatización del hogar (domótica) y dispositivos portátiles (wearables) (Corporación Ruta N, 2015). El costo reducido de componentes como sensores y procesadores junto con el aumento de la conectividad inalámbrica ha dado lugar a que muchos productos se hagan "inteligentes" y sean capaces de comunicarse entre sí sin intervención humana. (González García, 2017)

Los productos exitosos deben cumplir a menudo los requisitos de la competencia, incluyendo bajo consumo de energía, largos rangos de conectividad inalámbrica y mayor potencia de procesamiento. Este documento explora cómo los microcontroladores inalámbricos basados en MicroChip, Atmel y ARM pueden satisfacer los requisitos más exigentes de los sistemas IoT.

La topología de red de los sistemas IoT se compone de nodos simples que recopilan y transmiten una cantidad limitada de datos a un controlador central o puerta de enlace que proporciona conectividad a Internet y servicios en la nube. Los nodos y las puertas de enlace deben estar diseñados para minimizar el consumo de energía, proporcionar conexiones de red confiables y robustas y extender el rango de conectividad inalámbrica tanto como sea posible.

En el corazón de los sistemas IoT hay una unidad de procesador o microcontrolador (uC) que procesa datos y ejecuta pilas de software conectadas a un dispositivo inalámbrico para la conectividad. Los requisitos tanto para uCs como para el dispositivo inalámbrico son específicos de la aplicación final y los requisitos del sistema. Los nodos de sensor de IoT avanzados consolidan las funciones del sensor y usan uCs de 8 bits o un dispositivo de 32 bits para ejecutar una pequeña pila de protocolos de

radiofrecuencia (RF). Estos dispositivos generalmente funcionan con baterías y se conectan a puertas de enlace donde se produce un procesamiento y una transmisión de datos más pesados.

### **1.2. Objetivo General.**

Diseñar una tarjeta de desarrollo con ESP8266 orientada a comunicación inalámbrica a través de sistemas microcontroladores aplicado al internet de las cosas (IoT).

### **1.3. Objetivos Específicos.**

- a. Describir brevemente el marco teórico de los fundamentos de redes inalámbricas y su uso en el internet de las cosas.
- b. Diseñar el sistema de desarrollo con ESP8266 orientada a wireless e IoT utilizando el microcontrolador ATmega 328<sup>a</sup>.
- c. Implementar la tarjeta de desarrollo de aplicaciones en el internet de las cosas.

## **CAPÍTULO 2: Fundamentación teórica.**

### **2.1. Introducción de IoT.**

El término Internet de las cosas (IoT) representa un concepto general para la capacidad de los dispositivos de red de detectar y recopilar datos de todo el mundo, y luego compartir esos datos a través de Internet, donde pueden procesarse y utilizarse para diversos fines interesantes. El IoT se compone de máquinas inteligentes que interactúan y se comunican con otras máquinas, objetos, entornos e infraestructuras. Hoy en día, todas las personas están conectadas entre sí mediante muchas formas de comunicación. Donde la forma de comunicación más popular es internet, en otras palabras, se puede decir que internet conecta a las personas. La idea esencial del IoT ha existido durante casi dos décadas, y ha atraído a muchos investigadores e industrias debido a su gran impacto estimado en la mejora de nuestra vida cotidiana y la sociedad. (Romero García, 2017)

Cuando cosas como los electrodomésticos están conectados a una red, pueden trabajar juntos en cooperación para proporcionar el servicio ideal en su conjunto, no como una colección de dispositivos que funcionan de forma independiente. Esto es útil para muchas de las aplicaciones y servicios del mundo real, y uno lo aplicaría, por ejemplo, para construir una residencia inteligente; Las ventanas se pueden cerrar automáticamente cuando se enciende el aire acondicionado, o se pueden abrir para recibir oxígeno cuando se enciende el horno de gas.

La idea de IoT es especialmente valiosa para las personas con discapacidad, ya que las tecnologías de IoT pueden apoyar actividades humanas a mayor escala, como la construcción o la sociedad, ya que los dispositivos pueden cooperar mutuamente para actuar como un sistema total. La capacidad de comunicación y el control manual remoto conducen al siguiente paso ... ¿cómo automatizo las cosas y, según la configuración y con un sofisticado procesamiento basado en la nube, hacer que las cosas sucedan sin mi intervención? Ese es el objetivo final de algunas aplicaciones de IoT. Y, para que esas aplicaciones se conecten y aprovechen Internet para lograr

este objetivo, primero deben volverse "inteligentes" (incorporar un uC/procesador integrado con una ID única asociada) y luego conectarse y, finalmente, controlarse. Esas capacidades pueden habilitar una nueva clase de servicios que facilita la vida de sus usuarios. (Biswas & Giaffreda, 2014)

## 2.2. Aplicaciones del IoT.

Existen diversas aplicaciones de IoT (véase la figura 2.1), por ejemplo, en centros comerciales se utilizarían para controlar la humedad y la temperatura del centro comercial a través de centrales de aire acondicionado mediante el uso de un sensor de temperatura. Aunque, también se puede utilizar en varias organizaciones, como el sistema de tabloneros de anuncios educativos o en la estación de ferrocarril, parada de autobús y aeropuertos para mostrar la información y la notificación. En la organización industrial también se pueden utilizar. El sistema de pantalla electrónica se puede usar para mostrar mensajes de emergencia en hospitales. Algunas áreas donde IoT se usa con frecuencia.



Figura 2. 1: Despliegue de aplicaciones del internet de las cosas (IoT).  
Fuente: (IoTNet, 2019)

### 2.2.1. Ciudades inteligentes

Hacer que la ciudad sea una ciudad inteligente para interactuar con el escape de datos producido en la ciudad y sus alrededores. A continuación, se describen brevemente las diferentes opciones que se tienen en una ciudad inteligente o smart city:

- Monitoreo la disponibilidad de estacionamiento en la ciudad.
- Monitoreo vibraciones y condiciones materiales en edificios, puentes y monumentos históricos.
- Detección de dispositivos Android, iPhone y, en general, cualquier dispositivo que funcione con interfaces inalámbricas como Bluetooth o WiFi.
- Medición de la energía radiada por las estaciones celulares y enrutadores Wi-Fi.
- Monitoreo de vehículos y niveles de peatones para optimizar las rutas de conducción y caminata.
- Detección de niveles de basura en contenedores para optimizar las rutas de recolección de basura
- Autopistas inteligentes con mensajes de advertencia y desvíos de acuerdo con las condiciones climáticas y eventos inesperados como accidentes o atascos.

La figura 2.2 muestra un diagrama general del despliegue de ciudades inteligentes.



Figura 2. 2: Diagrama del desarrollo de ciudades inteligentes (Smart Cities).

Fuente: (Hammi, Khatoun, Zeadally, Fayad, & Khoukhi, 2018)

### **2.2.2. Seguridad y emergencias.**

- Control de acceso perimetral: detección y control de personas no autorizadas y restringidas.
- Presencia de líquidos: detección de líquidos en centros de datos, terrenos de edificios sensibles y almacenes para evitar averías y corrosión.
- Niveles de radiación: en las estaciones de energía nuclear, los alrededores distribuyeron la medición de los niveles de radiación para generar alertas de fugas.
- Gases explosivos y peligrosos: detección de fugas y niveles de gas en entornos industriales, alrededores de fábricas químicas y minas interiores

### **2.2.3. Agricultura inteligente**

A continuación, se indican las aplicaciones que se desarrollan en la actualidad en sistemas de agrarios inteligentes.

- Mejora de la calidad del vino: control de la humedad del suelo y el diámetro del tronco en los viñedos para controlar la cantidad de azúcar en las uvas y la salud de la vid.
- Invernaderos: controle las condiciones de microclima para maximizar la producción de frutas y verduras y su calidad.
- Campos de golf: riego selectivo en zonas secas para reducir los recursos hídricos necesarios en el green.
- Red de estaciones meteorológicas: estudio de las condiciones climáticas en los campos para pronosticar la formación de hielo, lluvia, sequía, nieve o cambios de viento.
- Compost: control de los niveles de humedad y temperatura en alfalfa, heno, paja, etc. para prevenir hongos y otros contaminantes microbianos.

La figura 2.3 muestra una gráfica obtenida de un sistema de sensores en agricultura de precisión sobre IoT.

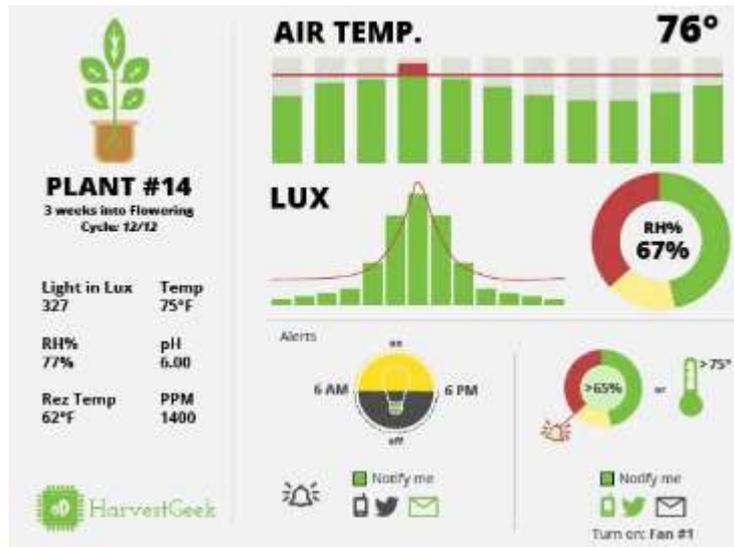


Figura 2. 3: Aplicación de IoT en agricultura de precisión mediante el uso de sensores.

Fuente: (Jayaraman, Yavari, Georgakopoulos, Morshed, & Zaslavsky, 2016)

## **Capítulo 3: Diseño de una tarjeta de desarrollo con ESP8266 orientada a Wireless y microcontroladores para IoT.**

Actualmente el mundo se encuentra en una década de innovación tecnológica acelerada, el cual se puede observar cómo es posible interconectar muchas de las cosas que se encuentran en nuestros hogares facilitando la vida en cierto modo. Este proyecto permitirá al lector poder realizar mediante una tarjeta de desarrollo, pruebas de conectividad y procesamiento de señales de sensores hacia el internet o redes locales. En sí, de que se trata el IoT, no es más que la interconexión y la relación de los objetos que se utilizan en la vida cotidiana con internet, esto permite a los usuarios poder tener un control casi total sobre las cosas y conocer estados de estos.

### **3.1. Elementos utilizados para la implementación de la tarjeta de desarrollo con ESP8266.**

En esta sección del capítulo se detalla los elementos a utilizar y una breve explicación de estos para el diseño de la tarjeta de desarrollo con dispositivo ESP8266.

#### **3.1.1. Dispositivo ESP8266**

Este dispositivo no es más que un chip WiFi (véase la figura 3.1), que trabaja con el protocolo TCP/IP y de muy bajo costo, además, contiene varias entradas y salidas de propósito múltiple, permitiendo tener conectividad hacia la red y procesamiento de estados.



Figura 3. 1: ESP8266.  
Elaborado por: Autor.

Este chip contiene un microcontrolador Tensilica Xtensa LX106 que tiene muchas características superiores con respecto a los competidores como microchip o Atmel, para este caso el microcontrolador es uno de 32 bits con arquitectura RISC a 80 MHz, con una memoria flash de 512KB. Las temperaturas de operación oscilan entre -40°C a 125°C, posee una entrada de conversión analógica/digital (*Analog to Digital Converter, ADC*) de 10 bits y su voltaje de operación oscila entre 2.5 V a 3.6 V, siendo este uno de los mejores candidatos para la utilización en el internet de las cosas. En la figura 3.2 se muestra el pin planner del microcontrolador.

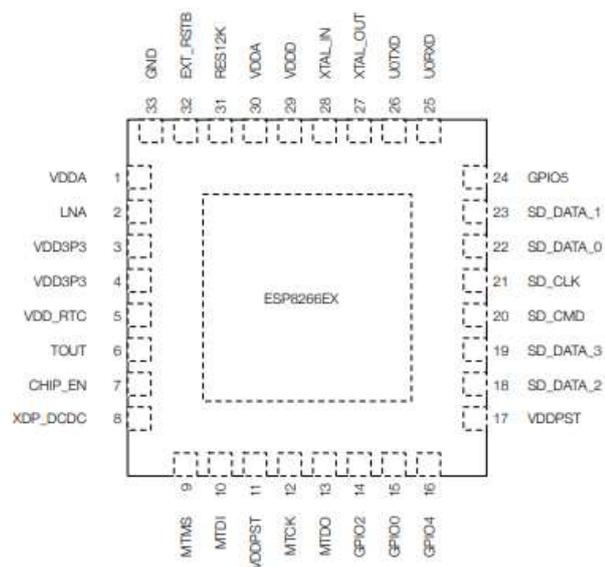


Figura 3. 2: Pin Planner.  
Elaborado por: Autor.

### 3.1.2. Microcontrolador Atmel 328p.

El microcontrolador es de 8bits de la compañía ATMEL, está basado en arquitectura RISC de muy alto rendimiento, tiene una memoria SRAM 2KB, además de 23 puertos de entrada y salida de propósito general, 32 registros USART programable en serie. Este es capaz de ejecutar instrucciones en un ciclo de reloj permitiendo llegar a rendimientos que se aproximan a 1MIPS por MHz. Este dispositivo está diseñado para aplicaciones industriales y en viviendas o edificios, en la figura 3.3 se muestra el microcontrolador.

El microcontrolador es muy utilizado ya que este fue utilizado en el proyecto de open hardware Arduino, siendo popularizado para

implementación de proyectos estudiantiles y de investigación consiguiendo el desarrollo en el área del internet de las cosas (*Internet of Things, IoT*).



Figura 3. 3: Microcontrolador ATMEGA 328p.  
Elaborado por: Autor.

Este microcontrolador contiene un núcleo AVR, su frecuencia de trabajo máxima es de 20 MHz, y su espacio de memoria es de 32 KB, además es posible trabajar con una fuente de alimentación que oscila entre 1.8 V y 5.5V, y cuenta también con un conversor análogo digital (Analog/Digital Converter, ADC) de 10 bits. En la figura 3.4 se puede observar el pin planner de este que entrega el fabricante.

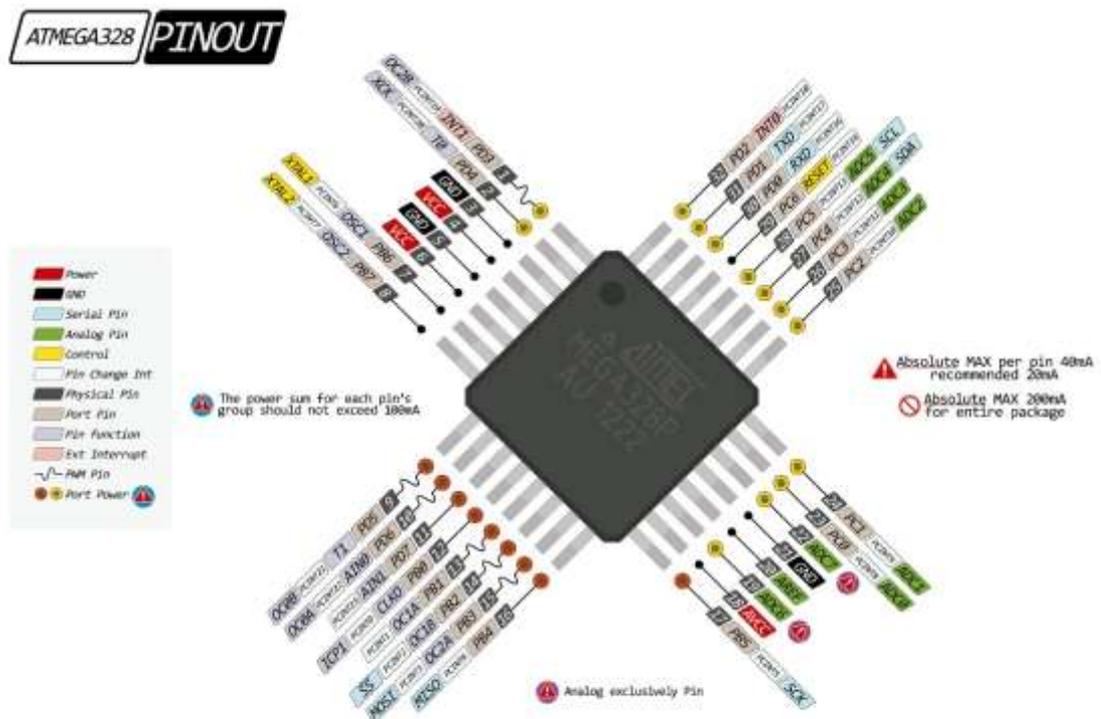


Figura 3. 4: Pin Planner Atmega 328p.  
Elaborado por: Autor.

Entre otras características importantes de este microcontrolador está la de poseer puertos para conectividad SPI y un puerto UART, un comparador analógico, la temperatura de operación oscila entre  $-40^{\circ}\text{C}$  a  $85^{\circ}\text{C}$  y por último un chip multiplicador de 2 tiempos.

### 3.1.3. Módulo USB-TTL

Este módulo permite la conversión del puerto USB del computador en un puerto con niveles TTL compatible con la mayoría de los microcontroladores que existen en el mercado actualmente, gracias a esto es posible la escritura de código de programación en un dispositivo que previamente tenga cargado un bootloader. La figura 3.5 muestra el módulo conversor, este posee cinco pines los cuales corresponden a 3.3 V, Tx, Rx, GND y 5 V respectivamente permitiendo conectar directamente el microcontrolador sin necesidad de una fuente externa.



Figura 3. 5: Módulo conversor USB - TTL.  
Elaborado por: Autor.

Este módulo posee en su interior un controlador PL2303, que realmente funciona como un puente entre el puerto USB y un puerto RS232 estándar, este dispositivo también cumple con la administración adecuada de la energía proveniente del puerto usb, proporcionando la corriente necesaria a un microcontrolador para ser grabado adecuadamente.

En la hoja de datos del fabricante se proporciona el diagrama de bloques de este dispositivo (véase la figura 3.6) en este se puede observar un transceiver USB que va directamente a un USB SIE, y paralelamente trabajan en conjunto la unidad de control y un registro. Posteriormente se emplea un

convertor RS232, y paralelamente igualmente que en el anterior caso es oscilador que controla la velocidad de reloj del sistema y la memoria EEPROM comunicado por el puerto SPI.

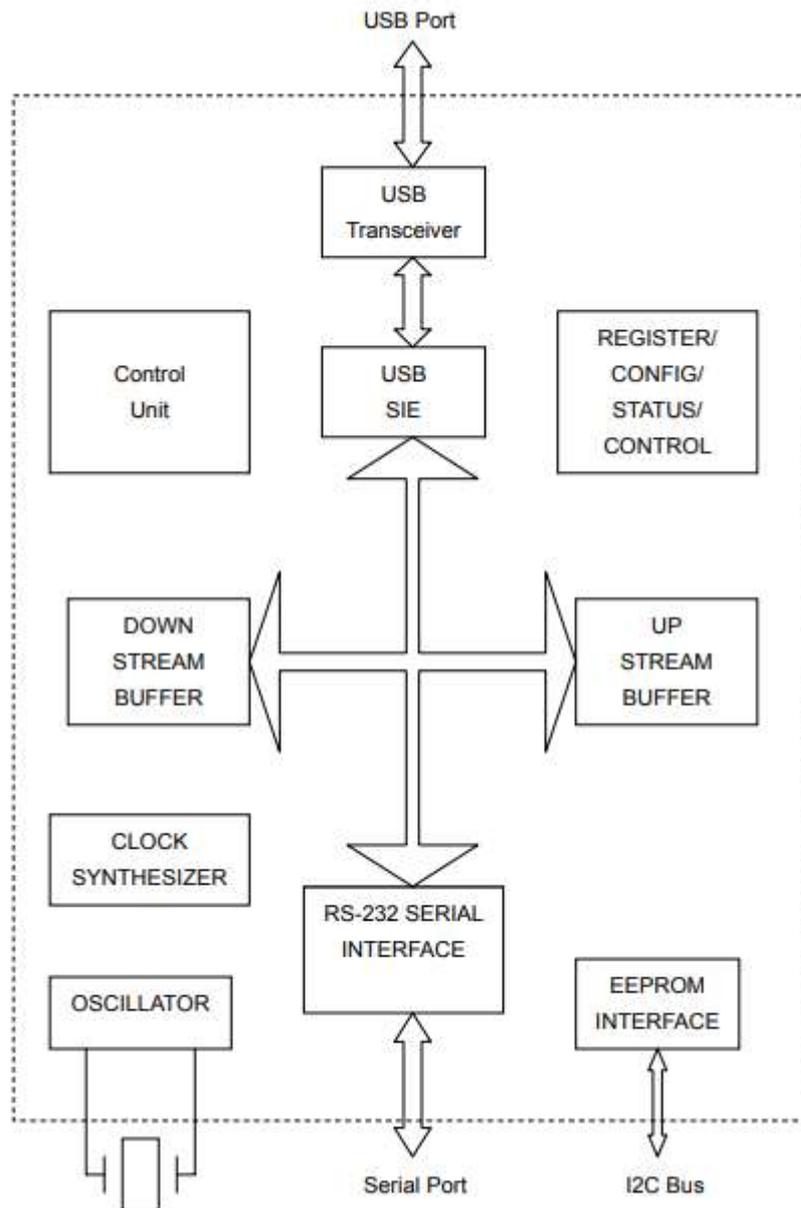


Figura 3. 6: Diagrama de bloques del sistema convertidor USB - TTL  
Elaborado por: Autor.

### 3.1.4. Módulo Step Down 1.5Amp

Este es un módulo de regulación de voltaje conmutada, permitiendo un rendimiento energético muy estable y así evitando las pérdidas de energía en calentamiento u otras razones, estas son más complicadas y elaboradas con respecto a su funcionamiento, pero es posible conseguirlos ya contruidos en el mercado nacional, como se muestra en la figura 3.7.



Figura 3. 7: Regulador de voltaje conmutada.  
Elaborado por: Autor.

Este módulo permite obtener un potencial regulado a partir de una fuente de alimentación mayor, por ejemplo, se tiene una fuente de 24v DC se puede regular a los 2.5 V, 3.3 V, 5 V, 9 V, 12 V o puede ser establecido manualmente gracias a un potenciómetro que se tiene presente en este dispositivo. Este es un circuito monolítico adecuado para el diseño fácil y conveniente de una fuente tipo Buck, es capaz de reducir corriente de hasta 1.5 A, además maneja una carga con excelente regulación de línea y bajo voltaje rizado.

### 3.1.5. Módulo de conectividad Ethernet W5500

Este módulo es un dispositivo que permite la conectividad de un microcontrolador a una red ethernet mediante el puerto RJ45 integrado en este, compatible con protocolos TCP/IP, UDP, ICMP, IPv4, ARP, IGMP, PPPoE, además posee una memoria interna de 32 KBytes para Tx/Rx, tal como se muestra en la figura 3.8.



Figura 3. 8: Módulo Ethernet W5500.  
Elaborado por: Autor.

Este dispositivo además de las características redactadas anteriormente se puede apreciar que soporta interfaz serie de alta velocidad periférica (SPI MODE0, 3), soporta negociación automática (Full y medio dúplex), salidas indicatoras led y funciona con un potencial de 3.3V con tolerancia de señal de entradas y salidas de 5v.

Entre las características físicas, se tiene que el módulo tiene un tamaño de 23mm x 25mm, una frecuencia de reloj SPI max 80MHz, y su peso es de apenas 10 gramos. Estos dispositivos son ideales para conectar un microcontrolador a la red ethernet y utilizar varios de los protocolos de internet de las cosas. En la figura 3.9 se presenta los pines utilizados para la comunicación SPI y alimentación además de sus dimensiones descritas anteriormente, gracias a esto es posible diseñar un puerto en la tarjeta de desarrollo que encaje con la descripción de este.

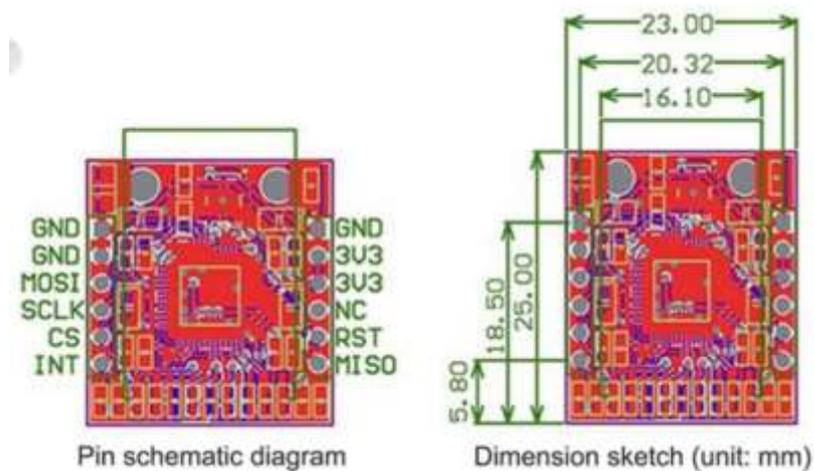


Figura 3. 9: Pines y tamaño del módulo ethernet.

Elaborado por: Autor.

### 3.2. Diseño de funcionamiento del sistema.

Este sistema propuesto es para el uso de microcontroladores en la investigación y uso del IoT, para ello se utilizará dos microcontroladores, el ESP8266 y el Atmega328p, ambos cargados con el bootloader de arduino permitiendo un entorno libre para el desarrollo de este. Para ello se muestra el diagrama de bloques que se puede ver en la figura 3.10, el funcionamiento de este, el cual permitirá utilizar varios sensores que son de libre acceso en el mercado nacional.

El ESP8266 tendrá la capacidad de comunicar y procesar datos procedentes de las entradas y salidas digitales utilizando los distintos protocolos de IoT existentes como son: mqtt, amqp, wamp, stomp, entre otros. Las salidas digitales están representadas con 5 leds de 5 mm, y las entradas

digitales están representadas en el circuito con un Dip switch de 5 pines, el cual simulara cualquier valor lógico alto o bajo.

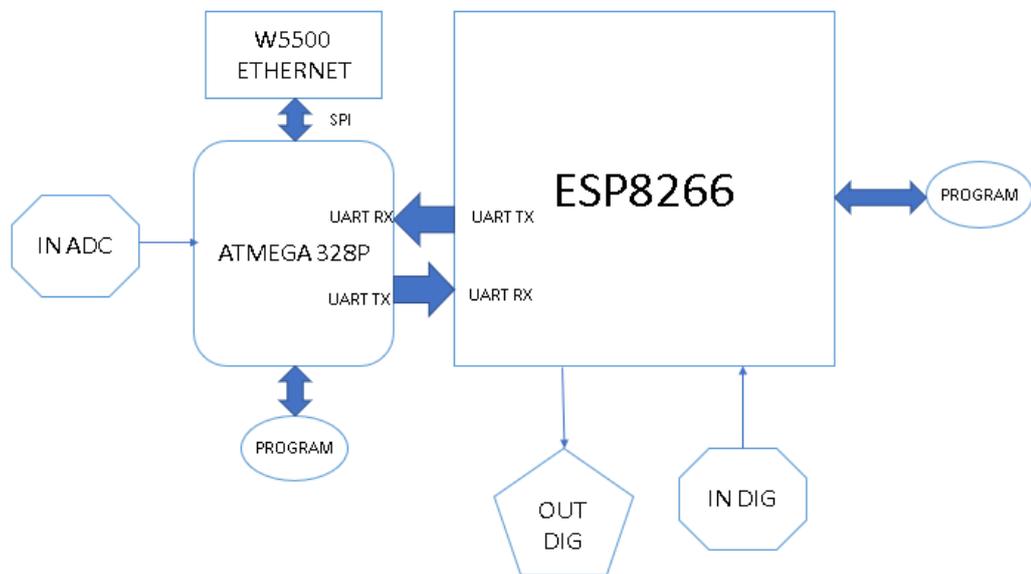


Figura 3. 10: Diagrama de bloques del sistema propuesto.  
Elaborado por: Autor.

El microcontrolador Atmega será la responsable de adicionar más puertos analógicos al sistema, este se comunicará por medio del puerto UART con el ESP8266, para que este pueda encapsular y administrar los datos recibidos, también podrá comunicarse vía ethernet por medio del módulo W5500 conector por el puerto Spi de este. Con esto se da una amplia gama de aplicación de los dos microcontroladores, estos tendrán su puerto UARTs disponible también para permitir al usuario final la programación con el módulo USB-TTL.

La placa de diseño de sistemas IoT está orientada además para el desarrollo de comunicaciones de propósito general no solo de manera inalámbrica, gracias al puerto UART y SPI del microcontrolador Atmega 328p se podrá experimentar diferentes resoluciones. El sistema está orientado para un uso continuo en fase de producción ya que se encuentra protegido con los módulos de regulación de voltaje por conmutación, estos son más estables y permiten la selección de varios tipos de potencias desde una misma fuente. En la figura 3.11 se presenta el diagrama de flujo que funcionara en el

microcontrolador ATmega 328p, este lee los datos de los puertos analógicos y los encapsula para ser enviados por el puerto UARTs al ESP8266.

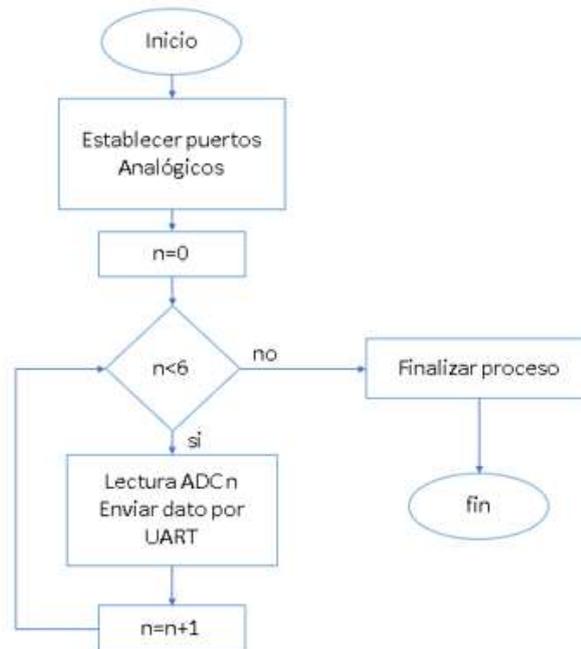


Figura 3. 11: Diagrama de flujo de programa de adquisición de datos y transmisión.  
Elaborado por: Autor.

Además, el microcontrolador tendrá la capacidad de adquirir datos digitales desde el ESP8266 para transmitirlos por cable Ethernet, en la figura 3.12 se demuestra dicho diseño.

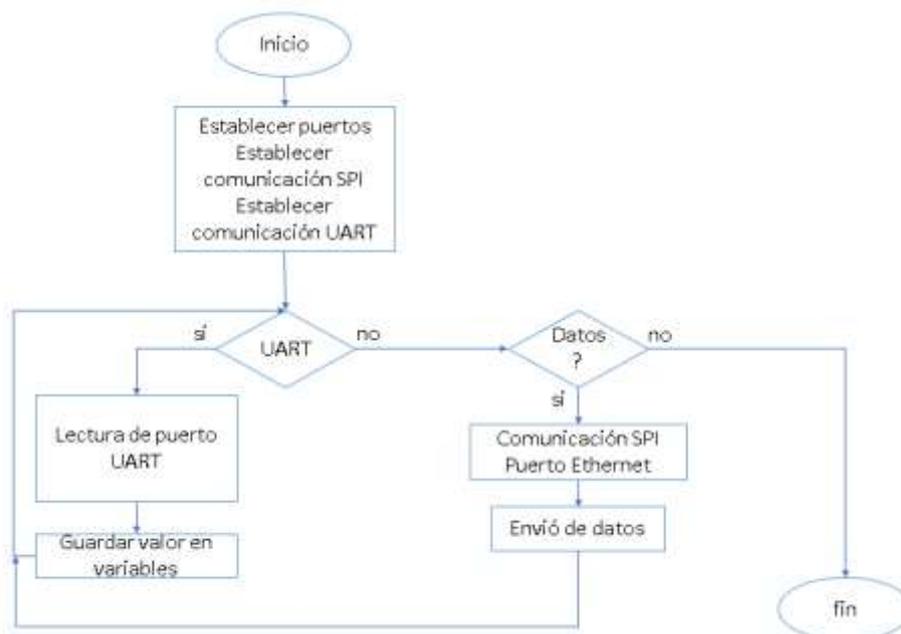


Figura 3. 12: Diagrama de flujo de adquisición de datos desde ESP8266 y envío por Ethernet.  
Elaborado por: Autor.

El diagrama anterior describe el funcionamiento del microcontrolador en modo de transmisor de datos por ethernet, este obtendrá las tramas desde el ESP8266 por el puerto UART donde gracias al dipswitch que posee la placa se puede simular datos digitales de entrada. El microcontrolador encapsulará los datos en tramas según el protocolo IoT elegido, donde podrá comunicar estos a un servidor local o en la nube. La figura 3.13 muestra el diagrama de flujo el funcionamiento del ESP8266.

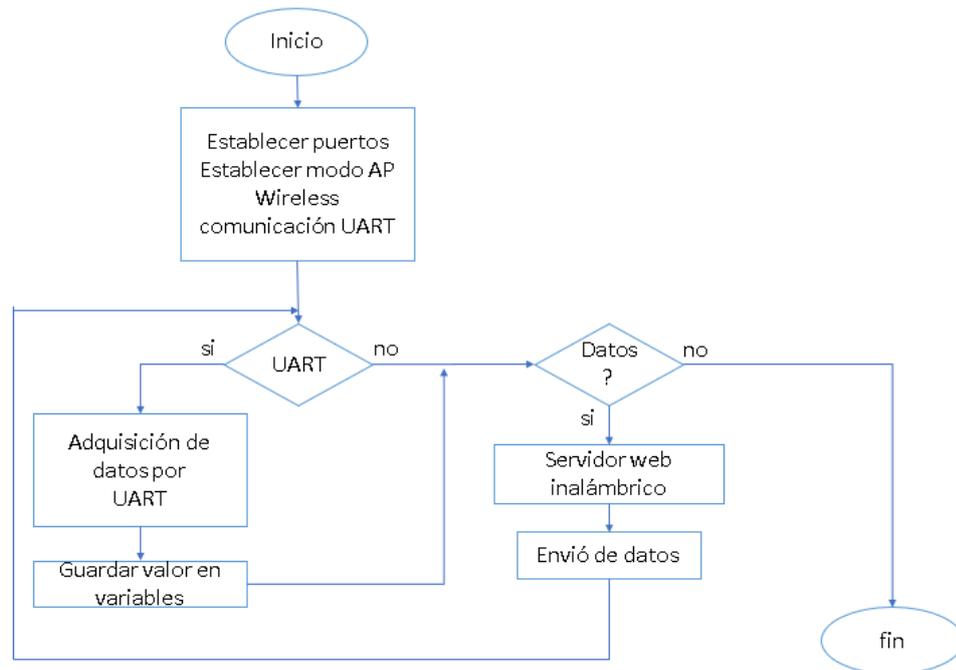


Figura 3. 13: Diagrama de flujo de adquisición de datos desde ESP8266 y envío por Ethernet.

Elaborado por: Autor.

En la figura 3.13 se puede apreciar el diagrama de flujo diseñado para funcionar como un servidor web, este tiene como objetivo adquirir datos de los puertos analógicos del microcontrolador ATmega mediante el puerto de transmisión y recepción UART, estos datos entran al ESP8266 empaquetados, posteriormente se interroga si existen para enviarlos hacia el exterior a un servidor IoT en la nube o en una red local.

### 3.3. Diseño de tarjeta de desarrollo e implementación de código.

En esta sección del capítulo se presenta al lector el proceso realizado para el diseño de la tarjeta de desarrollo basado en ESP8266 para la utilización de investigadores o estudiantes en el avance de las tecnologías

IoT, para ello se utilizó el programa Proteus en su versión 8.7 SP3. Previamente al diseño de la pcb, es necesario el desarrollo de un esquemático que pueda seguir las reglas según el diagrama de bloques previamente descrito en este documento, es necesario también crear los componentes que no se encuentren en las librerías que suministra el software.

En la figura 3.14 se presenta en primera instancia como el procedimiento para iniciar un proyecto en Proteus, para ello en la pantalla principal se seleccionen el menú file, nuevo proyecto, a continuación, nos solicita información acerca de este.

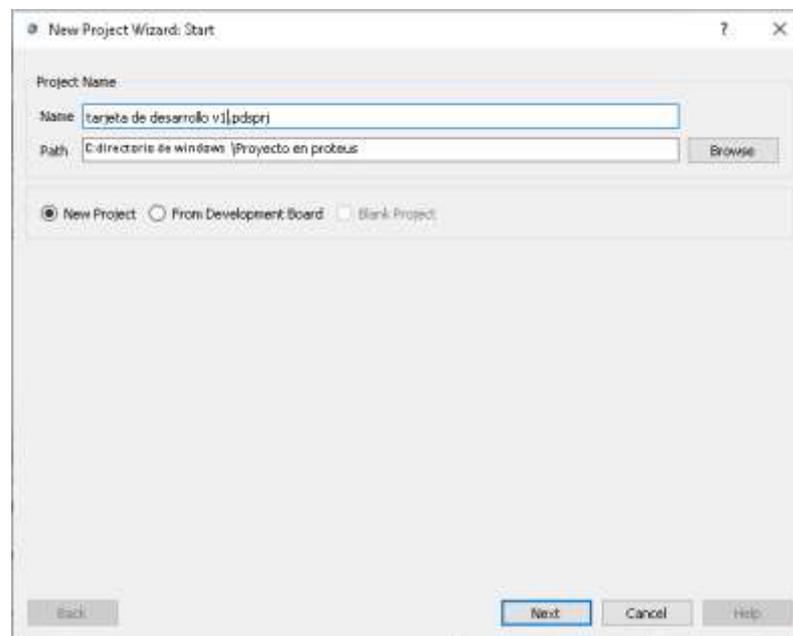


Figura 3. 14: Nuevo proyecto en Proteus.  
Elaborado por: Autor.

Cuando es posible tener una mesa de trabajo en el software Proteus es necesario crear los módulos que no se encuentran en las librerías nativas de este, para ello es necesario abrir el apartado de diseño pcb y con ayuda de la hoja de datos del ESP8266 se dibuja manualmente los Pads y el contorno, en la figura 3.15 se presenta los datos del Datasheet del módulo. Como se puede apreciar, los Pads cuentan con una dimensión de 2.5mm x 1mm, además el tamaño total del módulo incluyendo la antena incorporada es de 16mm x 24mm, en la figura 3.16 se puede apreciar el resultado obtenido al dibujarlo en Proteus.

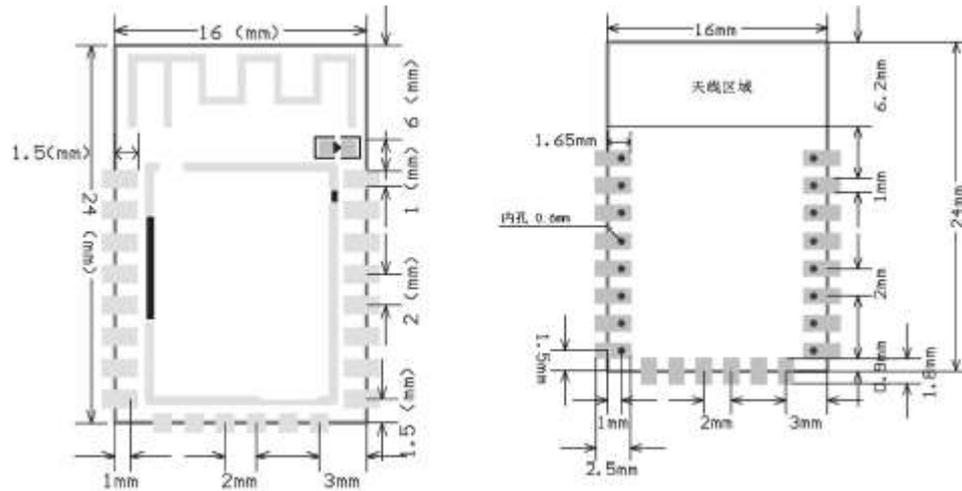


Figura 3. 15: Dimensiones para el módulo ESP8266.  
Elaborado por: Autor.

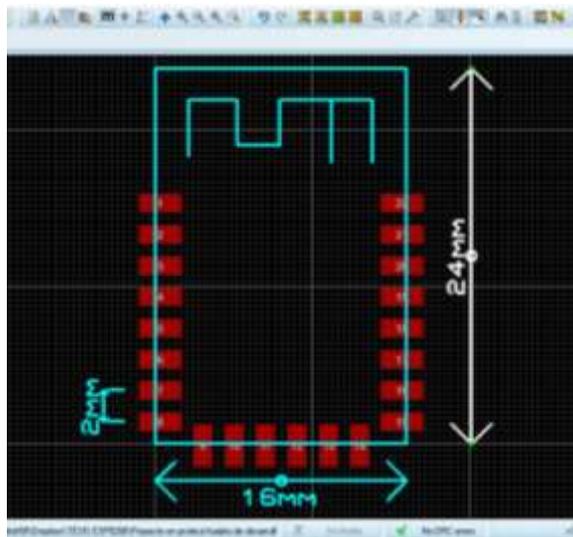


Figura 3. 16: Dibujo realizado en Proteus para el módulo ESP8266.  
Elaborado por: Autor.

A continuación, se procede a crear el Package y guardarlo en la librería del usuario del software Proteus, en la imagen 3.17 se muestra dicho procedimiento. En la ventana de creación del módulo se describen las características de este módulo, así como su aspecto en 3D para visualización posterior al diseño de la PCB.

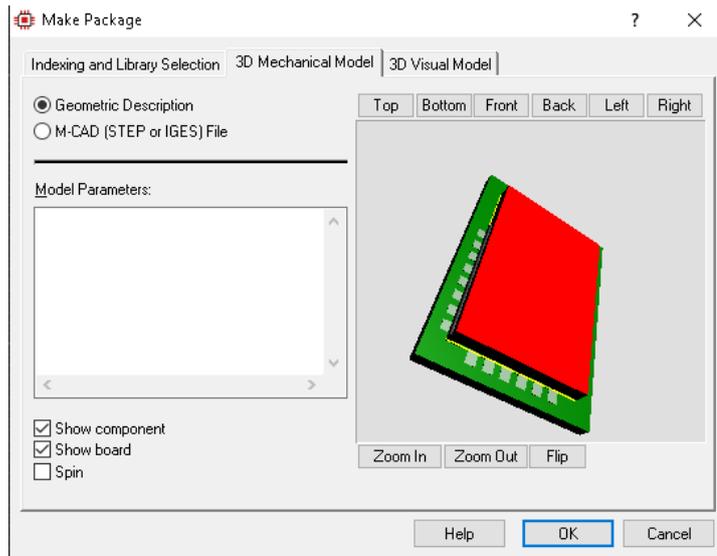


Figura 3. 17: Creación de Package ESP8266 en Proteus.  
Elaborado por: Autor.

Posteriormente a este procedimiento descrito, es necesario construir el símbolo en la sección de esquemático, este será la representación física del módulo para el diseño y utilización de manera virtual, este debe tener los pines correspondientes al circuito físico: La figura 3.18 se presenta el resultado de dicho diseño.

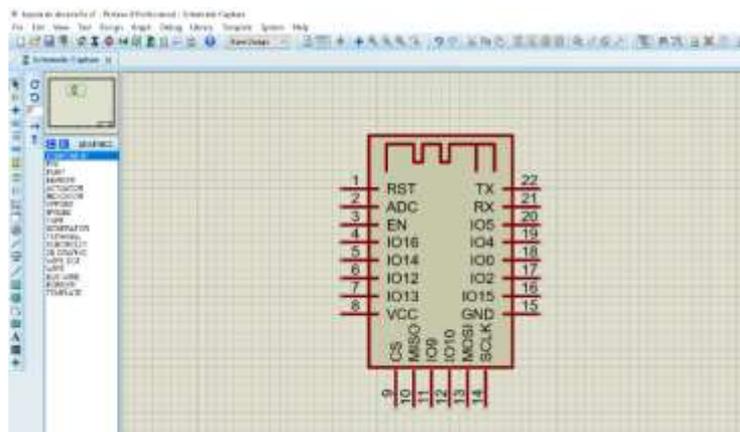


Figura 3. 18: Creación del símbolo del módulo ESP8266.  
Elaborado por: Autor.

Con ello ya construido el símbolo y el símbolo para el pcb, se procede a la vinculación de ambas para que al momento de seleccionar este artículo se pueda previsualizarlo como es debido, en la figura 3.19 se presenta el resultado obtenido de la creación de este módulo y su vinculación de ambas partes.

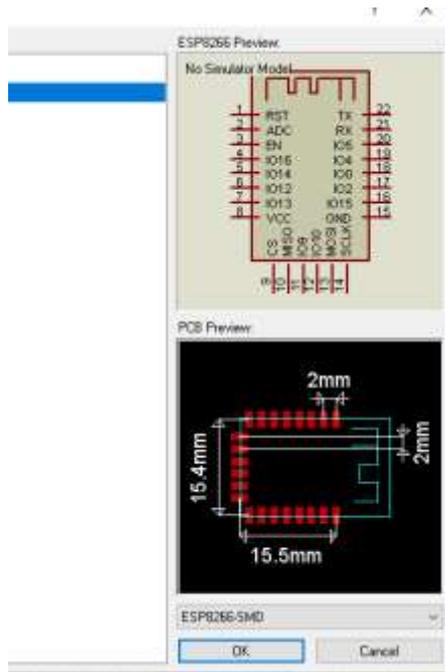


Figura 3. 19: Resultado obtenido a partir de la vinculación de símbolo y pcb del módulo.

Elaborado por: Autor.

El mismo procedimiento se realiza para la creación de los módulos reguladores de voltaje o step down, y el módulo W5500 que permite la comunicación del microcontrolador por cable Ethernet hacia el internet o a una red local utilizando protocolos conocidos para IoT.

A continuación, se presenta en la figura 3.20 el esquemático de todo el circuito desarrollado con el fin implementar para el área de estudio los protocolos de IoT. Se tiene dos reguladores de voltaje, uno a 5vdc y el otro a 3.3vdc ya que el microcontrolador y el módulo ESP8266 trabajan a estos voltajes respectivamente, además, se tiene las entradas analógicas y el W5500 conectadas por etiquetas en las vías al microcontrolador, también se cuenta con las entradas y salidas digitales al ESP8266, y ambos cuentan con los puertos necesarios para la programación por el puerto UART utilizando el módulo USB-TTL anteriormente descrito.

Una vez enrutadas todas las vías con sus respectivos nombres como se aprecia en la figura 3.20, se procede a la creación del diseño electrónico de la tarjeta pcb, este debe tener ciertas características especiales para que sea



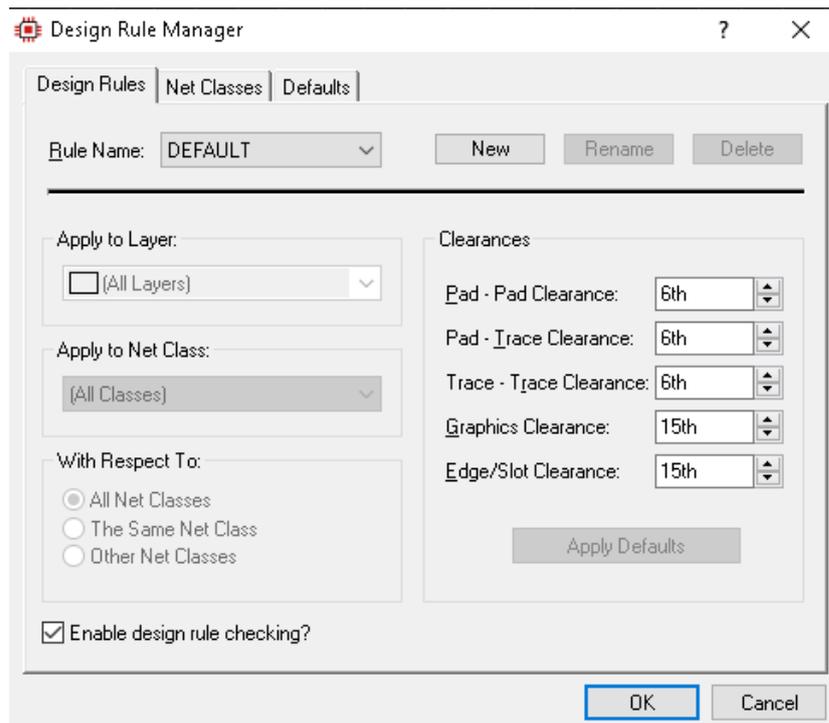


Figura 3. 21: Administrador de diseño.  
Elaborado por: Autor.

Después de concluir con las reglas del ruteo en la pcb se procede a ordenar los elementos y conectar las vías de cada uno de ellos, con ciertas horas de trabajo se logra tener un diseño profesional de este como se aprecia en la figura 3.22, se llega a utilizar dos capas, el botón copper y top copper, ambas tienen colores diferentes para ser reconocidas como es el azul y rojo respectivamente. Esta tarjeta de desarrollo tiene unas medidas de 141mm x 72mm, además cuenta con dos arquitecturas, montaje superficial y de orificios, entre los elementos de su interior más importantes, se cuenta con un módulo ethernet el cual se llamó WZ1, el módulo ESP8266 el cual se llama ESP1, el microcontrolador U1 y los reguladores de voltajes REG1 y REG2, los demás elementos son diodos leds, capacitores, resistencias y un cristal.

Si se desea obtener una vista más precisa al final del diseño, tomando en cuenta las dimensiones de los elementos y haciendo un pronóstico de como llegarán a encajar uno a uno en la placa pcb, Proteus posee una utilidad que permite la previsualización en 3D de este, en las figuras 3.23, 3.24, 3.25, se presenta dicho diseño en sus vistas frontal superior, inferior, lateral respectivamente.

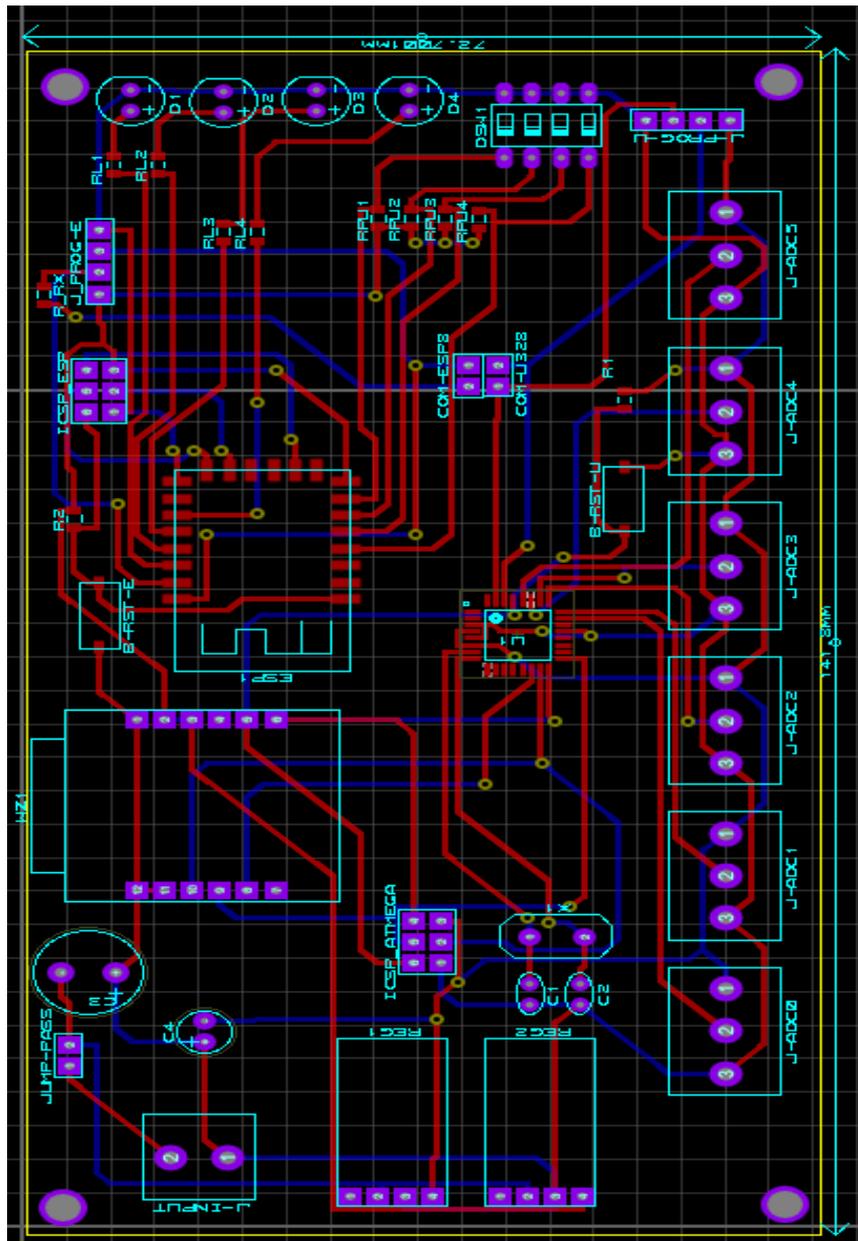


Figura 3. 22: Diseño PCB del prototipo.  
Elaborado por: Autor.

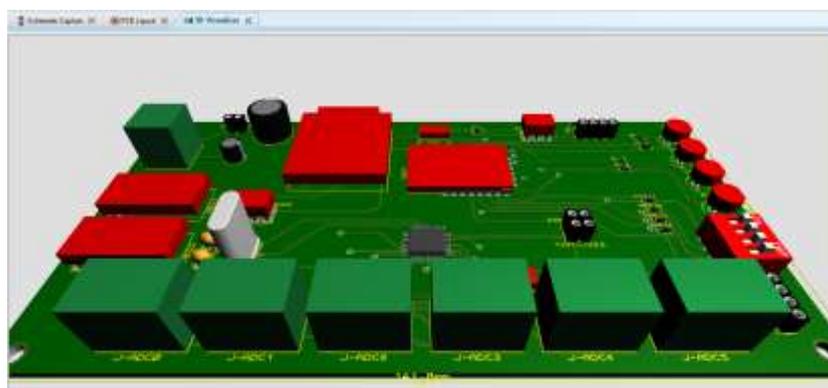


Figura 3. 23: Vista superior frontal 3D.  
Elaborado por: Autor.

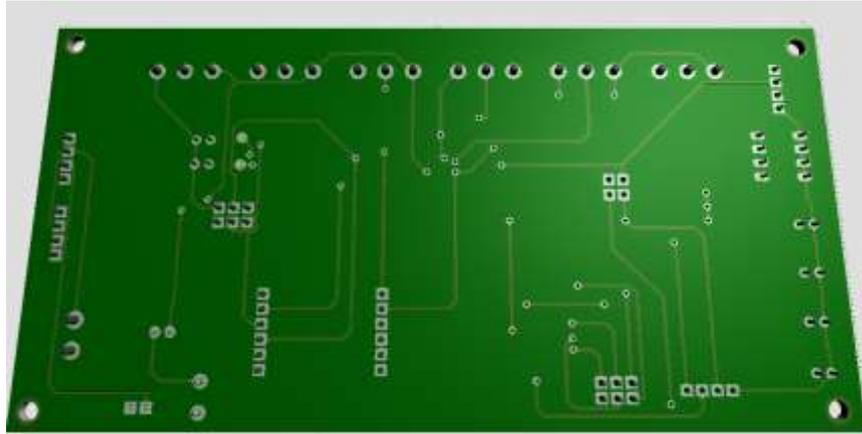


Figura 3. 24: Vista inferior 3D.  
Elaborado por: Autor.

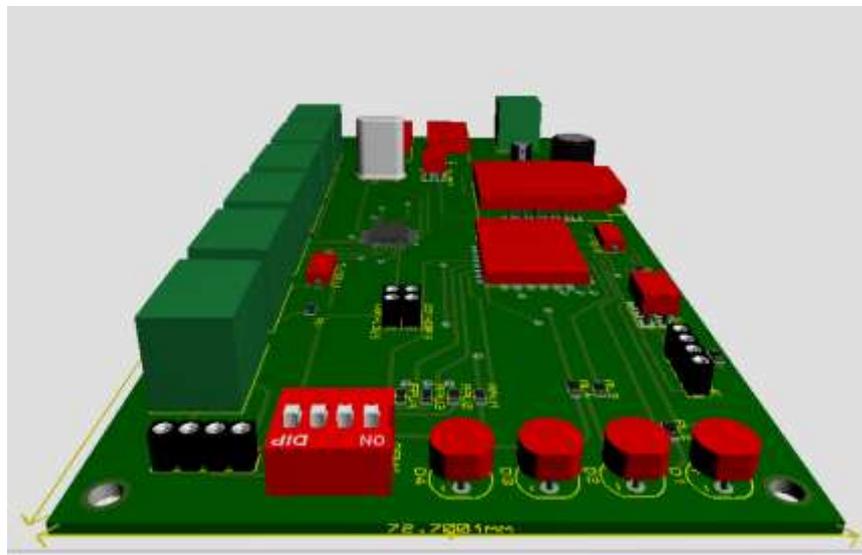


Figura 3. 25: Vista lateral 3D.  
Elaborado por: Autor.

Cuando la tarjeta de desarrollo se encuentre en producción, es necesario la construcción del código basándose en los diagramas de flujos previamente descritos, este será cargada a los distintos microcontroladores que en esta se encuentran, para ello se utilizara el Ide de Arduino, y para iniciar se requiere cargar las librerías correspondientes. En la figura 3.26 se muestra la instalación de la librería para el ESP8266.

Una vez realizado este procedimiento, se construye el código requerido para un servidor web en el módulo ESP8266, en la figura 3.27 se muestra el encabezado en el que se llama la librería instalada, y se definen variables y puertos para la salida.



Figura 3. 26: Instalación de librería ESP8266.

Elaborado por: Autor.

```
// carga de libreria wifi ESP8266
#include <ESP8266WiFi.h>

// Reemplazar por la red wifi a conectar
const char* ssid      = "REEMPLAZAR NOMBRE DE SSID";
const char* password = "REEMPLAZAR POR PASSWORD DE RED WIFI";

// Configuramos el puerto del servidor 80
WiFiServer server(80);

// Variable para HTTP request
String header;

// Variables auxiliares para salidas digitales de muestra
String output5State = "off";
String output4State = "off";

// Asignación de pines de salida a puertos del módulo ESP
const int output5 = 5;
const int output4 = 4;

// función millis para contabilizar el tiempo
unsigned long currentTime = millis();
// Valor asignado inicialmente
unsigned long previousTime = 0;
// Definición de desborde del contador (ejemplo: 2000ms = 2s)
const long timeoutTime = 2000;
```

Figura 3. 27: Encabezado del código.

Elaborado por: Autor.

En el código como se muestra, están comentadas que es lo que realiza cada parte del código para un mejor entendimiento por parte del lector, a continuación, se presenta la siguiente parte del código que es definir la función Setup dentro de la estructura del código. Esta función permite la inicialización de pines y puertos de comunicaciones, en la figura 3.28 se muestra este proceso.

```
void setup() {
  Serial.begin(115200);
  // Inicialización de pines de salida
  pinMode(output5, OUTPUT);
  pinMode(output4, OUTPUT);
  // Salidas definidas en estado bajo lógico
  digitalWrite(output5, LOW);
  digitalWrite(output4, LOW);

  // Conexion a red wifi
  Serial.print("Conectando a ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Muestra de IP adquirida por la red wifi
  Serial.println("");
  Serial.println("WiFi conectada.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  server.begin();
}
```

Figura 3. 28: Función Setup.

Elaborado por: Autor.

En la siguiente sección del código se encuentra la estructura del programa principal, el cual efectuara la verificación de conexión de clientes conectados al servidor web, muestra del estado por el puerto serie donde es posible visualizarlo en la computadora mediante un monitor, este se conectará con el convertidor USB-TTL, además, se encuentra en este una sección de las funciones a realizar cuando un cliente se conecta, figura 3.29.

```

void loop(){
  WiFiClient client = server.available(); // Escucha de clientes conectados al servidor

  if (client) {
    Serial.println("New Client."); // Si un nuevo cliente se conecta,
    String currentLine = ""; // Imprimir mensaje en el puerto serie
    // Hacer un String para mantener de incognito los datos
    currentTime = millis();
    previousTime = currentTime;
    // loop mantener mientras el cliente este conectado
    while (client.connected() && currentTime - previousTime <= timeoutTime) {
      currentTime = millis();
      if (client.available()) { // Si existe interacción del cliente con el servidor,
        char c = client.read(); // Leer los datos y entonces
        Serial.write(c); // Imprimir en el monitor serie
        header += c;
        if (c == '\n') { // Si la linea es un nuevo caracter
          // esta es la linea final de peticion HTTP del cliente al servidor
          if (currentLine.length() == 0) {
            // Encabezado HTTP (e.g. HTTP/1.1 200 OK)
            // Y a continuación se imprime dicha conexión del cliente:
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println("Connection: close");
            client.println();

```

Figura 3. 29: Función principal del código.  
Elaborado por: Autor.

En la siguiente parte del código, figura 3.30, se muestra las sentencias utilizadas para declarar el estado de las salidas digitales del módulo ESP8266, éstas de acuerdo a su estado, reflejaran en la tarjeta su valor lógico.

```

// Las salidas digitales estan en off
if (header.indexOf("GET /5/on") >= 0) {
  Serial.println("GPIO 5 on");
  output5State = "on";
  digitalWrite(output5, HIGH);
} else if (header.indexOf("GET /5/off") >= 0) {
  Serial.println("GPIO 5 off");
  output5State = "off";
  digitalWrite(output5, LOW);
} else if (header.indexOf("GET /4/on") >= 0) {
  Serial.println("GPIO 4 on");
  output4State = "on";
  digitalWrite(output4, HIGH);
} else if (header.indexOf("GET /4/off") >= 0) {
  Serial.println("GPIO 4 off");
  output4State = "off";
  digitalWrite(output4, LOW);
}

```

Figura 3. 30: Sección de código para activación de leds.  
Elaborado por: Autor.

A continuación, se muestra en la figura 3.31 la parte del código construido para la visualización de la página web escrito en http que permitirá

al usuario una interacción con la placa de desarrollo IoT, esta utiliza estilos CSS, fuentes y colores predeterminado en el lenguaje Html.

```
// Muestra pagina web
client.println("<!DOCTYPE html><html>");
client.println("<head><meta name='\"viewport\"' content='\"width=device-width, initial-scale=1\">");
client.println("<link rel='\"icon\"' href='\"data:,\">");
// Stilos CSS para encendido y apagado de los botones
client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}");
client.println(".button { background-color: #195B6A; border: none; color: white; padding: 16px 40px;}");
client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}");
client.println(".button2 {background-color: #77878A;}</style></head>");

// Encabezado de pagina web
client.println("<body><h1>ESP8266 Servidor Web de Tarjeta de desarrollo IoT</h1>");

// Muestra estados de los botones, on/off
client.println("<p>GPIO 5 - State " + output5State + "</p>");
// Si el estado es OFF, se muestra habilitado el ON
if (output5State=="off") {
  client.println("<p><a href='\"/5/on\"'><button class='\"button\"'>ON</button></a></p>");
} else {
  client.println("<p><a href='\"/5/off\"'><button class='\"button button2\"'>OFF</button></a></p>");
}
}
```

Figura 3. 31: Diseño HTML de visualización.  
Elaborado por: Autor.

Y como última parte del código (ver figura 3.32) las sentencias donde se interrumpe el ciclo while, limpia la cabecera, cierra la conexión y termina el programa principal con el fin de cerrar la interacción entre cliente y servidor.

```
// El HTTP responde al final con una linea en blanco
client.println();
// Interrumpe el Loop While
break;
} else { // si tienes una nueva línea, borra currentLine
currentLine = "";
}
} else if (c != '\r') { // Si se tiene un retorno de carril
currentLine += c; // Agregar al final de la linea actual
}
}
}
// Limpiar cabecera
header = "";
// Cierra conexión
client.stop();
Serial.println("Cliente desconectado.");
Serial.println("");
}
}
```

Figura 3. 32: Ultima sección del código.  
Elaborado por: Autor.

Ahora para el caso de envío de resultados del puerto analógico del uC ATmega 328p al módulo ESP8266 es necesario utilizar el siguiente código que se muestra en la figura 3.33. Aquí se adquieren los datos de uno de los puertos analógicos que están enrutados en la placa y convierte a una escala conocida, encapsula y envía por el puerto serie, este debe estar conectado al puerto serie del módulo ESP.

```

const int analogInPin0 = A0;

int Sensor1 = 0;

int outputValue = 0;

void setup() {
  // Inicializar el puerto serie de comunicaciones:
  Serial.begin(9600);
}

void loop() {
  // Lectura del puerto analogico
  sensor1 = analogRead(analogInPin0);
  // conversion de valores a una escala conocida:
  output1 = map(sensor1, 0, 1023, 0, 255);
  // change the analog out value:
  analogWrite(analogOutPin, outputValue);

  // envio de info a ESP8266r:
  Serial.println(output1);
  //tiempo para estabilizacion del puerto ADC
  delay(2);
}

```

Figura 3. 33: Código para envío de datos analógicos desde el microcontrolador ATmega al ESP8266.  
Elaborado por: Autor.

La figura 3.34 muestra la cabecera del código para el módulo ESP8266 que adquiere datos enviados por el uC y desencapsula estos, luego los procesa para enviarlos mediante el protocolo mqtt. El protocolo mqtt es necesario tener la librería descargada llamada PubSubClient, que trabaja de manera local o por internet conectándose a un bróker mqtt en una red remota.

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>

const char* ssid = ".....";
const char* password = ".....";
const char* mqtt_server = "definir_servidorMQTT";

WiFiClient espClient;
PubSubClient client(espClient);

void setup_wifi() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  randomSeed(micros());
}

```

Figura 3. 34: Encabezado del código del ESP8266.  
Elaborado por: Autor.

En la imagen anterior se definen los valores para las constantes ssid, password y mqtt\_server, esta configuración es necesaria antes de iniciar cualquier intervención en la comunicación con un servidor mqtt, además se crea una función en el que permite configurar los parámetros de la red wifi y además con este se procede a conectar el módulo a la red inalámbrica configurada.

A continuación, se presenta en la imagen 3.35, se presenta la siguiente parte de este código, como se puede apreciar se inicia con una función setup que permite inicializar el puerto UART, llamar la función setup\_wifi descrita anteriormente, y la configuración del cliente mqtt al servidor.

```
void setup() {
  Serial.begin(9600);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
}

void loop() {
  if (!client.connected()) {

    if (Serial.available()) {
      int datoRX = Serial.read();
      char buffer[10];
      dtostrf(datoRX,0, 2, buffer);
      client.publish("outTopic", buffer);
    }
  }
  client.loop();
}
```

Figura 3. 35: Función setup y principal del ESP8266  
Elaborado por: Autor.

Posteriormente se inicia la función principal donde cuestiona la conexión con el servidor mqtt, de ser cierta se procede a la lectura del puerto serial, en caso de tener una respuesta favorable entonces se asigna a una variable datoRX los valores leídos.

Posteriormente se crea un arreglo para el buffer y se asigna el valor recibido por el puerto serial a este, para terminar con publicación por mqtt al servidor con el valor guardado en el buffer.

Para finalizar, se cierran los if anidados y se llama a `client.loop ()`, el cual es una función nativa de la librería `pubsubclient`, esta permite la estabilización de la conexión y limpieza del buffer dentro de la librería.

## Conclusiones

Tras el análisis de los objetivos generales y específicos que se enunciaron previamente, es posible concluir lo siguiente:

- Se logro describir la teoría del proyecto, apuntando siempre al internet de las cosas (IoT) utilizando dispositivos que se encuentran en el medio.
- Se pudo diseñar una tarjeta de desarrollo utilizando el módulo ESP8266 siendo adaptable para el uso de estudiantes de telecomunicaciones de la Facultad de Educacion Técnica para el Desarrollo.
- Se implementó el prototipo en simulaciones donde se pudo ver los diferentes aplicativos en la que se puede usar este prototipo utilizando protocolo mqtt uno de los más conocidos en el medio de comunicaciones inalámbricas sobre IoT.

## **Recomendaciones.**

- Se sugiere la posibilidad del estudio del uso de aplicaciones no solo para IoT, sino en comunicaciones en general basados en la banda de frecuencia de 2.4 GHz.
- Se recomienda en la aplicación desarrollada utilizar elementos de buena calidad para la implementación física de la tarjeta de desarrollo.
- Es posible que esta tarjeta no sea solo para prácticas de laboratorio, si no también para investigación en el campo del IoT, así también comenzar a utilizarla para la adquisición de datos y posteriormente apuntar a un uso de la Big Data.

## Bibliografía.

- Biswas, A. R., & Giaffreda, R. (2014). IoT and cloud convergence: Opportunities and challenges. *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 375–376. <https://doi.org/10.1109/WF-IoT.2014.6803194>
- Corporación Ruta N. (2015). Observatorio CT+i: Informe No. 1 Área de oportunidad en Internet of Things. Recuperado de Ruta N Medellín website:  
[https://www.rutanmedellin.org/images/biblioteca/observatoriocti/2015/3\\_TICS/VT\\_INTERNET-OF-THINGS\\_TECNNOVA\\_UNAL.pdf](https://www.rutanmedellin.org/images/biblioteca/observatoriocti/2015/3_TICS/VT_INTERNET-OF-THINGS_TECNNOVA_UNAL.pdf)
- González García, A. J. (2017). *IoT: Dispositivos, tecnologías de transporte y aplicaciones* (Máster Universitario de Ingeniería de Telecomunicación, Universitat Oberta de Catalunya). Recuperado de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/64286/3/agonzalezgarcia0TFM0617memoria.pdf>
- Hammi, B., Khatoun, R., Zeadally, S., Fayad, A., & Khoukhi, L. (2018). IoT technologies for smart cities. *IET Networks*, 7(1), 1–13. <https://doi.org/10.1049/iet-net.2017.0163>
- IoTNet. (2019). Solutions IoT: Smart city concept with different icon and elements. Modern city design with future technology for living. Recuperado el 19 de julio de 2019, de Solutions IoT website: <http://www.iotnet.eu/wp/solutions/>
- Jayaraman, P., Yavari, A., Georgakopoulos, D., Morshed, A., & Zaslavsky, A. (2016). Internet of Things Platform for Smart Farming: Experiences and Lessons Learnt. *Sensors*, 16(11), 1884. <https://doi.org/10.3390/s16111884>
- Romero García, M. T. (2017). *La protección de datos ante el Internet de las cosas* (Trabajo Fin de Carrera, Universidad Politécnica de Madrid). Recuperado de

[http://oa.upm.es/47426/1/TFC\\_MARIA\\_TERESA\\_ROMERO\\_GARCIA.pdf](http://oa.upm.es/47426/1/TFC_MARIA_TERESA_ROMERO_GARCIA.pdf)

## DECLARACIÓN Y AUTORIZACIÓN

Yo, **HIGUERA ASTUDILLO, LUIS ALBERTO** con C.C: # 092385830-2 autor del Trabajo de **componente práctico del examen complejo: Diseño de una tarjeta de desarrollo con ESP8266 orientada a wireless y microcontroladores para IoT**, previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 16 de septiembre del 2019

f. \_\_\_\_\_

Nombre: HIGUERA ASTUDILLO, LUIS ALBERTO

C.C: 092385830-2

## **REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA**

### **FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN**

<b>TÍTULO Y SUBTÍTULO:</b>	DISEÑO DE UNA TARJETA DE DESARROLLO CON ESP8266 ORIENTADA A WIRELESS Y MICROCONTROLADORES PARA IOT.		
<b>AUTOR(ES)</b>	HIGUERA ASTUDILLO, LUIS ALBERTO		
<b>REVISOR(ES)/TUTOR(ES)</b>	M. Sc. PACHECO BOHÓRQUEZ, HÉCTOR IGNACIO		
<b>INSTITUCIÓN:</b>	Universidad Católica de Santiago de Guayaquil		
<b>FACULTAD:</b>	Facultad de Educación Técnica para el Desarrollo		
<b>CARRERA:</b>	Ingeniería en Telecomunicaciones		
<b>TÍTULO OBTENIDO:</b>	Ingeniero en Telecomunicaciones		
<b>FECHA DE PUBLICACIÓN:</b>	16 de septiembre del 2019	<b>No. DE PÁGINAS:</b>	45
<b>ÁREAS TEMÁTICAS:</b>	Fundamentos de comunicación, comunicaciones inalámbricas		
<b>PALABRAS CLAVES/ KEYWORDS:</b>	Comunicaciones, Inalámbrico, Internet, IoT, Dispositivos, Sistemas		
<b>RESUMEN/ABSTRACT:</b>	<p>El presente documento presente el desarrollo del componente práctico del examen complejo, denominado "Diseño de una tarjeta de desarrollo con ESP8266 orientada a wireless y microcontroladores para IoT". En general, estamos entrando en una nueva era de tecnología informática, es decir, Internet de las cosas (IoT). IoT es una especie de "red neuronal global universal" en la nube que conecta varias cosas. IoT es un dispositivo y un sistema inteligentemente conectado que comprende máquinas inteligentes que interactúan y se comunican con otras máquinas, entornos, objetos e infraestructuras y la identificación por radiofrecuencia (RFID) y las tecnologías de red de sensores se enfrentarán para enfrentar este nuevo desafío. En el capítulo 1, se describe una introducción de IoT, objetivo general y específico del componente práctico. En el capítulo 2, se describe brevemente la parte fundamental de IoT. Finalmente, en el capítulo 3, se muestra el propósito del presente trabajo que fue diseñar una tarjeta electrónica que utiliza el dispositivo ESP8266 para realizar la comunicación inalámbrica orientada al internet de las cosas.</p>		
<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
<b>CONTACTO CON AUTOR/ES:</b>	<b>Teléfono:</b> +593-9-97964705	<b>E-mail:</b> <a href="mailto:higuera.12@hotmail.com">higuera.12@hotmail.com</a>	
<b>CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE</b>	<b>Nombre:</b> Palacios Meléndez Edwin Fernando		
	<b>Teléfono:</b> +593-9-67608298		
	<b>E-mail:</b> <a href="mailto:edwin.palacios@cu.ucsg.edu.ec">edwin.palacios@cu.ucsg.edu.ec</a>		
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>			
<b>Nº. DE REGISTRO (en base a datos):</b>			
<b>Nº. DE CLASIFICACIÓN:</b>			
<b>DIRECCIÓN URL (tesis en la web):</b>			