

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO

DE TUXTLA GUTIÉRREZ

**Departamento de ingeniería Eléctrica y Electrónica**

**Reporte de Residencia**

Título de proyecto:

**Sistema de comunicación remoto para medir en tiempo real la temperatura  
utilizando sensores de fibra óptica.**

Presenta

**Ariel Fong González**

Asesor

**Dr. Jorge Luis Camas Anzueto**

Tuxtla Gutiérrez, Chiapas, México; 07 de enero de 2019



# **AGRADECIMIENTOS**

A mis asesores: **Dr. Jorge Luis Camas Anzueto** y **Dr. Rubén Grajales Coutiño**.

Por regalarme la oportunidad de formar parte en el laboratorio Optomecatrónica, en el grupo SPIE, así como por sus conocimientos compartidos, su confianza, su apoyo y su amistad a lo largo de este tiempo.

A mis maestros de las diferentes materias a lo largo de estos 9 semestres.

A mis compañeros y amigos con quienes tuve la oportunidad de trabajar durante la carrera.

# DEDICATORIA

## **A mi Dios.**

Por otorgarme la vida durante todos estos años, por la salud y bondades recibidas, así como también las cargas que me ha quitado.

## **A mis padres Ariel Fong Guzmán y Eliza González Aquino.**

Por ser siempre mi modelo a seguir. Gracias a ellos por formar al hombre que hoy en día soy, por sus consejos y lecciones pues de ellos he aprendido los valores que a diario utilizo.

Gracias por su apoyo incondicional y por ser la calidad de padres ejemplares que muchos quisieran tener.

## **A mis hermanos Elisa, Alejandro y Genesis, y mi sobrino Abner.**

Por estar siempre conmigo y apoyarme en cada una de las decisiones que tomo. A ellos también les agradezco.

# ÍNDICE

ÍNDICE DE FIGURAS .....	7
<b>Capítulo 1. GENERALIDADES .....</b>	<b>10</b>
<b>1.1 Introducción.....</b>	<b>11</b>
<b>1.2. Descripción de la institución .....</b>	<b>12</b>
<b>1.2.1. Historia del ITTG.....</b>	<b>12</b>
<b>1.2.2. Valores .....</b>	<b>12</b>
<b>1.2.3. Misión.....</b>	<b>13</b>
<b>1.2.4. Visión .....</b>	<b>13</b>
<b>1.2.5. Localización .....</b>	<b>13</b>
<b>1.3. Objetivos.....</b>	<b>13</b>
<b>1.3.1. Objetivo General: .....</b>	<b>13</b>
<b>1.3.2. Objetivos Específicos:.....</b>	<b>13</b>
<b>Capítulo 2. MARCO TEÓRICO.....</b>	<b>15</b>
<b>2.1. Propiedades de la Luz.....</b>	<b>15</b>
<b>2.1.1. Ondas Electromagnéticas .....</b>	<b>15</b>
<b>2.1.2. Espectro Electromagnético.....</b>	<b>16</b>
<b>Propagación de la luz.....</b>	<b>17</b>
<b>2.2. Fibras Ópticas .....</b>	<b>19</b>
<b>Introducción .....</b>	<b>19</b>

Composición de una fibra óptica.....	21
Propagación de la luz dentro de la fibra óptica .....	22
<b>COMUNICACIONES POR FIBRA ÓPTICA.....</b>	<b>22</b>
Aspectos principales de la propagación de la luz en las fibras ópticas .....	23
Criterios de clasificación de las fibras .....	24
<b>2.3. Sensores de fibra óptica .....</b>	<b>25</b>
<b>2.3.1. Definición y características generales .....</b>	<b>25</b>
<b>2.4. Internet de las cosas .....</b>	<b>27</b>
<b>Capítulo 3. MATERIALES E INFORMACIÓN.....</b>	<b>29</b>
<b>3.1. Recopilación de información .....</b>	<b>29</b>
<b>3.1.1. Arduino Ethernet.....</b>	<b>29</b>
<b>3.1.2. ESP8266.....</b>	<b>31</b>
<b>3.1.3. NodeMCU .....</b>	<b>39</b>
<b>3.1.4. Selección de placa Arduino optima .....</b>	<b>53</b>
<b>3.1.5. Servidores web.....</b>	<b>54</b>
<b>3.1.6. OPF372 y OPF472 .....</b>	<b>61</b>
<b>CAPÍTULO 4. DESARROLLO DE PRUEBAS .....</b>	<b>66</b>
<b>4.1. Dispositivos OPF.....</b>	<b>66</b>
<b>CAPÍTULO 5. RESULTADOS .....</b>	<b>80</b>
<b>5.1. ThingSpeak.....</b>	<b>80</b>

<b>5.1.1. Características principales.....</b>	<b>81</b>
<b>5.1.2. Apps en ThingSpeak.....</b>	<b>82</b>
<b>5.2. Programación.....</b>	<b>83</b>
<b>5.3. Grafica de datos en tiempo real .....</b>	<b>86</b>
<b>Conclusión.....</b>	<b>92</b>
<b>REFERENCIAS.....</b>	<b>93</b>

# ÍNDICE DE FIGURAS

FIGURA 1 - LOCALIZACIÓN ITTG.....	13
FIGURA 2 - ONDA ELECTROMAGNÉTICA .....	15
FIGURA 3 - ESPECTRO ELECTROMAGNÉTICO .....	16
FIGURA 4 - REFLEXIÓN DE LA LUZ.....	17
FIGURA 5 - REFRACCIÓN DE LA LUZ.....	18
FIGURA 6 - ENLACE DE PUNTO A PUNTO MEDIANTE FIBRAS .....	20
FIGURA 7 - COMPONENTES DE LA FIBRA ÓPTICA.....	21
FIGURA 8 – CLASIFICACIÓN DE FIBRAS .....	24
FIGURA 9 - ESTRUCTURA GENERAL DE UN SISTEMA DE MEDICIÓN BASADO EN UN SENSOR DE FIBRA ÓPTICA. ....	27
FIGURA 10 - ARDUINO ETHERNET .....	29
FIGURA 11 - ESP8266.....	32
FIGURA 12- ESP8266 CARACTERÍSTICAS FÍSICAS .....	33
FIGURA 13 - MODOS DE OPERACIÓN .....	35
FIGURA 14 - URLS PLACAS ADICIONALES.....	36
FIGURA 15 - URLS PLACAS ADICIONALES.....	36
FIGURA 16 - GESTOR DE TARJETAS .....	37
FIGURA 17 - GESTOR DE TARJETAS .....	37
FIGURA 18 - MÓDULOS ESP .....	38
FIGURA 19 - MÓDULOS CON ESP8266.....	39
FIGURA 20 - NODEMCU .....	40
FIGURA 21 - VERSIONES DE NODEMCU .....	43
FIGURA 22 - VERSIONES NODEMCU .....	44
FIGURA 23 - NODEMCU V1 .....	44
FIGURA 24 - NODEMCU V2 .....	45
FIGURA 25 - NODEMCU V3 .....	46
FIGURA 26 - DIAGRAMA DE PINES NODEMCU V2 .....	47
FIGURA 27 - DIAGRAMA DE PINES DIGITALES .....	48
FIGURA 28 - PIN ANALOGO.....	49
FIGURA 29 - URLS PLACAS ADICIONALES.....	50
FIGURA 30 - URLS PLACAS ADICIONALES.....	51
FIGURA 31 - GESTOR DE TARJETAS .....	51
FIGURA 32 - GESTOR DE TARJETAS .....	52
FIGURA 33 - MÓDULOS ESP .....	52
FIGURA 34 - CONFIGURACIÓN NODEMCU .....	53

FIGURA 35 - OPF372A.....	61
FIGURA 36 - POTENCIA DE ACOPLE .....	62
FIGURA 37 - ESTRUCTURA FÍSICA OPF372.....	63
FIGURA 38 - ESPECIFICACIONES ELÉCTRICAS.....	63
FIGURA 39 - ESTRUCTURA FÍSICA OPF472.....	65
FIGURA 40 - ESPECIFICACIONES ELÉCTRICAS.....	65
FIGURA 41 - DISEÑO ESQUEMÁTICO .....	67
FIGURA 42 - DISEÑO PCB .....	67
FIGURA 43 - DISEÑO 3D.....	67
FIGURA 44 - PLACA .....	67
FIGURA 45 - DISEÑO ESQUEMÁTICO .....	68
FIGURA 46 - DISEÑO PCB .....	68
FIGURA 47 - DISEÑO 3D.....	68
FIGURA 48 - PLACA .....	68
FIGURA 49 – POTENCIA 0 A 100 °C .....	69
FIGURA 50 – POTENCIA 100 A 0°C.....	70
FIGURA 51 - VOLTAJE 0 A 100 °C .....	71
FIGURA 52 - VOLTAJE 100 A 0 °C .....	72
FIGURA 53 - DHT11.....	73
FIGURA 54 – PROGRAMACIÓN DHT11 .....	74
FIGURA 55 – PROGRAMACIÓN DHT11 .....	74
FIGURA 56 – PROGRAMACIÓN DHT11 .....	75
FIGURA 57 - SENSOR TH100PT .....	75
FIGURA 58 - MAX31865.....	76
FIGURA 59 - PROGRAMACIÓN TH100PT.....	77
FIGURA 60 - PROGRAMACIÓN TH100PT.....	77
FIGURA 61 - PROGRAMACIÓN TH100PT.....	78
FIGURA 62 - ESTRUCTURA POR ZONAS DE UNA FIBRA ÓPTICA ESTRECHADA (TAPER).....	79
FIGURA 63 - THINGSPEAK .....	81
FIGURA 64 - PROGRAMACIÓN STFO .....	84
FIGURA 65 - PROGRAMACIÓN STFO .....	84
FIGURA 66 - PROGRAMACIÓN STFO .....	85
FIGURA 67 - PROGRAMACIÓN STFO .....	86
FIGURA 68 - QUANTUM TC125 Y QPOD .....	86
FIGURA 69 - TRASMISOR EN OPERACIÓN .....	87



FIGURA 70 - VARIACIONES DE TEMP. ....	88
FIGURA 71 - SENSOR DE TEMPERATURA.....	88
FIGURA 72 - SENSOR DENTRO DEL QPOD.....	89
FIGURA 73 - SENSADO DE TEMPERATURA .....	89
FIGURA 74 – TEMPERATURA .....	90
FIGURA 75 - HUMEDAD .....	90
FIGURA 76 - VOLTAJE.....	91
FIGURA 77 - CHANNEL THINGSPEAK.....	91

# ACRÓNIMOS

**Hz** Hertz.

**Km** Kilometros.

**Mi** Millas.

**S** Segundos.

**nm** nanómetros.

**LED** Light-Emitting Diode (Diodo emisor de luz).

**TIC** Tecnologías de la información y la comunicación.

**IoT** Internet of Things (Internet de las cosas).

**IO** Internet de los Objetos.

**PCB** Printed circuit board (Placa de circuito impreso).

**URL** Uniform Resource Locator (localizador de recursos uniforme).

**HTML** HyperText Markup Language (Lenguaje de Marcado para Hipertextos).

**HTTP** hypertext transfer protocol (Protocolo de transferencia de hipertexto)

**GaAIAs** Aluminio Galio Arsenio

**µm** micrómetros

**RTD** Resistive Temperature Detector (detector de temperatura resistivo)

**ADC** Analog to Digital Converter (convertidor de señal analógica a digital)

**mV** miliVolts

# Capítulo 1. GENERALIDADES

## 1.1 Introducción

El hombre como ser social ha tenido siempre la necesidad de comunicarse, es por ello que a lo largo de la historia las formas de comunicarnos han ido cambiando favorablemente. Pues hemos partido desde las señales y gritos en los tiempos de las cavernas, así como las pinturas rupestres con las cuales enmarcaban las antiguas vivencias. Pasando por señales de humo, mensajeros, el periódico, el telégrafo de Graham Bell, la radiotelegrafía de Marconi, las comunicaciones por televisión, etc.

Historias antiguas cuentan que la luz también ha servido para comunicarse a distancia, pero se ha visto limitadas por las técnicas no tan precisas y por las condiciones climáticas o atmosféricas en las cuales se han trabajado.

En los tiempos de la antigua Grecia, la luz solar solía hacerse reflejar mediante espejos, para hacer señales entre torres. Esta técnica ha ido variando conforme al tiempo, reemplazando la luz solar por luz artificial, codificándola mientras se enciende y apaga esa luz, semejando el código morse.

Durante el siglo XIX, Alexander Graham Bell trabajó en el proyecto denominado: "Photophone" para enviar señales de voz mediante luz solar. Consistía en un espejo que reflejaba la luz solar el cual vibraba mediante las ondas de voz. Al llegar al receptor el cual era una fotocélula la cual emitía una corriente eléctrica hacia un altavoz.

Con la invención del "Láser" en 1958 se llevaron a cabo estudios de su transmisión mediante el aire, estas líneas de transmisión podían ser redireccionadas mediante espejos, pero nuevamente las condiciones climatológicas impedían mejoras en los procesos de comunicación.

Hasta que en 1970 se descubrió la primer Fibra Óptica de bajas pérdidas, fabricada en Silicio de 250 micras de diámetro, similar al diámetro de un cabello. Fue entonces que utilizando a la fibra como medio de transporte se lograban "inmunidad" al ruido que pudiera ser propiciado por corrientes eléctricas o alteraciones climatológicas.

En el siguiente reporte se explican de manera general las actividades realizadas a lo largo de 16 semanas de trabajo, aplicadas dentro del laboratorio de Optomecatrónica del Instituto Tecnológico de Tuxtla Gutiérrez, en Tuxtla Gutiérrez, Chiapas, México. Se lleva a cabo el proyecto titulado: "".

Revisando en este apartado un poco las teorías de las propiedades de la luz, fibras ópticas, sensores de fibra óptica y el internet de las cosas, teorías con las que nos hemos basado para realizar el proyecto en cuestión.

La finalidad del proyecto tiene como principio fundamental la recepción de la señal de luz para ser convertida a voltaje. Esta variable, el voltaje es transmitida a un servidor web para su graficación en tiempo real.

## **1.2. Descripción de la institución**

### **1.2.1. Historia del ITTG**

El Instituto Tecnológico de Tuxtla Gutiérrez es una institución pública de educación superior, dependiente de la Secretaría de Educación Pública. Imparte 8 licenciaturas y 2 programas de posgrado en las áreas de Ingeniería, Tecnología y Ciencias Administrativas.

El Instituto Tecnológico de Tuxtla Gutiérrez fue fundado el 24 de octubre de 1972, se encuentra ubicado en el centro del estado de Chiapas.

La investigación para el desarrollo del proyecto se llevó a cabo Dentro del Instituto Tecnológico de Tuxtla Gutiérrez, donde el alumno ubica su área de trabajo dentro del laboratorio de Optomecatrónica (Z3) - ubicado en el edificio "Z" (departamento de posgrado) – haciendo uso del material ofrecido dentro.

### **1.2.2. Valores**

- El Ser Humano      Espíritu de Servicio      Liderazgo      Trabajo en Equipo
- La Calidad      Alto Desempeño      Respeto al Medio Ambiente

### 1.2.3. Misión

Formar de manera integral profesionistas de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora, respeto al medio ambiente y apego a los valores éticos.

### 1.2.4. Visión

Ser una Institución de Excelencia en la Educación Superior Tecnológica del Sureste, comprometida con el desarrollo socioeconómico sustentable de la región.

### 1.2.5. Localización

En la figura 1 se muestra el mapa donde se puede apreciar la localización del ITTG.

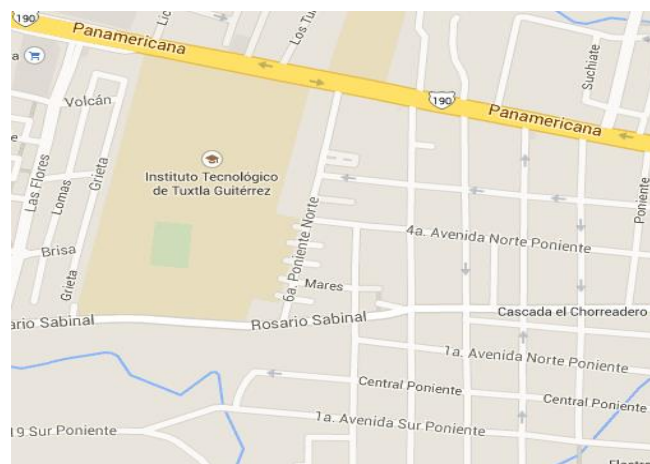


Figura 1 - Localización ITTG

## 1.3. Objetivos

### 1.3.1. Objetivo General:

- Implementar un sistema de comunicación que pueda transmitir a un servidor web la información recopilada de los sensores de fibra óptica.

### 1.3.2. Objetivos Específicos:

- Establecer comunicación entre el servidor web y el dispositivo emisor de datos.

- Caracterización de transmisión de potencia y voltaje a través de sensores de fibra óptica.
- Acoplamiento de señales para la transmisión de la variable voltaje al servidor web para su graficación en tiempo real.

## Capítulo 2. MARCO TEÓRICO

### 2.1. Propiedades de la Luz

#### 2.1.1. Ondas Electromagnéticas

Una onda es una propagación de una perturbación de alguna propiedad en un medio determinado.

Una onda electromagnética es la propagación simultánea de los campos eléctrico y magnético producidos por una carga eléctrica en movimiento (Fig.2). Las principales características de las ondas electromagnéticas que son:

- No necesitan un medio para propagarse: pueden propagarse en el vacío o en cualquier otro medio.
- Son tridimensionales: se propagan en las tres direcciones del espacio.
- Son transversales: la perturbación se produce perpendicularmente a la dirección de propagación.

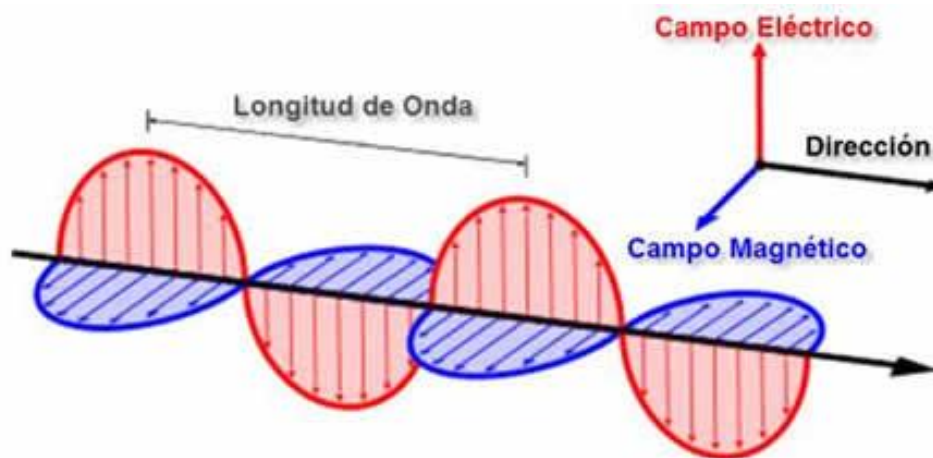


Figura 2 - Onda Electromagnética

## 2.1.2. Espectro Electromagnético

El espectro electromagnético es el conjunto de todas las frecuencias (número de ciclos de la onda por unidad de tiempo) posibles a las que se produce radiación electromagnética. (Fig.3 representa el espectro y sus frecuencias).

La luz se comporta como una onda electromagnética y pertenece al espectro electromagnético. El número de oscilaciones por segundo que completa una onda electromagnética se denomina *frecuencia*. La luz visible tiene una frecuencia alrededor de  $2,3 \times 10^{14}$  Hz (*Hertz*).

Actualmente se han detectado frecuencias electromagnéticas inferiores a 30 Hz y superiores a  $2,9 \times 10^{27}$  Hz. Aunque formalmente el espectro es infinito y continuo, se cree que la longitud de onda electromagnética (distancia entre dos valores de amplitud máxima de la onda) más pequeña posible es la longitud de Planck ( $l_p \approx 1,616252 \times 10^{-35}$  m), distancia o escala de longitud por debajo de la cual se espera que el espacio deje de tener una geometría clásica (medidas inferiores no pueden ser tratadas en los modelos de física actuales debido a la aparición de efectos de gravedad cuántica). Igualmente, se piensa que el límite máximo para la longitud de una onda electromagnética sería el tamaño del universo.

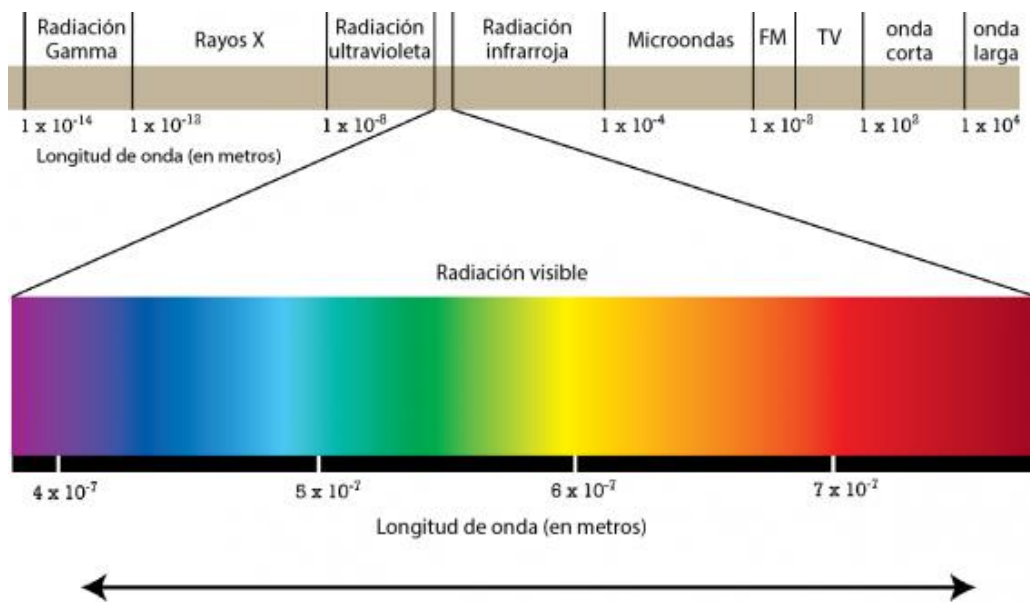


Figura 3 - Espectro Electromagnético



## Propagación de la luz

A las ondas luminosas se les referencia por su longitud de onda, que está relacionada con la frecuencia mediante la expresión  $\lambda = c/f$ , donde:  $\lambda$  es la longitud de onda,  $c$  la velocidad de la luz y  $f$  es la frecuencia.

En el espacio libre, la luz viaja en línea recta a la velocidad de 299.800 Km/S o 186.299 Mi/S. La dirección a lo largo de la cual se propagan las ondas de luz se denomina Rayo de Luz y se usa en la Óptica para explicar muchas características de la fibra óptica.

La propagación de la luz de un medio hacia otro da parte a dos fenómenos más:

## Reflexión

La reflexión se produce cuando una onda encuentra una superficie contra la cual rebota. Mientras más pulida y limpia es la superficie, mayor es la intensidad del haz reflejado (Fig.4). La reflexión producida por un espejo es llamada especular o regular, mientras que la producida por una superficie porosa, o irregular, se llama difusa. La reflexión difusa no produce imágenes. La reflexión especular se describe mediante las leyes de la reflexión:

- El rayo incidente, el reflejado y la normal están en un mismo plano.
- Tanto la velocidad del rayo incidente y el rayo reflejado, como los ángulos de incidencia y reflexión, son iguales:  $i = r$ .



$$\text{Ángulo de incidencia} = \text{Ángulo de reflexión}$$

Figura 4 - Reflexión de la luz.

## Refracción

La refracción tiene lugar cuando una onda que se propaga en un medio pasa a otro en el cual su velocidad de propagación es distinta (Fig.5). Como consecuencia de esa distinta velocidad de propagación se produce una especie de “flexión” de la onda, que modifica su dirección de propagación. Al pasar de un medio a otro en el cual la velocidad es distinta, la longitud de onda va a variar, mientras que la frecuencia permanece inalterada.

Para las ondas luminosas se define el índice de refracción del medio,  $n$ , como el cociente entre la velocidad de la luz en el aire,  $c$ , y la velocidad de la luz en el medio,  $v$ . Entonces  $n = c/v$ .

### Leyes de la refracción

1. El rayo incidente, el refractado y la normal están en un mismo plano.
2. La relación entre el ángulo de incidencia y el de refracción viene dado por la siguiente expresión (Ley de Snell)

$$n_1 \text{ sen } i = n_2 \text{ sen } r$$

donde  $n_1$  es el índice de refracción del primer medio, o medio en el que se propaga el rayo incidente, y  $n_2$  es el índice de refracción del segundo medio o medio en el que se propaga el rayo refractado.

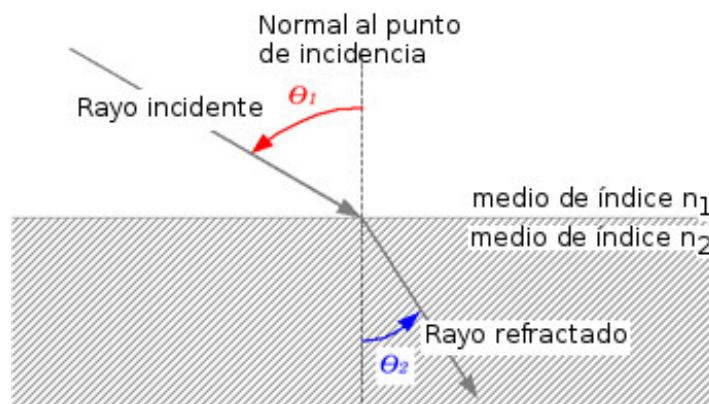


Figura 5 - Refracción de la luz

## 2.2. Fibras Ópticas

### Introducción

Durante las últimas décadas, los avances en el campo de las telecomunicaciones y la sensórica han venido de la mano con la implementación de una tecnología revolucionaria, que ha alcanzado difusión a escala mundial y cuyo desarrollo aún no se ha detenido en virtud al interés que ha despertado en el ámbito científico. Lógicamente se habla de los dispositivos ópticos, los cuales son utilizados para procesos, monitoreo y control de sistemas (Derickson: 1998), y de las fibras ópticas; aquel medio tan grueso como un cabello que guía la luz a través de enormes distancias, transportando datos de manera tan eficiente y con las ventajas que, frente a otras tecnologías, la ubicaron en la importante posición actual (la fig.6 muestra un esquema de comunicación por fibra).

Los sistemas de fibra óptica utilizan la energía óptica como soporte, por lo que presentan un conjunto importante de ventajas sobre otros soportes utilizados en la transmisión y tratado de señales analógicas y digitales. Entre ellas están:

- Alta sensibilidad a los cambios que se producen en su entorno
- Gran ancho de banda, que permite la transmisión de un gran volumen de información a altas velocidades
- Atenuación baja. Las pérdidas en intensidad son menores, lográndose mayor alcance.
- Inmunidad a interferencias electromagnéticas. La fibra óptica es absolutamente inmune a las interferencias de radiofrecuencias e impulsos electromagnéticos, lo que minimiza errores en la transmisión de señales digitales. Esto es de gran importancia en aplicaciones de control industrial donde se genera gran cantidad de ruido
- Seguridad y aislamiento eléctrico. En determinadas aplicaciones para ambientes peligrosos (ambientes explosivos o inflamables) o en electromedicina, las fibras ópticas son imprescindibles debido a la imposibilidad de producir descargas eléctricas o chispas.

- Menor peso y volumen. Comparando las fibras ópticas y los cables coaxiales necesarios para obtener las mismas prestaciones, las primeras ocupan un volumen muy inferior y tienen menor peso
- Seguridad frente a posibles intervenciones de la línea. Aunque no es imposible intervenir una fibra óptica, resulta más difícil que en otros medios y es fácilmente detectable.

El actual abanico de aplicaciones de las fibras ópticas es muy amplio y continúa en aumento; no obstante, dos de los campos de mayor importancia son:

1. Telecomunicaciones: Por su gran ancho de banda, la fibra óptica es el medio de transmisión preferido: en la actualidad las fibras comunican continentes y regiones apartadas a través de cables que se han esparcido por océanos, aire y tierra para unir a todo el mundo en la comunidad global que hoy tenemos. Esto, a su vez, ha impulsado el desarrollo de nuevos dispositivos relacionados con esta tecnología como circuladores ópticos, amplificadores, compensadores, multiplexores, demultiplexores, WDM, aisladores, controladores de polarización, fuentes de luz, etc. (Schott, 2008).
2. Sensores: Se han desarrollado dispositivos capaces de medir parámetros físicos como presión, temperatura, deformaciones mecánicas, entre otros. Que son utilizados en los sectores eléctrico, gas, petróleo, civil, medicina, aeroespacial, automotriz, químico, metrología y control, medio ambiente, etcétera. (Gavea SENSORS, 2008).

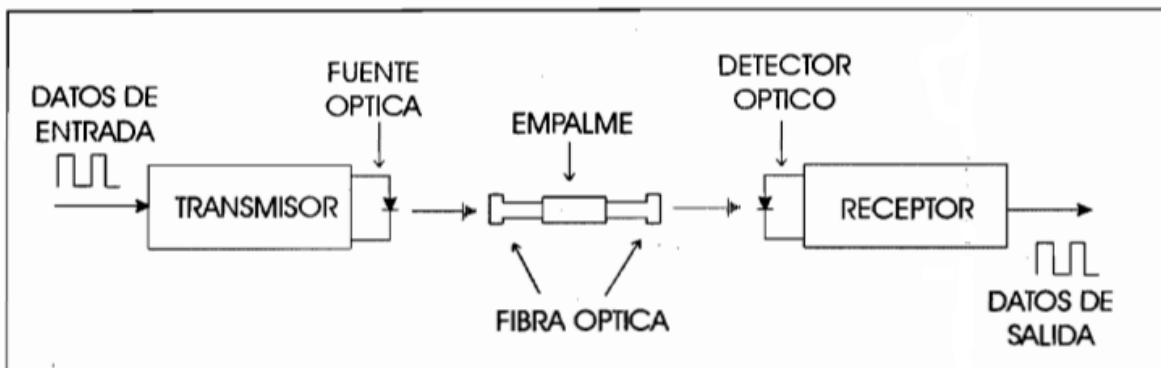


Figura 6 - Enlace de punto a punto mediante fibras

## Composición de una fibra óptica

Una fibra óptica consiste en un material cilíndrico y largo que confina y propaga ondas luminosas. Esta de tres capas diferentes: el núcleo central que lleva la luz, el revestimiento que cubre el núcleo y que confina la luz dentro del núcleo, y el recubrimiento que dota de protección al revestimiento (Fig.7). El núcleo y el revestimiento están formados frecuentemente por vidrio de silicio mientras que el recubrimiento es un plástico o cubierta acrílica.

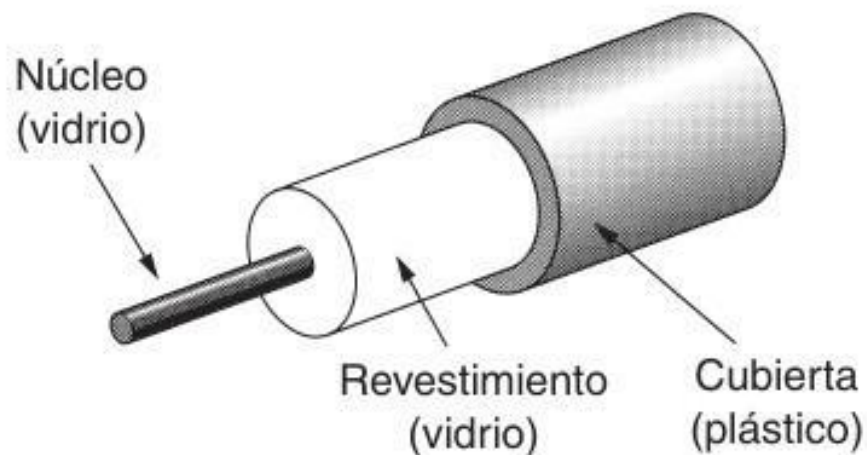


Figura 7 - Componentes de la fibra óptica

Las capas del núcleo de silicio y del revestimiento difieren ligeramente en su composición debido a pequeñas cantidades de materiales como boro o germanio, que son añadidos durante el proceso de fabricación. Esto altera las características del índice de refracción de ambas capas, dando lugar a las propiedades de confinamiento de luz necesarias para la propagación de los rayos.

El índice de refracción del núcleo de silicio tiene un valor alrededor de 1.5 y el del revestimiento es ligeramente menor, alrededor de 1.48. El índice de refracción del aire es de 1.003. El recubrimiento de la fibra está normalmente coloreado, usando códigos de colores estándar del fabricante, que facilitan la identificación de la fibra. Las fibras ópticas pueden también estar hechas completamente de plástico o de otros materiales. Estas fibras son generalmente más caras, pero tienen una mayor atenuación (perdidas) y una aplicación limitada.

## **Propagación de la luz dentro de la fibra óptica**

El núcleo y el recubrimiento de la fibra óptica, están compuestos de vidrio silíceo. El núcleo y el recubrimiento se dopan en grado diferente, para que el núcleo tenga un índice de refracción un poco mayor que el recubrimiento.

En el caso más sencillo de la fibra del índice escalonado, el índice de refracción es constante tanto en el núcleo como en el recubrimiento. La luz, guiada en el núcleo de vidrio, al incidir en la separación entre el núcleo y el recubrimiento, es reflejada totalmente cuando el ángulo de incidencia es más pequeño que el ángulo de reflexión total.

## **COMUNICACIONES POR FIBRA ÓPTICA**

En las comunicaciones por fibra óptica se trabaja con radiaciones electromagnéticas con longitudes de onda comprendidas entre 750 nm y 1650 nm (las radiaciones visibles al ojo humano están comprendidas entre 400 nm y 750 nm). En general se usan tres puntos de trabajo bien definidos denominados:

- Primera Ventana (850 nm)
- Segunda Ventana (1310 nm)
- Tercera Ventana (1559 nm)

El empleo de las diferentes ventanas depende de cómo se pueden obtener las mejores prestaciones de transmisión de las fibras, o del equipo óptico en su conjunto. Definidas las características geométricas y físicas de una fibra, al variar la longitud de onda " $\lambda$ ", varía la cantidad de modos que pueden ser conducidos. Cada modo se caracteriza por un valor de longitud de onda, más allá del cual ese modo no puede propagarse. A esta regla escapa sólo un modo, llamado Modo Fundamental, que existe en todos los  $\lambda$ .

Del mismo modo, una vez determinada la longitud de onda de la radiación, disminuyendo el diámetro del núcleo disminuye el número de modos en que se propagan.

Al ser la radiación luminosa considerada de dimensiones muy pequeñas, los rayos quedan confinados en el plano que pasa por el eje de la fibra.

En las fibras ópticas la luz se propaga por reflexión total interna en la interfase entre el núcleo y el recubrimiento. Por lo tanto, puede definirse un ángulo de aceptación máximo, más allá del cual la radiación que incide el recubrimiento no se propagará. Para establecer la diferencia entre las fibras multimodo y monomodo, además de tener en cuenta que tienen índice de refracción y dimensiones geométricas diferentes, se debe conocer un parámetro llamado "longitud de onda de corte o límite", que define el punto sobre el cual la propagación tiene lugar en forma monomodal.

### **Aspectos principales de la propagación de la luz en las fibras ópticas**

Los aspectos principales para la propagación de la luz en las fibras ópticas son:

- Que la pureza del material del núcleo sea tan alta, que la atenuación se mantenga dentro de los límites razonables.
- Que los rayos, que por una razón y otra tiendan a cambiar su dirección de propagación, se mantengan dentro del núcleo de la fibra.

La alta pureza fue un problema en el procesamiento del material de la fibra que ya ha sido resuelto.

Cabría esperar que tanto el índice de refracción como la transparencia, varíen con la longitud de onda y la temperatura. Una cierta pérdida por dispersión desde la fibra no puede, por razones teóricas, ser evitada: la así llamada dispersión de Rayleigh.

A más largas longitudes de onda las pérdidas suben debido a la absorción por rayos infrarrojos (absorción de calor).

Los rayos son mantenidos en el núcleo debido a que el índice de refracción disminuye cuando aumenta la distancia desde el centro de una imaginaria sección transversal del núcleo de la fibra.

Por esto el índice de refracción puede disminuir por pasos, como en la fibra con índice escalonado o hacerlo gradualmente como en la fibra con índice gradual.

## Criterios de clasificación de las fibras

Una fibra se puede clasificar según el modo de propagación de los rayos en el interior del núcleo en (Fig.8):

1. Fibra monomodo: tiene un sólo modo de propagación de los rayos en el interior del núcleo que es paralelo al eje de la fibra.
2. Fibra multimodo: la luz se propaga por múltiples modos, que siguen diferentes caminos.

Según la variación del índice de refracción en el núcleo en:

1. Fibra con salto de índice (step index): el índice de refracción del núcleo se mantiene constante, al variar la distancia desde el centro de la fibra hacia el exterior.
2. Fibra con variación de índice (graded index): el índice de refracción varía, al aumentar la distancia desde el centro de la fibra hacia el exterior.

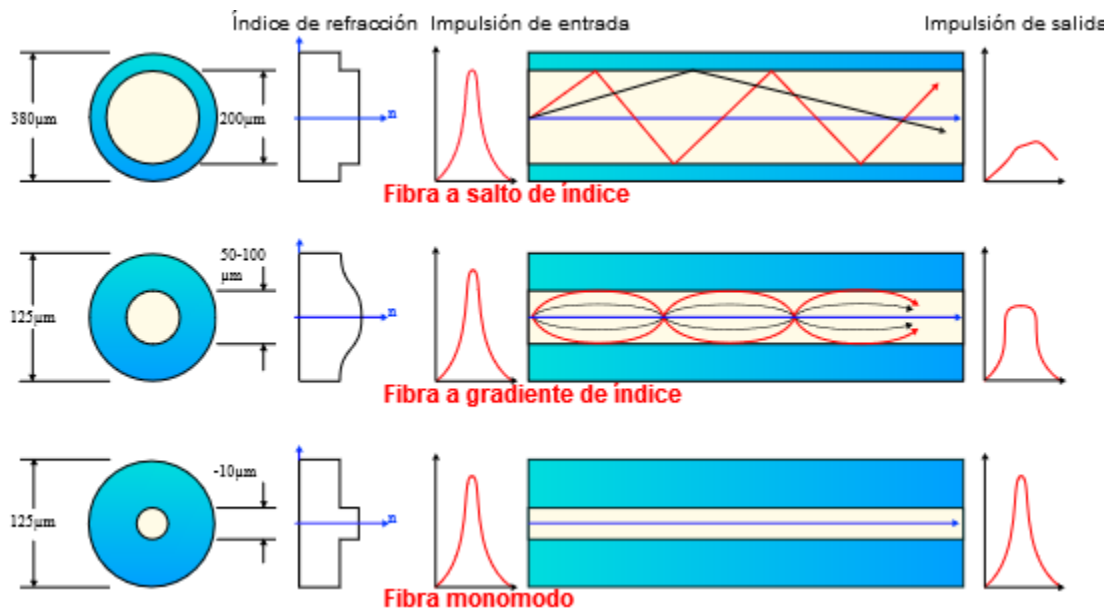


Figura 8 – Clasificación de fibras



## **2.3. Sensores de fibra óptica**

### **2.3.1. Definición y características generales**

El gran desarrollo de los sensores de fibra óptica tiene su origen en los grandes avances del sector de las comunicaciones ópticas. Estos sensores han permitido dar solución a varias de las limitaciones que presentan los sensores convencionales. Se puede definir un sensor de fibra óptica como aquel dispositivo que basado en fibra óptica es capaz de manifestar los cambios de una magnitud a medir mediante la modulación de al menos un parámetro de la luz que viaja por dicha fibra.

Las ventajas que proporcionan este tipo de sensores son numerosas, y entre ellas podemos encontrar su ligero peso y pequeño tamaño, inmunidad frente a las radiaciones electromagnéticas, gran resolución y sensibilidad, y su idoneidad para ser multiplexados. Por el contrario, su mayor inconveniente sea seguramente su elevado coste comparado con sensores mejor establecidos en el mercado. En función del parámetro modulado se podría hacer una primera clasificación de los sensores de fibra óptica:

- Sensores basados en la modulación de la intensidad: el valor de la intensidad de luz en la fibra varía debido a diferentes mecanismos (curvatura, atenuación,) en función de la magnitud a medir. Son simples y de coste reducido lo que les confiere un gran potencial comercial.
- Sensores basados en la modulación de la longitud de onda: probablemente los más utilizados dentro de esta categoría sean los basados en redes de Bragg. Entre sus grandes ventajas están la facilidad para ser multiplexados en una sola fibra e instalados en grandes estructuras (incluso embebidos en ellas) permitiendo la monitorización de su salud estructural.
- Sensores basados en la modulación de la fase: el cambio de fase de la luz es detectado mediante métodos interferométricos en los que se hacen coincidir dos haces de luz, uno de referencia y otro expuesto a la magnitud a medir. Mach-Zehnder, Michelson, Fabry-Perot y Sagnac son los interferómetros más

comúnmente utilizados. Su mayor ventaja es la gran sensibilidad que son capaces de proporcionar.

- Sensores basados en la modulación de la polarización: cambios en la tensión de la fibra u otros efectos, como el efecto Faraday, modifican el estado de la polarización en el interior de la fibra, permitiendo la medida de magnitudes como la presión, corriente eléctrica, etc.
- Sensores basados en la modulación del espectro: los sensores basados en espectroscopía analizan las variaciones del espectro transmitido para obtener la medida deseada.
- Sensores basados en la modulación de la luz dispersada: la amplitud y la localización de la magnitud a medir puede ser determinada por el estudio de la luz dispersada por cualquier fenómeno de dispersión (Rayleigh, Raman o Brillouin).

Otra posible clasificación de los sensores de fibra óptica se puede realizar de acuerdo a la magnitud a medir. Así, encontramos sensores que miden magnitudes mecánicas, térmicas, electromagnéticas, radiación, composición química, flujos de fluidos o aquellas propias de ámbito biomédico.

Si se atiende a la distribución espacial de los sensores, estos se pueden clasificar en tres categorías diferentes. Los sensores de punto proporcionan la medida en un punto discreto. Los sensores distribuidos son capaces de proporcionar el valor de la magnitud a medir a lo largo de toda la fibra óptica. Entre ambos se encuentran los cuasi-distribuidos, que se obtienen por la multiplexación de varios sensores de punto a lo largo de la misma fibra.

La estructura general para un sistema de medición basado en un sensor de fibra óptica se encuentra representada en la Figura 9. Sus cinco elementos fundamentales serían la fuente luz (normalmente un láser o LED), la fibra óptica que sirve como canal de transmisión de la luz, el elemento transductor encargado de modular las propiedades de la luz, un detector óptico y finalmente toda la electrónica correspondiente al procesamiento de la información.

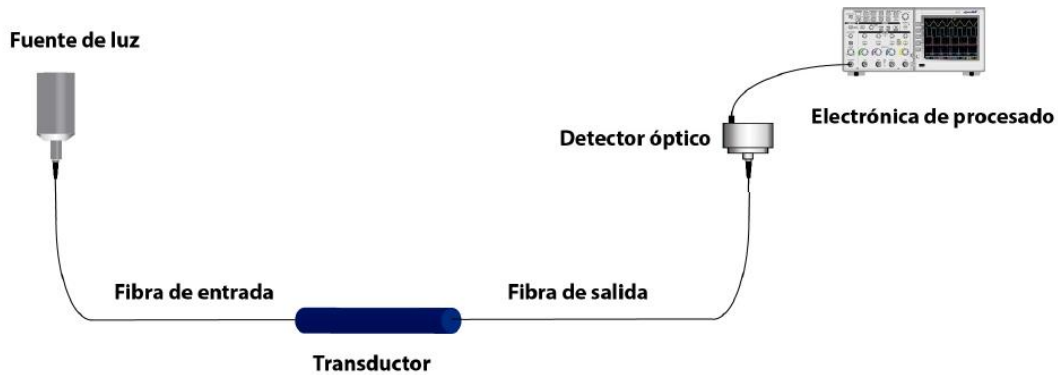


Figura 9 - Estructura general de un sistema de medición basado en un sensor de fibra óptica.

## 2.4. Internet de las cosas

Las recientes innovaciones de la electrónica, la informática y las tecnologías de la información y la comunicación (TIC) han propiciado, de una parte, un crecimiento exponencial de la capacidad material de procesamiento de los sistemas de tratamiento de información y, de otra parte, han permitido la miniaturización de microprocesadores que son empleados como sensores para la captación de datos. Todo acompañado, en paralelo, de una gradual reducción sustancial de los costes de fabricación y comercialización de dicho hardware. Estas invenciones, además, se construyen y amplifican mutuamente en una convergencia de tecnologías a través de los mundos físico y digital. (Barrio, 2018)

El Internet de las Cosas (IoT), también denominado por algunos Internet de los Objetos (IO) va más allá de la comunicación entre las personas, o entre las personas y el contenido digital, ahora se extiende a miles de millones de objetos cotidianos.

Consiste en la integración de sensores y dispositivos en objetos cotidianos que quedan conectados a Internet a través de redes fijas e inalámbricas. El hecho de que Internet esté presente al mismo tiempo en todas partes permite que la adopción masiva de esta tecnología sea más factible.

De esta manera, cualquier objeto es susceptible de ser conectado y manifestarse en la Red. Además, el IoT implica que todo objeto puede ser una fuente de datos. Esto está

empezando a transformar la forma de hacer negocios, la organización del sector público y el día a día de millones de personas.

La mayor parte de aplicaciones del Internet de las Cosas se ha producido fuera del ámbito doméstico. La gran mayoría de usos del IoT se encuentran en casi todos los sectores industriales.

- Empresas de transporte y navegación utilizan IoT para el seguimiento de vehículos y paquetes.
- Los sistemas médicos se basan en monitorización, diagnóstico y tratamiento conectados a Internet.
- Las empresas manufactureras emplean IoT para el manejo de materiales, la optimización de inventarios y la conexión de sistemas robóticos.
- Los servicios públicos municipales, el alumbrado, los sistemas de control del tráfico y otras aplicaciones inteligentes de las *smart cities* forman ahora parte del Internet de las Cosas.

Internet de las cosas (IoT) es una tendencia actual, en la cual una gran cantidad de equipos o dispositivos están conectados a internet. Éstos se comunican por medio de plataformas y servicios en la nube. Estas plataformas analizan y procesan los datos. Las soluciones IoT son aplicadas a muchas industrias como monitoreo de variables ambientales, monitoreo de pacientes, análisis de tráfico, monitoreos industriales, etc. (LTDA., 2018)

# Capítulo 3. MATERIALES E INFORMACIÓN

## 3.1. Recopilación de información

### 3.1.1. Arduino Ethernet

Arduino Ethernet es una placa basada en microcontrolador Atmega328 (fig.10). Cuenta con 14 pines de entrada o salida digital, 6 pines de entrada analógica, un cristal oscilador de 16MHz, Jack de alimentación, un cabezal ICSP y un botón de reset. El Ethernet difiere de otras tarjetas en que no tiene un chip driver USB-a-serial incorporado, pero tiene una interface Wiznet Ethernet. Esta es la misma interface encontrada en los dispositivos Ethernet Shield. Incorpora un lector de tarjetas microSD, que puede ser usado para guardar archivos que sirvan en la red, accesibles a través de la librería SD.



Figura 10 - Arduino Ethernet

### **Características**

El ATmega328 tiene 32KB (con 0.5KB usados por el gestor de arranque). También tiene 2KB de SRAM y 1KB de EEPROM (que se puede leer y escribir con la librería EEPROM).

Cada uno de los 14 pines digitales de la tarjeta Ethernet pueden ser utilizados como salida o entrada. Operan a 5V. Cada pin puede proveer o recibir un máximo de 40mA y tiene un resistor pull-up interno (desconectado por defecto) de 20-50 Kohm. Adicionalmente, algunos pines tienen funciones especializadas

- Serial: 0 (RX) y 1 (TX). Usado para recibir (RX) y transmitir (TX) datos seriales TTL.
- Interrupciones externas 2 y 3. Estos pines pueden ser configurados para disparar una interrupción a un valor bajo, un canto de subida o bajada o un cambio de valor.
- PWM: 3,5,6,9 y 10. Proveen una salida PWM de 8-bit.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Soportan comunicación SPI usando librería SPI.
- LED:9. hay un LED incorporado conectado al pin digital 9.
- La tarjeta Ethernet tiene 6 salidas análogas, etiquetadas de A0 hasta A5, cada una de las cuales provee 10 bits de resolución (1024 diferentes valores). Miden desde tierra a 5 volts.
- TWI: A4 (SDA) y A5 (SCL). Soporta comunicación TWI usando la librería Wire.
- AREF, voltaje de referencia para entradas análogas.
- Reset. Lleva esta línea a BAJO para resetear el microcontrolador.
- Largo 2.7 y ancho 2.1 pulgadas respectivamente, conector RJ45 y conector Jack que se extienden fuera de este rango.

### **Comunicación**

El ATmega328 también soporta comunicación TWI y SPI. El software Arduino incluye una librería Wire para simplificar el uso del bus TWI. Para la comunicación SPI, usar la librería SPI.

Esta tarjeta también se puede conectar a una red de cables vía Ethernet. Cuando se conecta a una red, necesitarás darle una dirección IP y una dirección MAC. La librería Ethernet es totalmente compatible.

El lector de tarjeta microSD incluido se puede acceder vía librería SD. Cuando se trabaja con esta librería, el SS está en el pin 4.

## ***Programación***

Es posible programar la tarjeta Arduino Ethernet de 2 formas: a través del cabezal de programación de 6 pines, o con un programador externo ISP.

El cabezal de programación serial de 6 pines es compatible con los cables USB FTDI y las tarjetas de conexión USB-a-serial básicas incluyendo el conector Arduino USB-serial. Incluye soporte para reseteo automático, lo que permite subir los bocetos sin presionar el botón de reseteo de la tarjeta. Cuando se conecta a un adaptador USB FTDI, el Ethernet Arduino se energiza desde el adaptador.

También puedes programar la tarjeta Ethernet con un programador externo como el AVRISP mkII o el USBTinyISP.

### **3.1.2. ESP8266**

El ESP8266 es un SoC (system on chip) fabricado por la compañía china Espressif. Este SoC agrupa distintos componentes en un mismo integrado (ver fig.11).

El ESP8266 es un chip altamente integrado diseñado para las necesidades de un nuevo mundo conectado. Ofrece una solución completa y autónoma de redes Wi-Fi, lo que le permite alojar la aplicación o servir como puente entre Internet y un microcontrolador.

El ESP8266 tiene potentes capacidades a bordo de procesamiento y almacenamiento que le permiten integrarse con sensores y dispositivos específicos de aplicación a través de sus GPIOs con un desarrollo mínimo y carga mínima durante el tiempo de ejecución. Su alto grado de integración en el chip permite una circuitería externa mínima, y la totalidad de la solución, incluyendo el módulo está diseñado para ocupar el área mínima en un PCB.

El ESP8266 no incorpora memoria Flash dentro del SoC, por lo que tiene que ser proporcionada por el módulo en el que se monta. La conexión entre la memoria se realiza por QSPI pero, de normal, su uso es transparente para nosotros.



Figura 11 - ESP8266

## Características

(las figuras 12 muestran ciertas características físicas).

- WIFI @ 2.4 GHz, soporta WPA / WPA2
- Modulo ultra-pequeño. Tamaño 11.5mm \* 11.5mm
- ADC integrado de 10 bit de precisión
- Protocolo stack TCP/IP integrado
- Switch TR integrado, balun, LNA, amplificador de potencia y red de adaptación
- PLL integrado, regulador de tensión y componentes de administración de energía
- Potencia de salida: 802.11b mode + 19.5dBm
- Soporta diversidad de antenas
- Corriente de fuga menor que 10uA
- CPU de 32 bits de baja potencia: puede funcionar como procesador de aplicaciones.
- SDIO 2.0, SPI, UART
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & agregación A-MSDU & 0.4ms de intervalo de guarda
- Encendido y transferencia de paquetes < 2ms.
- El consumo de energía en espera es de menos de 1.0mW (DTIM3)



- Rango de temperatura de operación: -40 ~ 125 °C
- Voltaje de alimentación: 3.3 V. Este módulo no tolera 5 V. Cualquier voltaje superior a 3.6 V en sus entradas o en el pin de alimentación puede dañarlo.

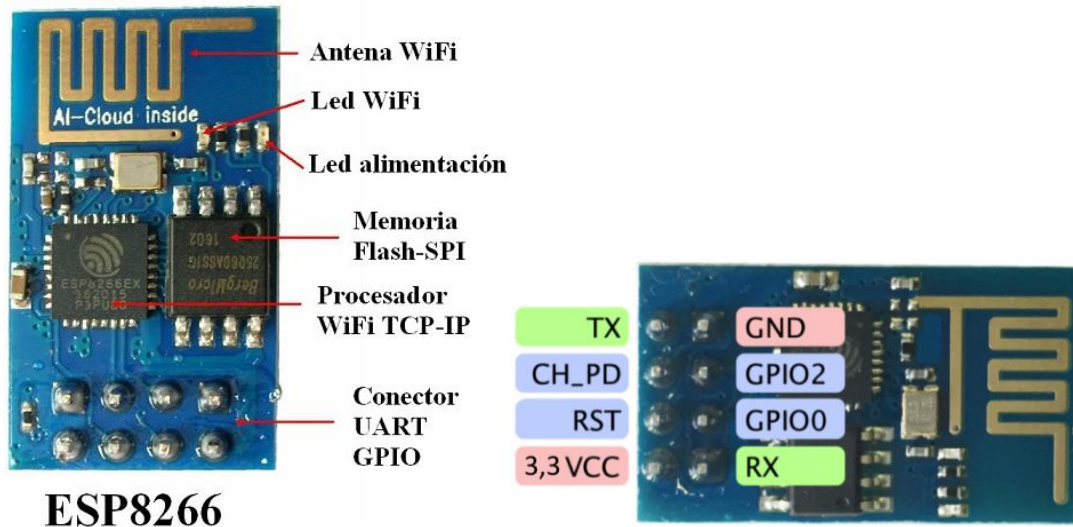


Figura 12- ESP8266 características físicas

## Historia

La historia del ESP8266 y su modelo superior ESP32 comienza en agosto de 2014 con la aparición de los módulos ESP01 de la mano del fabricante AI-Thinker.

La forma de comunicarse con el ESP8266 era a través de módulos AT, la documentación era escasa y en chino, y el SDK complejo y poco accesible, por lo que la utilidad era aún limitada. Pero eso no impidió que se viera su potencial y la comunidad y distintos fabricantes comenzaron a trabajar activamente en él.

Otro hito importante fue la aparición del firmware NodeMCU, que también da nombre a una placa de desarrollo. Este firmware permitía programar el ESP8266 con LUA, un lenguaje semicompilado basado en C y Perl.

El trabajo de la comunidad continuó generando de documentación, tutoriales y herramientas. Así llegamos al siguiente gran hito, la liberación por parte de la comunidad de SDK alternativos Open Source para el ESP8266 basados en la toolchain GCC.

Esto permitió la programación con el entorno de Arduino con el ESP8266 Arduino Core. Esto supuso la catapulta definitiva para el ESP8266 en el sector maker, y le permitió beneficiarse del apoyo de la enorme comunidad de amantes de Arduino.

Espressif reaccionó (o vio el potencial, o le dio por apoyar a la comunidad) y generó nuevos SDK con una licencia similar a la MIT, lo que en cierta forma dio un soporte a la comunidad de usuarios.

Desde entonces, una gran cantidad de fabricantes han generado placas de desarrollo que integran el ESP8266. Algunas de las más famosas, entre otras, son la NodeMCU y la WeMos, ambas con distintas variantes.

También tenemos distintos SDK y firmwares disponibles, que permiten programar los módulos con ESP8266 en diferentes lenguajes. Así, tenemos MicroPython (lenguaje Python) ESPruino (Javascript), ESP-OPEN-ROT (basado en FreeRTOS) o Mongoose OS, entre otras opciones.

En septiembre de 2016 se lanzó el ESP32, un modelo muy superior que solucionaba algunas carencias del ESP8266.

### ***Operación***

Existen dos modos de operación y se configuran a través de los pines GPIO0 y GPIO2: el modo **Flash** y el modo **UART** (ver fig.13).

- Modo UART, carga de programa en la memoria del ESP-01

Cuando queremos cargar un programa en el ESP-01 debemos encenderlo o resetearlo teniendo el pin GPIO0 a nivel bajo (LOW = 0 V = GND) y el GPIO2 a nivel alto (HIGH = 3,3 V = Vcc). El ESP8266 trabaja con niveles lógicos de 3,3 V. El pin GPIO2 está por defecto a HIGH, ya que tiene un pull-up interno, por lo que podemos dejarlo simplemente desconectado.

- Modo Flash: ejecución de programa en el ESP-01

Una vez cargado el programa hay que ejecutarlo: esto lo conseguimos teniendo el GPIO0 y el GPIO2 a nivel alto (3,3 V). Tanto el GPIO0 como el GPIO2 están por defecto a HIGH, ya que ambos tienen un pull-up interno, por lo que podemos dejarlos simplemente desconectados.

MODOS				
	GPIO0	GPIO2	CH_PD	RESET
Carga programa	LOW	HIGH	HIGH	HIGH
Ejecuta programa	HIGH	HIGH	HIGH	HIGH

Figura 13 - Modos de operación

- RX y TX se utilizan para cargar el programa. Al finalizar la carga se pueden utilizar como pines de entrada y salida digitales.
- Los modos de trabajo se indican a través de los pines 0 y 2 cuando se resetea o reinicia la placa. Una vez que el modo de ejecución esté funcionando (normal), ya puede conectarse cualquier componente a estos pines.

### **Programación**

El entorno de desarrollo a utilizar es el entorno de desarrollo de Arduino. Al seguir los siguientes pasos entonces estaremos listos para programar desde el compilador de Arduino (figuras 14-18).

#### 1. Añadir la URL para placas adicionales

- ✓ Abrir “preferencias”. Se encuentran en *Archivo > Preferencias*

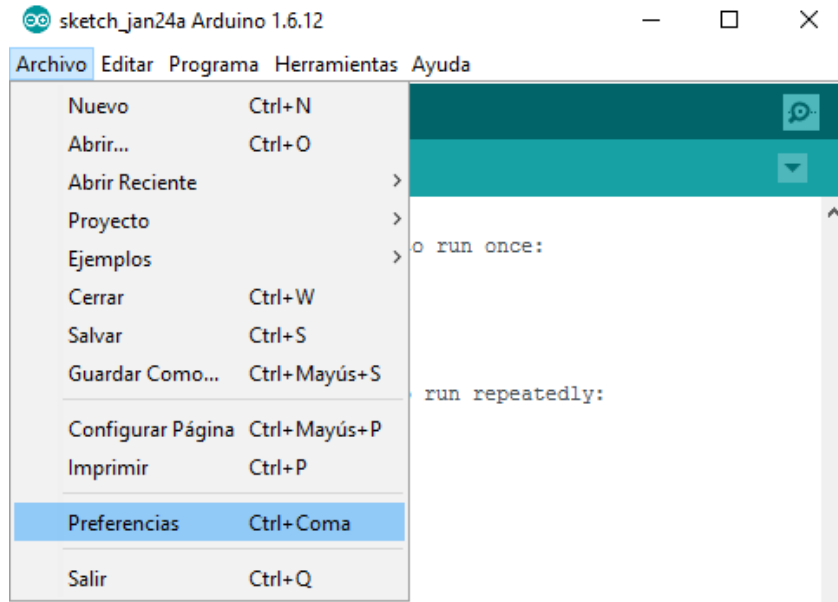


Figura 14 - URLs Placas Adicionales

✓ En “Gestor de URLs Adicionales de Tarjetas” copiar el siguiente enlace:

❖ [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json).

✓ Dar click en “OK”.

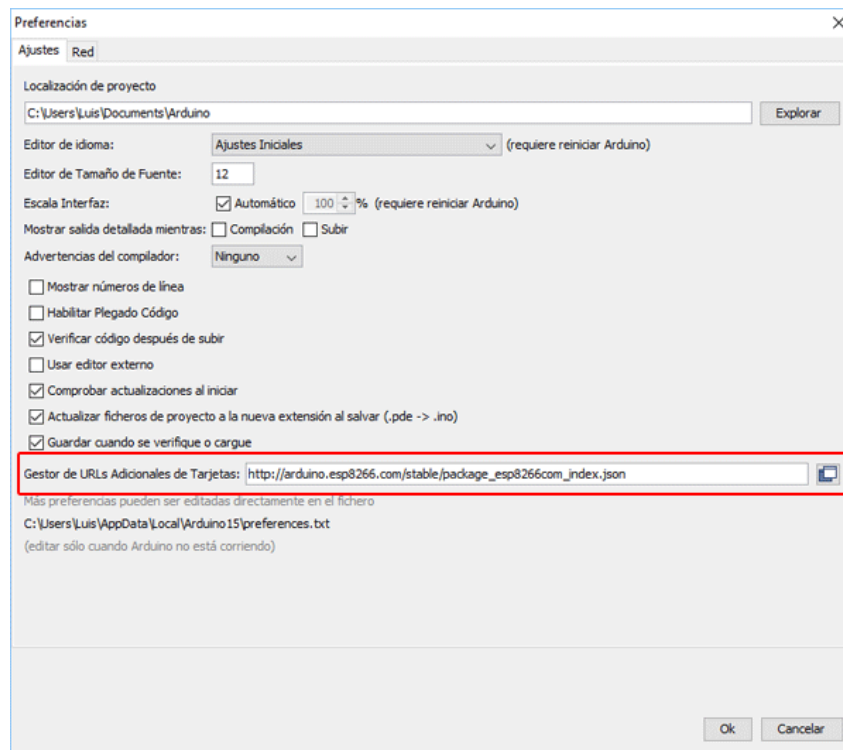


Figura 15 - URLs Placas Adicionales

## 2. Añadir los drivers del ESP8266

- ✓ Ir a *Herramientas > Placa: "Arduino UNO" > Gestor de Tarjetas*

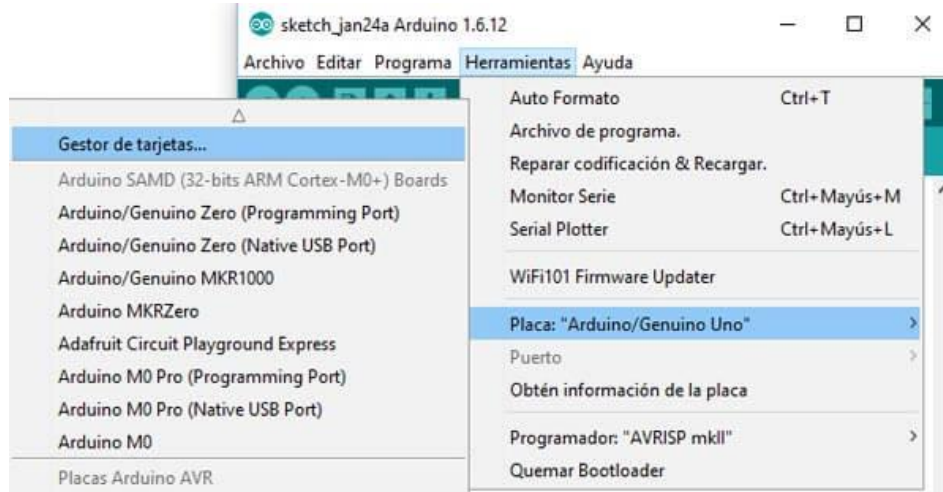


Figura 16 - Gestor de Tarjetas

- ✓ En gestor de tarjetas buscar "esp" e instalar "*ESP8266 by ESP8266 Community*".

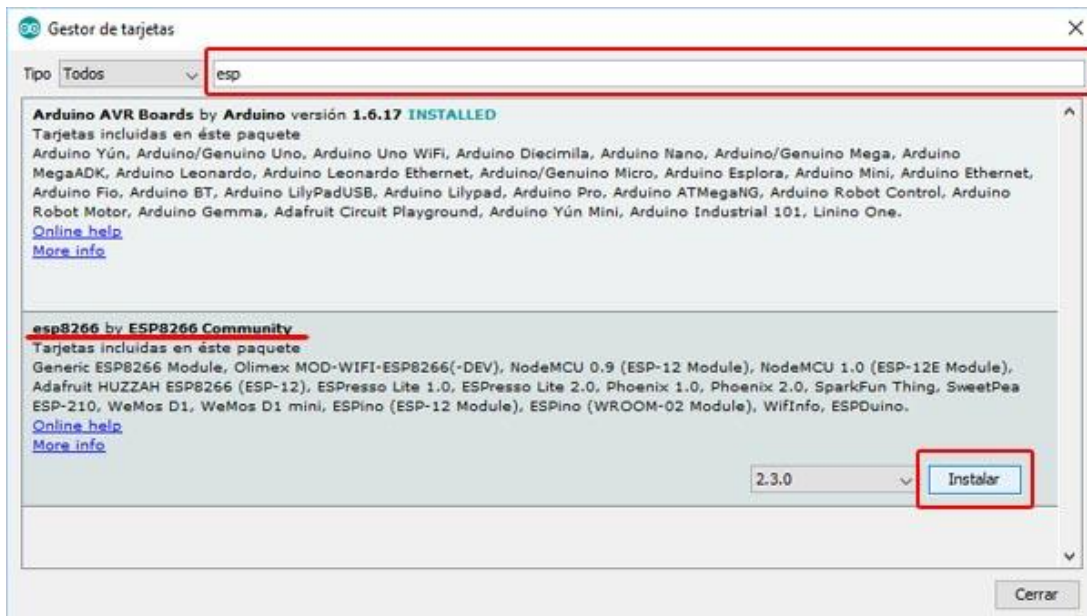


Figura 17 - Gestor de Tarjetas

## 3. Seleccionar la placa "Generic ESP8266 module".

- ✓ Ya instalado, podemos elegir entre los diferentes tipos de módulos ESP. Para utilizar el ESP-01 seleccionaremos “*Generic ESP8266 module*”.

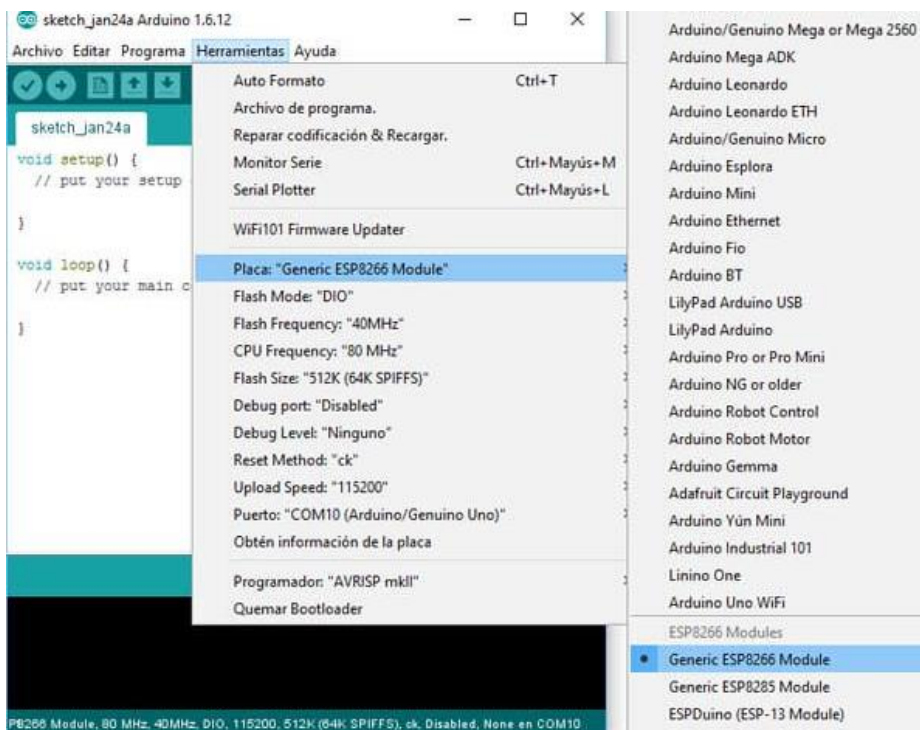


Figura 18 - Módulos ESP

### **Versiones del ESP8266**

Existen distintos módulos que integran el SoC ESP8266 (ver fig.19). Las características principales de estos módulos son similares y, básicamente, se distinguen en la memoria Flash disponible y en su forma física, que a su vez condiciona el número de pines GPIO que tienen accesibles.

En algunas placas (las menos) los GPIO tienen forma de pin, por lo que es posible soldar un cable, o conectar un terminal. Pero en la mayoría, los módulos tienen forma de “medio pin”, ya que están pensado para integrarse (soldados) en PCBs o placas de desarrollo.



Figura 19 - Módulos con ESP8266

### 3.1.3. NodeMCU

NodeMCU es un nombre que recoge tanto a un firmware Open Source y como a una placa de desarrollo basados en el ESP8266. 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.

En la figura 20 podemos observar que el NodeMCU es una tarjeta de desarrollo similar a Arduino, especialmente orientada al Internet de las cosas (IoT). Está basado en el SoC (System on Chip) **ESP8266**, un chip altamente integrado, diseñado para las necesidades de un mundo conectado. Integra un potente procesador con Arquitectura de 32 bits (más potente que el Arduino Due) y conectividad Wifi.

Para el desarrollo de aplicaciones se puede elegir entre los lenguajes Arduino y Lua. Al trabajar dentro del entorno Arduino podremos utilizar un lenguaje que ya conocemos y hacer uso de un IDE sencillo de utilizar, además de hacer uso de toda la información sobre proyectos y librerías disponibles en internet. La comunidad de usuarios de Arduino es muy activa y da soporte a plataformas como el ESP8266.

NodeMCU viene con un firmware preinstalado el cual nos permite trabajar con el lenguaje interpretado LUA, enviándole comandos mediante el puerto serial (CP2102). Las tarjetas NodeMCU y Wemos D1 mini son las plataformas más usadas en proyectos de Internet de las cosas (IoT). No compite con Arduino, pues cubren objetivos distintos, incluso es posible programar NodeMCU desde el IDE de Arduino.

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.

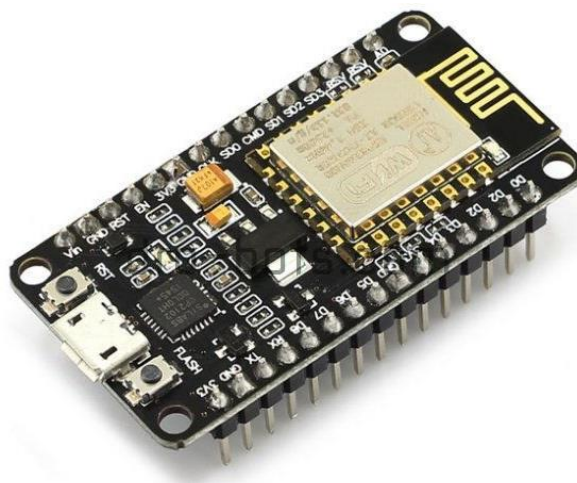


Figura 20 - NodeMCU



## **Características**

NodeMCU es capaz de conectarse vía wifi para recibir o transmitir información. Al mismo tiempo es capaz de interactuar con motores, leds, sensores de temperatura, detectores de humo, relés, etc.

- Voltaje de Alimentación (USB): 5V DC
- Voltaje de Entradas/Salidas: 3.3V DC
- SoC: ESP8266 (Módulo ESP-12)
- CPU: Tensilica Xtensa LX3 (32 bit)
- Frecuencia de Reloj: 80MHz/160MHz
- Instruction RAM: 32KB
- Data RAM: 96KB
- Memoria Flash Externa: 4MB
- 13 pines digitales numerados del D0 al D12
- 1 pin analógico numerado A0
- 3 pines de salida de 3,3V
- 1 pin de 5V (versión V3 2 pines 5V)
- 4 pines de tierra GND (versión V3 5 pines GND)
- 2 leds integrados. Asociados a los pines D0 (GPIO16) y D4 (GPIO2)
- 2 pulsadores FLASH y RESET (RST)
- Chip USB-Serial: CP2102
- Certificación FCC
- Antena en PCB
- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Stack de Protocolo TCP/IP integrado
- PLLs, reguladores, DCXO y manejo de poder integrados
- Potencia de salida de +19.5dBm en modo 802.11b
- Corriente de fuga menor a 10uA
- STBC, 1x1 MIMO, 2x1 MIMO

- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Consumo de potencia Standby < 1.0mW (DTIM3)
- Wake up and transmit packets in < 2ms
- Consumo de potencia Standby < 1.0mW (DTIM3)

## ***Historia***

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.

En los primeros momentos del ESP8266 apenas había información y la poca que había era confusa. 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. La programación en Lua permitía la conexión y programación del ESP8266 de una forma mucho más sencilla que las herramientas oficiales proporcionadas por Espressif.

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.

En 2015 el equipo de desarrollo original dejó de mantener el firmware de NodeMCU. Aunque sigue siendo mantenido por una comunidad de desarrolladores el interés en el

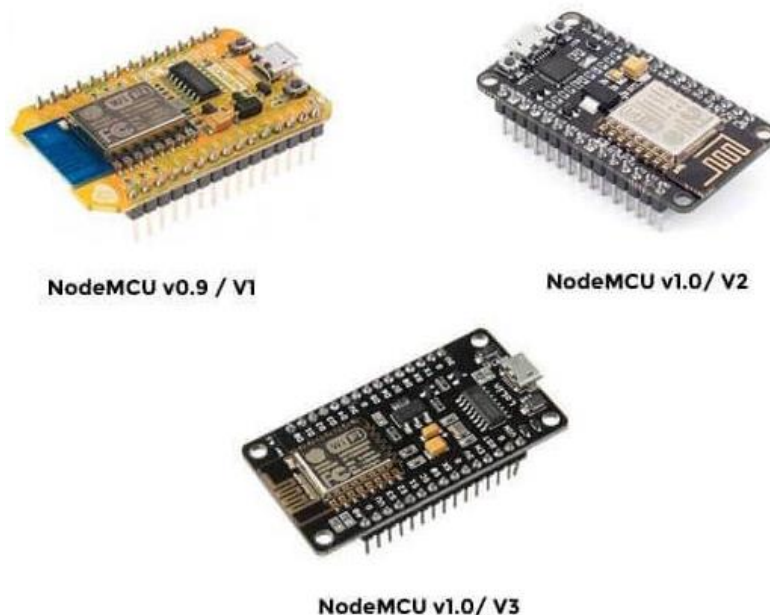
firmware ha caído casi por completo. Por este motivo, actualmente nos referimos con NodeMCU más a la placa de desarrollo que al firmware.

No obstante, aunque actualmente el firmware haya caído un poco en el olvido, no hay que olvidar la contribución que el proyecto NodeMCU, tanto firmware como placa de desarrollo, han supuesto para la proliferación e implantación del ESP8266.

### ***Versiones del NodeMCU***

Existen diferentes versiones de NodeMCU debido a que se trata de una placa de hardware abierto y cualquier fabricante puede crear su propia distribución (fig.21).

Pero 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. Las diferencias que vamos a encontrar son básicamente el número de pines a los que tenemos acceso y el tamaño de cada placa.



*Figura 21 - Versiones de NodeMCU*

Actualmente en el mercado existen 3 versiones (fig.22). 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	Común
1ª	v0.9	V1
2ª	v1.0	V2
2ª	v1.0	V3

*Figura 22 - Versiones NodeMCU*

La última columna refiere al nombre con el que se puede encontrar en tiendas online. 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.

Fabricantes principales:

- ✓ Amica es el distribuidor oficial
- ✓ DOIT
- ✓ Lolin / Wemos

Las placas de 1ª y 2ª generación utilizan el módulo ESP-12 con 4 MB de Flash, pero la 2ª generación utiliza la más nueva y mejorada ESP-12E. A continuación, se estudian un poco las características de las diferentes versiones.

### **1ª Generación / v0.9 / V1**



*Figura 23 - NodeMCU V1*

Quizá el tamaño es el mayor inconveniente que encontramos con esta placa. Es muy incómodo cuando se le conecta al protoboard, ya que no deja espacio suficiente para

conexiones de pines. El módulo que utiliza es el ESP-12 basado en el ESP8266 con una memoria Flash de 4 MB (fig.23).

## 2ª generación / v1.0 / V2

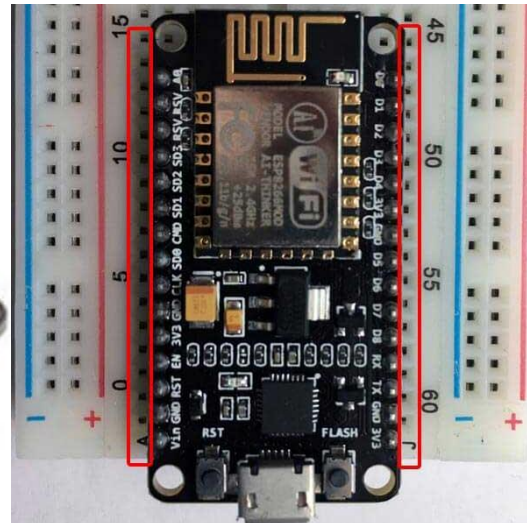


Figura 24 - NodeMCU V2

Esta 2ª generación sí encaja perfectamente en la protoboard y nos permite conectar cables a los pines de una forma sencilla en la figura 24 se aprecia el óptimo tamaño de esta placa.

Otra diferencia es que el chip fue actualizado del ESP-12 al ESP-12E y esto nos permite tener algunos pines extra.

“Con esta nueva generación de NodeMCU se produjo un salto sustancial de calidad y facilidad en la programación y prototipado. Es la placa recomendada y con la que vamos a trabajar en este tutorial”. (Valle, 2017, s/p)

## 2ª generación / v1.0 / V3



Figura 25 - NodeMCU V3

No es una especificación oficial por parte de NodeMCU. Se trata de una versión creada por Lolin. Aporta ciertas mejoras como un puerto USB más robusto y dos de los pines que no se utilizan en la versión V2 se utilizan como salida de 5V directa del USB y un GND adicional.

La V3 es un poco más ancha que la V2 (fig. 25). Aun así, es recomendable seguir las especificaciones de NodeMCU V2 ya que es la última versión oficial.

### **Acceso a pines**

Una vez que hemos visto qué es el NodeMCU y las versiones que existen, vamos a empezar describiendo las características más importantes que nos aporta esta placa a nivel de hardware, los pines digitales, analógicos y de alimentación mostrados en la figura 26.

En la siguiente imagen puedes ver una visión general de todos los pines de NodeMCU V2. Este diagrama de pines de NodeMCU, equivaldrá a si ponemos la placa con el puerto USB hacia abajo como se muestra en la siguiente imagen.

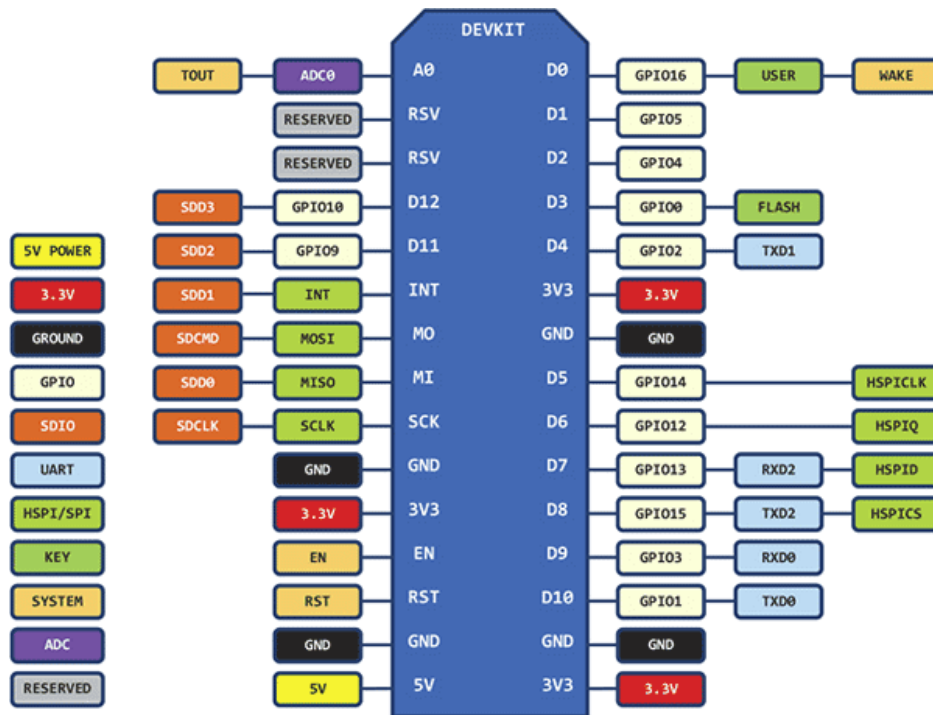


Figura 26 - Diagrama de pines NodeMCU V2

## Pines digitales

Los pines digitales de NodeMCU van numerados del D0 al D12 y en la figura 27 podemos apreciarlos. Existen dos nomenclaturas para nombrar los pines, los que aparecen en la placa escritos y en ocasiones verás el nombre asociado a cada Dx con GPIOx.

El GPIOx indica a qué pin del propio ESP8266 está conectado esa patilla. Pines digitales útiles de NodeMCU

El ESP8266 no tiene memoria flash y por lo tanto necesita de una memoria externa en la que almacenar los programas y datos. Para conectarse a ella, necesita utilizar pines del propio ESP8266.

Debido a esto, alguno de los pines de NodeMCU no pueden ser utilizados. Dependerá de la versión y el fabricante con el que estemos trabajando, lo recomendable es no utilizar los pines GPIO6, GPIO7, GPIO8, GPIO9 (D11), GPIO10 (D12) y GPIO11.

Solo dos de ellos son accesibles a través de la placa NodeMCU, el pin GPIO9 (D11) y el pin GPIO10 (D12).

De los pines restantes es aconsejable no utilizar tampoco los pines D9 y D10 correspondientes al Rx (recibir) y Tx (transmitir).

Estos pines se utilizan para transmitir un programa o sketch o para la comunicación entre el NodeMCU y el PC a través del puerto serie. Si se utilizan pueden ocasionar interferencias a la hora de cargar un programa. Por lo tanto, nos quedarían los pines del D0 al D8, 9 pines digitales de entrada y salida.

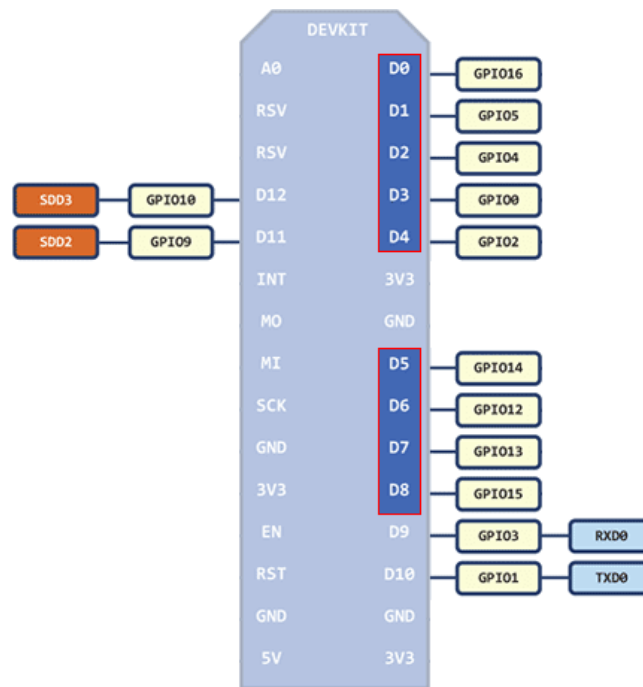


Figura 27 - Diagrama de pines digitales

## Pin análogo

En el pin analógico vamos a poder tener un rango de valores. Este rango vendrá determinado por la resolución del convertor ADC (del inglés Analog Digital Converter) Conversor Analógico Digital. El NodeMCU tiene solo un pin analógico que admite un rango de valores de 0 a 3,3V con una resolución de 10-bit. Esto implica que dentro del código tendremos un valor entre 0 y 1023 que se mapea con el voltaje entre 0 y 3,3V (ver fig.28).



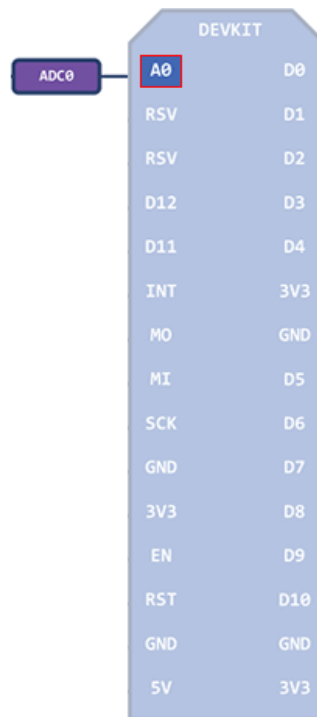


Figura 28 - Pin analogo

## Pines de alimentación

Los pines de alimentación tienen 2 funciones:

- Alimentar sensores y componentes (salida)
- Alimentar la propia placa (entrada)

El voltaje de operación de NodeMCU es de 3,3V y, por lo tanto, en principio no podríamos alimentar ningún componente que necesitara 5V.

Cuando se alimenta a través del puerto USB con 5V, internamente tiene un regulador de voltaje que saca 3,3V y 5V. Los 3,3V se utilizan para alimentar el NodeMCU (electrónica) y para sacarlo por los 3 pines marcados con ese valor.

Los 5V se utilizan para alimentar otros componentes dentro de la placa y para sacarlos por el pin de 5V.

Además, por cualquiera de estos pines podemos suministrar el mismo voltaje permitiendo así alimentar a la placa además del puerto USB. Eso sí, si alimentamos con 3,3V por alguno de los pines marcados con ese valor, la salida de 5V ya no nos suministrará esos 5V.

Esto hay que tenerlo en cuenta si en nuestro proyecto hay sensores que se alimentan con 5V.

## **Programación**

El entorno de desarrollo a utilizar es el entorno de desarrollo de Arduino. Al seguir los siguientes pasos entonces estaremos listos para programar desde el compilador de Arduino (podemos apreciar los pasos en las figuras 29-34).

### 1. Añadir la URL para placas adicionales

- ✓ Abrir “preferencias”. Se encuentran en *Archivo > Preferencias*

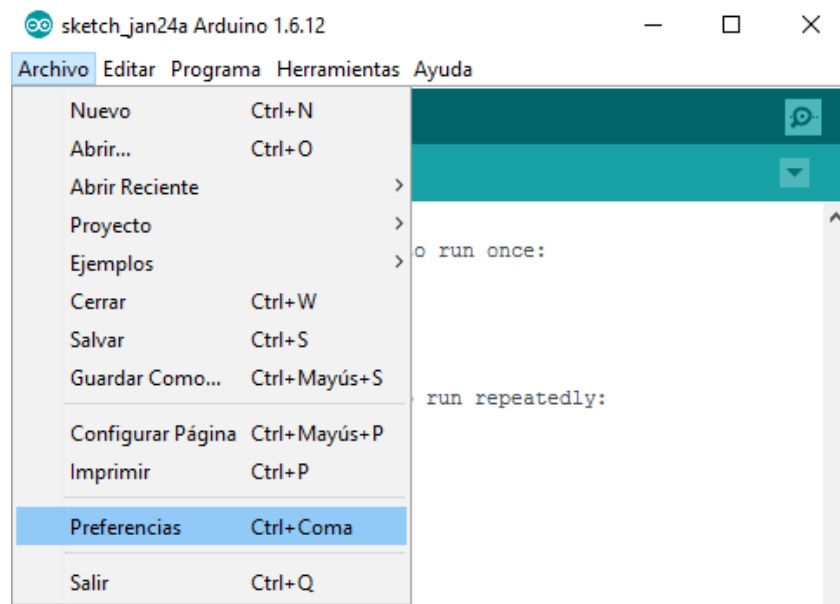


Figura 29 - URLs Placas Adicionales

- ✓ En “Gestor de URLs Adicionales de Tarjetas” copiar el siguiente enlace:

- ❖ [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json).

- ✓ Dar click en “OK”.

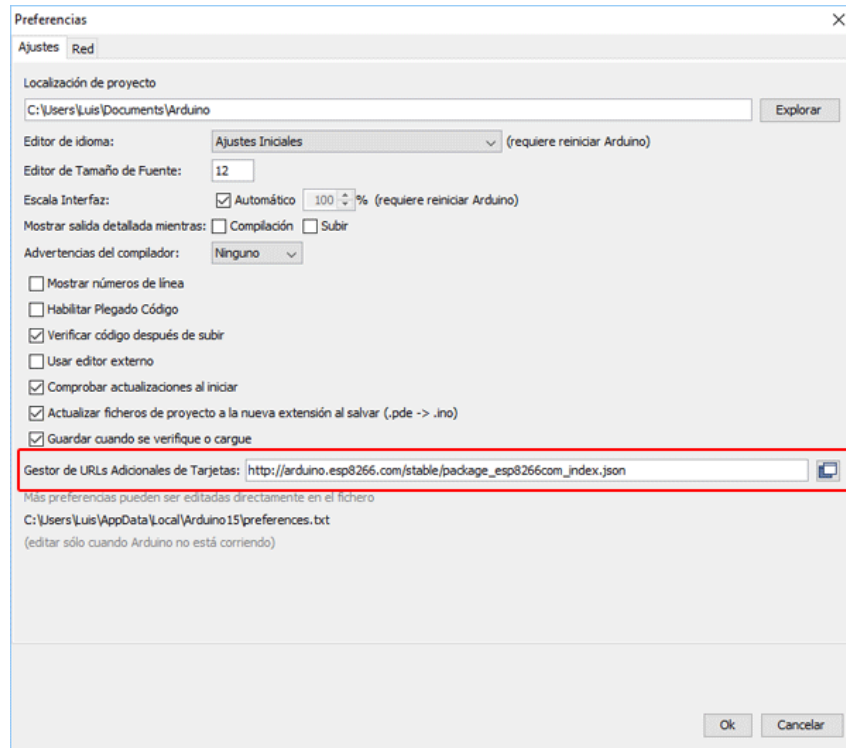


Figura 30 - URLs Placas Adicionales

## 2. Añadir los drivers del ESP8266

- ✓ Ir a *Herramientas* > *Placa: “Arduino UNO”* > *Gestor de Tarjetas*

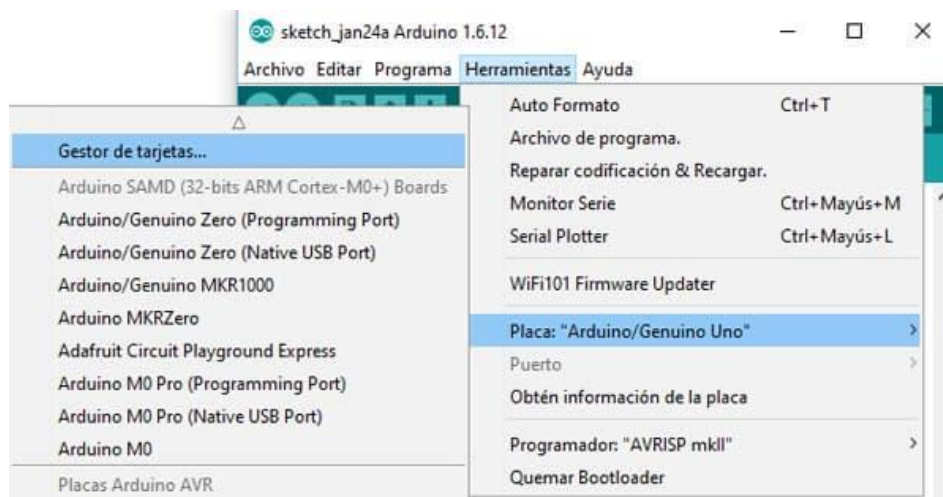


Figura 31 - Gestor de Tarjetas

- ✓ En gestor de tarjetas buscar “esp” e instalar “ESP8266 by ESP8266 Community”.

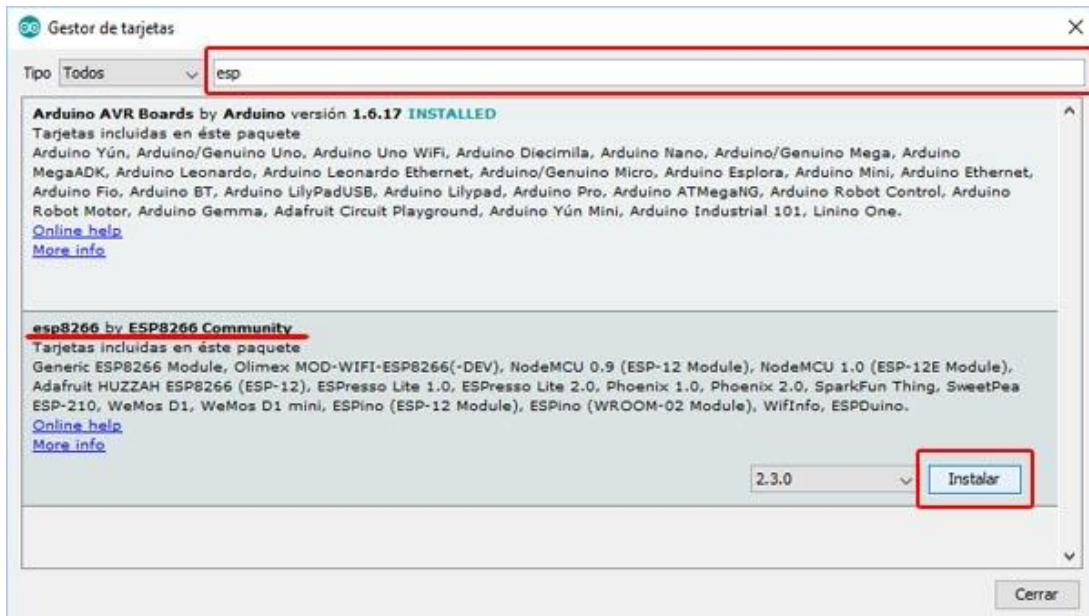


Figura 32 - Gestor de Tarjetas

### 3. Seleccionar la placa “NodeMCU 1.0(ESP-12E Module)”.

- ✓ Ya instalado, podemos elegir entre los diferentes tipos de módulos ESP. Para utilizar el NodeMCU seleccionaremos “NodeMCU 1.0(ESP-12E Module)”.

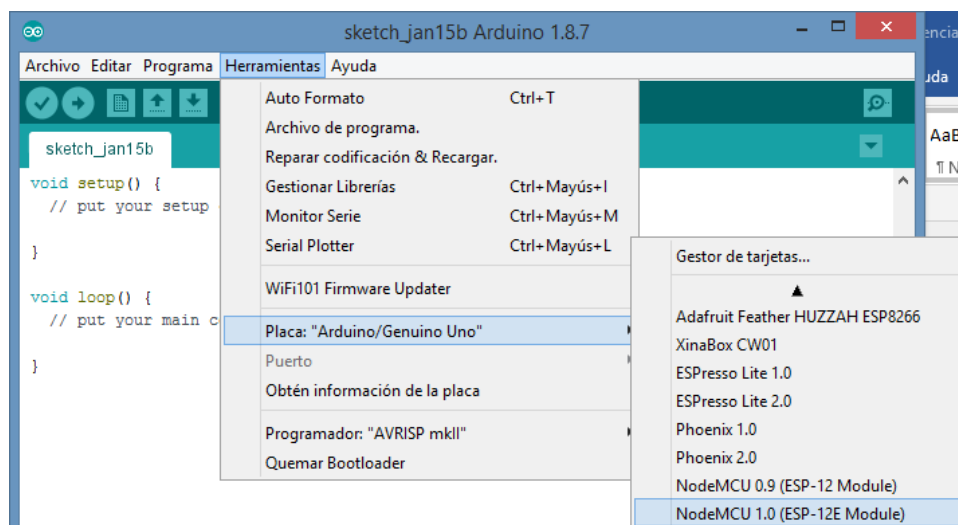


Figura 33 - Módulos ESP

- ✓ Ésta es la configuración recomendable para todas las NodeMCU. Para programar el IDE de Arduino cambiar la velocidad del monitor serial y la velocidad de carga a 115200 bps.

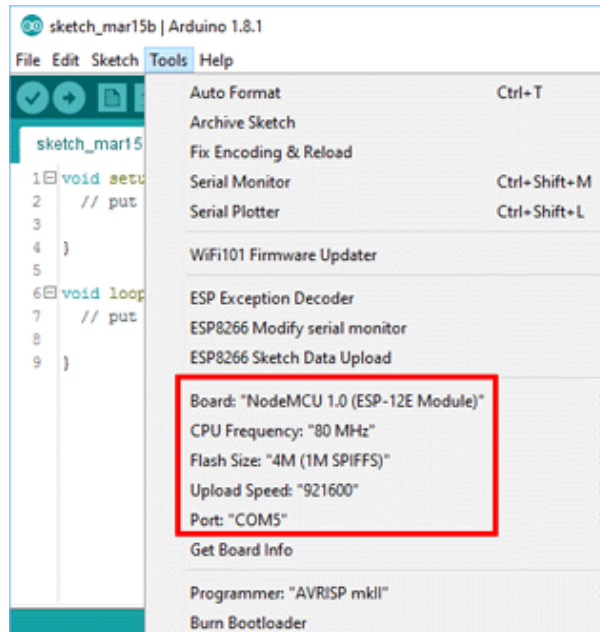


Figura 34 - Configuración NodeMCU

### 3.1.4. Selección de placa Arduino optima

En este punto es necesario realizar una elección de la cual dependerá mucho el proyecto. Luego de haber investigado y leído las características de distintas placas. Habiendo ya adquirido un panorama acerca de estos pequeños módulos: Arduino Ethernet, ESP8266 NodeMCU. Llega la etapa en la cual se deben evaluar las ventajas y desventajas mencionadas.

De manera unánime se descartó el uso del Arduino Ethernet pues esta placa ha quedado obsoleta por lo que su comercialización ha finalizado.

Es un juicio entonces que se debe emitir entre el ESP8266 y el NodeMCU.

Dadas las características de hardware y software se decidió utilizar el NodeMCU pues esta placa como se ha visto, es configurable como modulo wi-fi lo cual favorece en gran manera, pues por

el tamaño de los sensores se requiere su delicado manejo, así como un lugar específico donde puedan realizarse las pruebas pertinentes a la investigación. El ESP8266 pareciera sacar partida por su reducido tamaño, pero la falta de pines hace que sea descartado para las mediciones requeridas.

La versatilidad del NodeMCU permitirá ahorrar espacio y cableado, lo que favorecerá al proyecto pues se requiere lo más mínimo de señales de ruido o de corrientes parasitas, dado que las señales leídas rondan apenas los órdenes de los micros.

Entonces, es así como a grandes rasgos optamos y procedemos con el manejo de esta pequeña pero poderosa placa Arduino NodeMCU.

Cabe mencionar que, a pesar de este hecho, anexamos la prueba del ESP8266.

### **3.1.5. Servidores web**

#### ***Servidor***

En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor web, este término podría referirse a la máquina que almacena y maneja los sitios web, y en este sentido es utilizada por las compañías que ofrecen hosting o hospedaje.

Alternativamente, el servidor web podría referirse al software, como el servidor de http de Apache, que funciona en la máquina y maneja la entrega de los componentes de las páginas web como respuesta a peticiones de los navegadores de los clientes.

Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.

Un servidor sirve información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor.

### ***Servidor web***

Un servidor Web es un programa que sirve datos en forma de páginas Web, hipertextos o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo\*, concretamente del protocolo HTTP.

Con esto, un servidor Web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que solemos conocer como un navegador Web.

A modo de ejemplo: al teclear <http://www.cnice.mec.es> en un navegador, éste realizará una petición HTTP al servidor que tiene asociada dicha URL. El servidor responde al cliente enviando el código HTML de la página; el navegador cuando recibe el código, lo interpreta y lo muestra en pantalla.

El cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página. El servidor se encarga de transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

Sin embargo, el hecho de que HTTP y HTML estén íntima mente ligados no debe dar lugar a confundir ambos términos. HTML es un formato de archivo y HTTP es un protocolo.

Cabe destacar el hecho de que la palabra servidor identifica tanto al programa como a la máquina en la que dicho programa se ejecuta. Existe, por tanto, cierta ambigüedad en el término, aunque no será difícil diferenciar a cuál de los dos nos referimos en cada caso. En este artículo nos referiremos siempre a la aplicación.

Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear `www.wikipedia.org` en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección.

El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla.

Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

### ***Protocolo HTTP***

El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una página y la respuesta con el contenido. También sirve el protocolo para enviar información adicional en ambos sentidos, como formularios con campos de texto.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. Al analizar la transacción todos los datos se pierden.

Por esto se popularizaron las cookies, que son pequeños ficheros guardados en el propio ordenador que puede leer un sitio web al establecer conexión con él, y de esta forma reconocer a un visitante que ya estuvo en ese sitio anteriormente.



Gracias a esta identificación, el sitio web puede almacenar gran número de información sobre cada visitante, ofreciéndole así un mejor servicio.

El protocolo HTTP está basado en el modelo cliente-servidor.

Un cliente HTTP abre una conexión y envía su solicitud al servidor, el cual responderá con el recurso solicitado - si está disponible y su acceso es permitido - y la conexión se cierra.

### ***Navegador web***

Un navegador web o browser es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet.

Esta red de documentos es denominada World Wide Web (WWW). Los navegadores actuales permiten mostrar o ejecutar: gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces.

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados.

Los documentos pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos (un software servidor web). Tales documentos, comúnmente denominados páginas web, poseen hipervínculos que enlazan una porción de texto o una imagen a otro documento, normalmente relacionado con el texto o la imagen.

El seguimiento de enlaces de una página a otra, ubicada en cualquier computadora conectada a la Internet, se llama navegación; que es de donde se origina el nombre de navegador. Por otro lado, hojeador es una traducción literal del original en inglés, browser, aunque su uso es minoritario. Otra denominación es explorador web inspirada en uno de los navegadores más populares el Internet Explorer.

## **Funcionamiento de los navegadores**

La comunicación entre el servidor web y el navegador se realiza mediante el protocolo HTTP, aunque la mayoría de los navegadores soportan otros protocolos como FTP, Gopher, y HTTPS (una versión cifrada de HTTP basada en Secure Socket Layer o Capa de Conexión Segura (SSL)).

La función principal del navegador es descargar documentos HTML y mostrarlos en pantalla. En la actualidad, no solamente descargan este tipo de documentos, sino que muestran con el documento sus imágenes, sonidos e incluso vídeos (streaming) en diferentes formatos y protocolos. Además, permiten almacenar la información en el disco o crear marcadores (bookmarks) de las páginas más visitadas.

Algunos de los navegadores web más populares se incluyen en lo que se denomina una Suite. Estas Suite disponen de varios programas integrados para leer noticias de Usenet y correo electrónico mediante los protocolos NNTP, IMAP y POP.

Los primeros navegadores web sólo soportaban una versión muy simple de HTML. El rápido desarrollo de los navegadores web propietarios condujo al desarrollo de dialectos no estándares de HTML y a problemas de interoperabilidad en la web. Los más modernos (como Amaya, Mozilla, Opera y versiones recientes de Internet Explorer) soportan los estándares HTML y XHTML (comenzando con HTML 4.01, los cuales deberían visualizarse de la misma manera en todos ellos).

Los estándares web son publicados por el World Wide Web Consortium. [www.w3.org](http://www.w3.org)

## ***Alojamiento web - hosting***

El alojamiento web (en inglés web hosting) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía Web. Los Web Host son compañías que proporcionan espacio de un servidor a sus clientes.

## Tipos de Alojamiento Web

El alojamiento web se divide en seis tipos: gratuitos, compartidos, revendedores, servidores virtuales, servidores dedicados y de colocación.

- **Alojamiento gratuito:** El alojamiento gratuito es extremadamente limitado cuando se lo compara con el alojamiento de pago. Estos servicios generalmente agregan publicidad en los sitios y tienen un espacio y tráfico limitado.
- **Alojamiento compartido (shared hosting):** En este tipo de servicio se alojan clientes de varios sitios en un mismo servidor, gracias a la configuración del programa servidor web. Resulta una alternativa muy buena para pequeños y medianos clientes, es un servicio económico y tiene buen rendimiento.
- **Alojamiento de Imágenes:** Este tipo de hospedaje se ofrece para guardar tus imágenes en internet, la mayoría de estos servicios son gratuitos y las páginas se valen de la publicidad colocadas en su página al subir la imagen.
- **Alojamiento revendedor (reseller):** Este servicio de alojamiento está diseñado para grandes usuarios o personas que venden el servicio de hosting a otras personas. Estos paquetes cuentan con gran cantidad de espacio y de dominios disponibles para cada cuenta.
- **Servidores virtuales (VPS, Virtual Private Server):** mediante el uso de una máquina virtual, la empresa ofrece el control de un ordenador aparentemente no compartido. Así se pueden administrar varios dominios de forma fácil y económica, además de elegir los programas que se ejecutan en el servidor. Por ello, es el tipo de producto recomendado para empresas de diseño y programación web.
- **Servidores dedicados:** El término servidor dedicado se refiere a una forma avanzada de alojamiento web en la cual el cliente alquila o compra un ordenador completo, y por tanto tiene el control completo y la responsabilidad de administrarlo. El cuidado físico de la máquina y de la conectividad a Internet es tarea de la empresa de alojamiento, que suele tenerlo en un centro de datos.
- **Colocación (o housing):** Este servicio consiste básicamente en vender o alquilar un espacio físico de un centro de datos para que el cliente coloque ahí su propio

ordenador. La empresa le da la corriente y la conexión a Internet, pero el ordenador servidor lo elige completamente el usuario (hasta el hardware).

### ***Formas de obtenerlo***

Por lo general, se distingue entre servicios pagados y servicios gratuitos.

- **Servicios Pagados**

Este tipo de obtención, por lo general viene dado por el contrato de un proveedor de internet, el cual, junto con dar conexión, ofrece la posibilidad de almacenamiento mediante disco virtual o espacio web o combinación de ambos.

Otro medio de obtención es contratando algún servicio de una empresa no dependiente de la conexión a internet, las cuales ofrecen según las capacidades de sus servidores o de su espacio. Casi siempre a la par, entregan servicios añadidos, como la ejecución de tareas automáticas o cuentas de correo electrónico gratuitas. Normalmente las transacciones son electrónicas, por tarjeta de crédito o por sistemas de pagos (paypal).

- **Servicios Gratuitos**

Este tipo de servicio viene dado por la base de ser gratuito, y sin costo alguno al suscriptor. Sin embargo, quienes usan este servicio, por lo general son páginas de bajos recursos de mantención o aquellas que los dueños no poseen suficiente dinero para ser mantenida.

Como medio de financiamiento, el servidor coloca avisos de publicidad de AdSense (Google) u otras empresas, haciendo que la página se llene de publicidad en algún punto.

Otra limitación de estas ofertas es que tiene un espacio limitado y no se puede usar como almacén de datos, ni pueden alojar páginas subversivas o de contenido adulto o no permitido.

De todas maneras, existe una amplia oferta de alojamientos gratuitos con características muy diferentes y que pueden satisfacer las necesidades de programadores que desean

un lugar donde hacer pruebas o que mantienen una web con un número no muy elevado de visitas.

### 3.1.6. OPF372 y OPF472

#### **OPF372**

Los transmisores de fibra óptica de la familia OPF372 son dispositivos de alto rendimiento empaquetados para enlaces de comunicación de datos. El transmisor es un GaAlAs LED de 850 nm y está diseñado específicamente para lanzar energía óptica de manera eficiente a las fibras que varían en tamaño.

Desde fibras 50 / 125µm hasta 200 / 300µm de diámetro. Se ofrecen múltiples rangos de potencia con límites superior e inferior que permite al diseñador seleccionar el dispositivo más adecuado para la aplicación.

La combinación de características de este producto, incluida la alta velocidad y la potencia acoplada eficiente, lo convierte en un transmisor ideal para la Integración en todo tipo de equipos de comunicaciones de datos.

El diseño mecánico de este paquete está diseñado para montaje en placa o panel de PC junto con una arandela de bloqueo, mermelada. Tuerca, 2 tornillos # 2-56 y una tapa protectora contra el polvo como aparece en la figura 35.

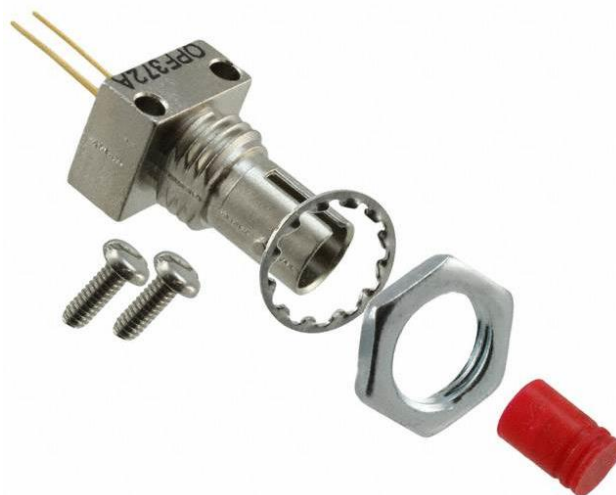


Figura 35 - OPF372A

## **Características**

- Tecnología LED de 850 nm a bajo costo.
- Conector popular tipo ST.
- Pre-probado con fibra para asegurar el rendimiento.
- Componente pre-montado y listo para usar.
- Operación a 35MHz.

## **Aplicaciones**

- Equipos industriales tipo Ethernet.
- Conversiones de medio Cobre – fibra.
- Enlaces de fibra óptica dentro de sistemas.
- Sistemas de videovigilancia.

## **Potencia típica de acople**

En la figura 36 podemos observar las potencias de acoples para esta familia.

Typical Coupled Power $I_f = 100\text{mA}, 25^\circ\text{C}$					
Fiber Size	Type	N.A.	OPF372A	OPF372B	OPF372D
50/125 $\mu\text{m}$	Graded Index	0.20	29 $\mu\text{W}$	19 $\mu\text{W}$	7.5 $\mu\text{W}$
62.5/125 $\mu\text{m}$	Graded Index	0.28	89 $\mu\text{W}$	51 $\mu\text{W}$	27 $\mu\text{W}$
100/140 $\mu\text{m}$	Graded Index	0.29	200 $\mu\text{W}$	129 $\mu\text{W}$	60 $\mu\text{W}$
200/300 $\mu\text{m}$	Step Index	0.41	750 $\mu\text{W}$	606 $\mu\text{W}$	320 $\mu\text{W}$

Figura 36 - Potencia de acople

## Estructura mecánica

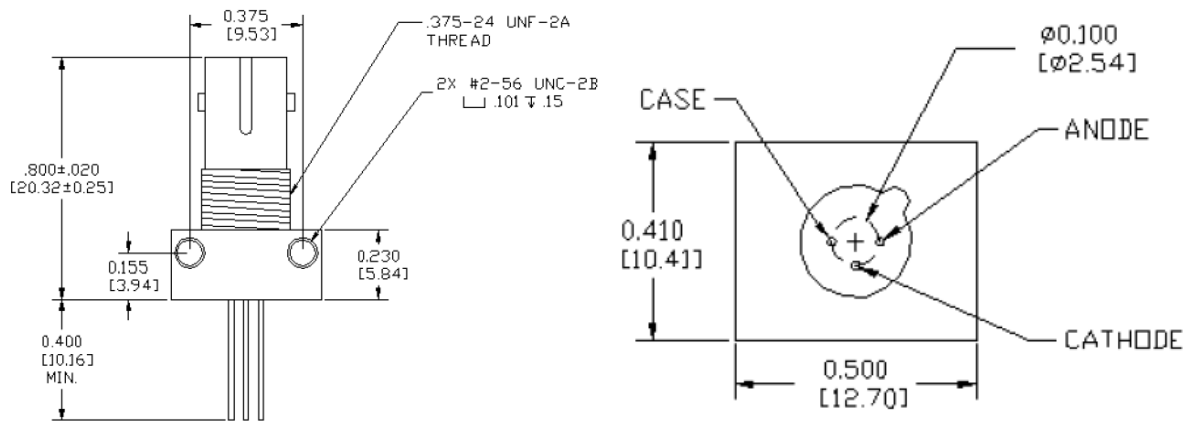


Figura 37 - Estructura física OPF372

Las dimensiones del dispositivo podemos observarlas en la fig. 37 y las medidas corresponden en pulgadas (milímetros).

## Especificaciones eléctricas

En la figura 38 se explican las especificaciones eléctricas.

### Electrical Specifications

Absolute Maximum Ratings ( $T_A = 25^\circ\text{C}$ unless otherwise noted)	
Storage Temperature Range	$-55^\circ\text{C}$ to $+100^\circ\text{C}$
Operating Temperature Range	$-40^\circ\text{C}$ to $+85^\circ\text{C}$
Lead Soldering Temperature <sup>(1)</sup>	$260^\circ\text{C}$
Continuous Forward Current <sup>(2)</sup>	100 mA
Maximum Reverse Voltage	1.0 V

Electrical Characteristics ( $T_A = 25^\circ\text{C}$ unless otherwise noted)							
SYMBOL	PARAMETER		MIN	TYP	MAX	UNITS	TEST CONDITIONS
$P_{oc}$	Total Coupled Power 50/125 $\mu\text{m}$ Fiber, NA = 0.20	OPF372A	25.0	29.0		$\mu\text{W}$	$I_F = 100\text{ mA}$
		OPF372B	15.0	19.0			
		OPF372D	5.0	7.5			
$V_F$	Forward Voltage		1.8	2.2		V	$I_F = 100\text{ mA}$
$V_R$	Reverse Voltage		1.8			V	$I_R = 100\ \mu\text{A}$
$\lambda$	Wavelength		830	850	870	nm	$I_F = 50\text{ mA}$
$\Delta\lambda$	Optical Bandwidth			50	60	nm	$I_F = 50\text{ mA}$
$t_r, t_f$	Rise and Fall Time			6.0	10.0	ns	$I_F = 100\text{ mA}$ ; 10% to 90% <sup>(3)</sup>

Figura 38 - Especificaciones eléctricas

## **OPF472**

El OPF472 es un fotodiodo PIN de silicio de bajo ruido montado en un paquete de bajo costo para aplicaciones de fibra óptica. Ofrece rápida respuesta, es de polarización moderada y es compatible con fuentes de diodo LED y láser en la región de longitud de onda de 800-1000 nm.

La baja capacitancia mejora el rendimiento de señal ante el ruido en aplicaciones LAN típicas de corto recorrido. El OPF472 está diseñado para ser compatible con fibras ópticas multimodo de 50/125 a 200/300 micras.

### ***Características***

- Alta velocidad – baja capacitancia.
- Conector popular tipo ST.
- Pre-probado con fibra para asegurar el rendimiento.
- Componente pre-montado y listo para usar.
- Operación a 35MHz.

### ***Aplicaciones***

- Equipos industriales tipo Ethernet.
- Conversiones de medio Cobre – fibra.
- Enlaces de fibra óptica dentro de sistemas.
- Sistemas de videovigilancia.



## Estructura mecánica

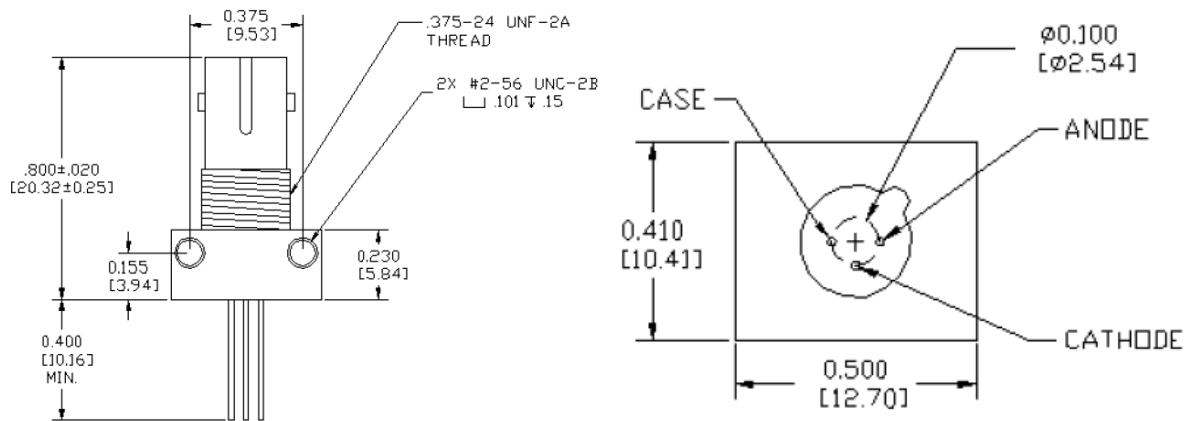


Figura 39 - Estructura física OPF472

Dimensiones en pulgadas (milímetros) Las dimensiones del dispositivo podemos observarlas en la Fig. 39 y las medidas UNC corresponden en pulgadas (milímetros).

## Especificaciones eléctricas

En la figura 40 se explican las especificaciones eléctricas.

### Electrical Specifications

Absolute Maximum Ratings ( $T_A = 25^\circ \text{C}$ unless otherwise noted)	
Storage Temperature Range	$-55^\circ \text{C}$ to $+100^\circ \text{C}$
Operating Temperature Range	$-40^\circ \text{C}$ to $+85^\circ \text{C}$
Lead Soldering Temperature <sup>(1)</sup>	$260^\circ \text{C}$
Continuous Power Dissipation <sup>(2)</sup>	200 mW
Maximum Reverse Voltage	100 VDC

Electrical Characteristics ( $T_A = 25^\circ \text{C}$ unless otherwise noted)						
SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	TEST CONDITIONS
R	Responsivity	0.45	0.55		A/W	$V_R = 5.0\text{V}$ ; 50/125 $\mu\text{m}$ fiber; $\lambda = 850\text{nm}$
$I_D$	Dark Current		0.1	5.0	nA	$V_R = 5.0\text{V}$
$\lambda_p$	Peak Response Wavelength		905		nm	
$t_r$	Output Rise Time		6.0		ns	$V_R = 15\text{V}$ ; $R_L = 50\Omega$ , 10%-90%
$C_T$	Total Capacitance		3.0		pF	$V_R = 20\text{V}$

Figura 40 - Especificaciones Eléctricas

## **CAPÍTULO 4. DESARROLLO DE PRUEBAS**

Conforme a la información recaudada en el capítulo anterior, en este nuevo capítulo partimos de ella para realizar las pruebas correspondientes a cada uno de los componentes necesarios en el diseño del proyecto.

A continuación, entonces veremos cómo se llevaron a cabo cada una de ellas.

### **4.1. Dispositivos OPF**

Los dispositivos OPF, los cuales sirvieron de fuente emisora de luz y dispositivo receptor de la misma a través de la fibra. Se necesitó del diseño electrónico para el posterior emplacado en placas fenólicas.

Para la realización del diseño electrónico o esquemático se utilizó el software PROTEUS. Con el cual también se realizaron el diseño del PCB necesario y una visualización en 3D.

#### ***PROTEUS***

Se trata de un completo programa que permite diseñar y simular circuitos electrónicos de forma práctica y accesible.

#### **Principales características**

- La aplicación ISIS permite generar circuitos reales, y comprobar su funcionamiento en un PCB (printed circuit board).
- Entorno de diseño gráfico de esquemas electrónicos fácil de utilizar y con efectivas herramientas.
- Entorno de simulación con la tecnología exclusiva de Proteus de modelación de sistemas virtuales (VSM).
- Herramienta ARES para el enrutado, ubicación y edición de componentes, utilizado para la fabricación de placas de circuito impreso.
- Interfaz intuitivo y atractivo estandarizado para todos los componentes de Proteus.

## Diseño

Las figuras 41-44 muestran la realización del diseño para el emplacado del OPF372A (Trasmisor).

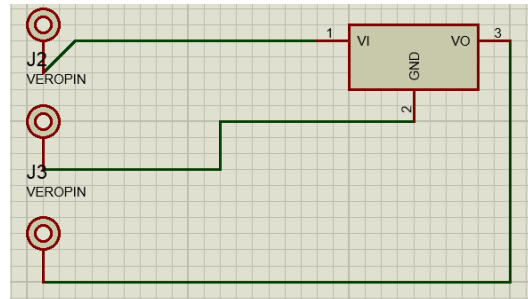


Figura 41 - Diseño Esquemático

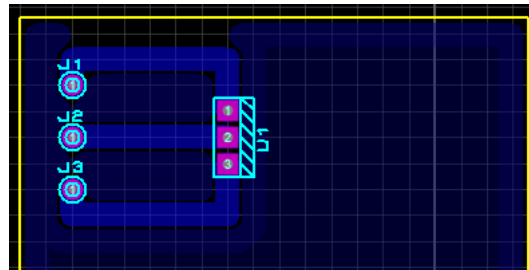


Figura 42 - Diseño PCB

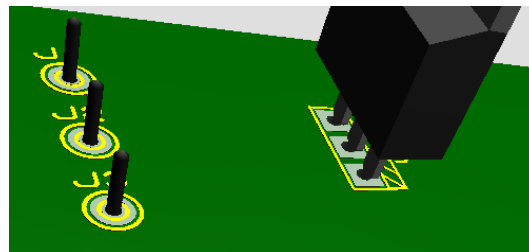


Figura 43 - Diseño 3D



Figura 44 - Placa

Las figuras 45-48 muestran la realización del diseño para el emplacado del OPF472 (Trasmisor).

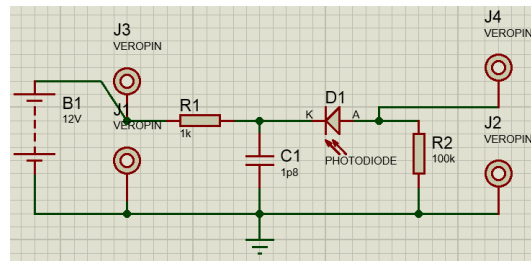


Figura 45 - Diseño Esquemático

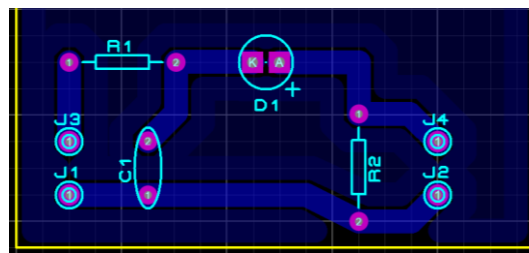


Figura 46 - Diseño PCB



Figura 47 - Diseño 3D



Figura 48 - Placa

## Caracterización de sensores

Las siguientes figuras muestran el comportamiento de un sensor de fibra probado en 3 ocasiones por cada parámetro a medir.

En la siguiente gráfica (Fig.49) podemos ver la correspondencia de la potencia conforme a la temperatura. Las mediciones que fueron desde 0 a 100 °C en intervalos de 10 °.

Podemos ver que el comportamiento es muy parecido en las 3 mediciones

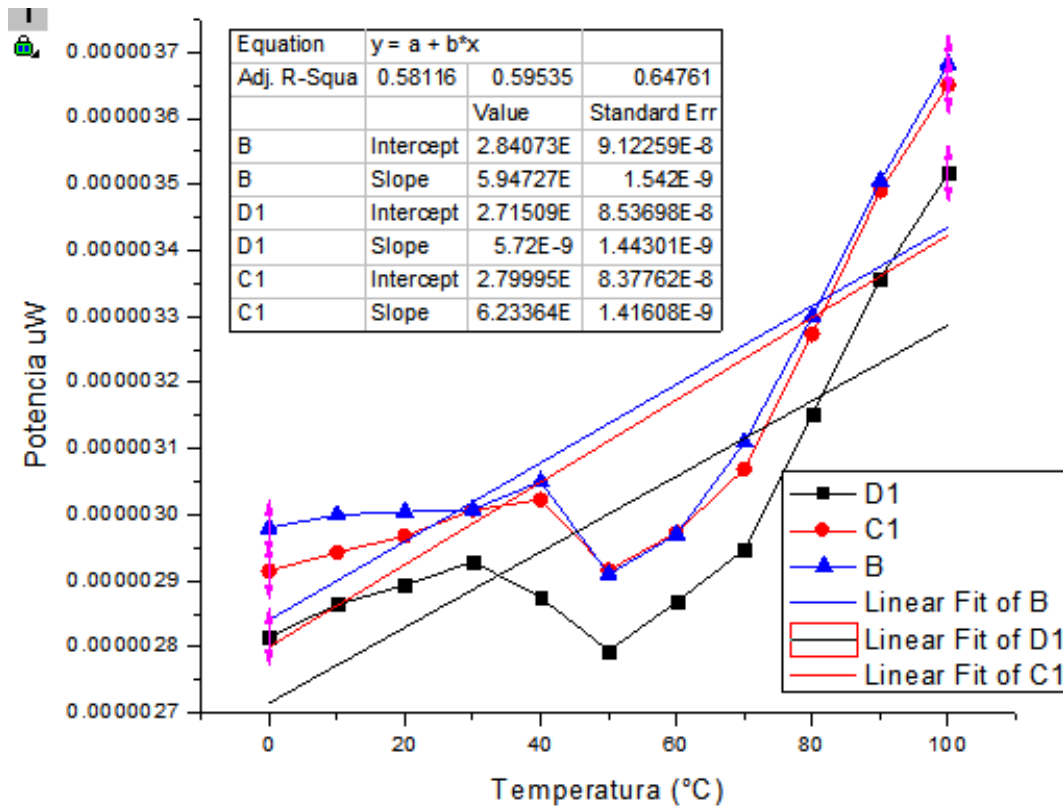


Figura 49 – Potencia 0 a 100 °C

La siguiente gráfica (Fig.50) expone la potencia según la temperatura, nuevamente la medición fue realizada en 3 ocasiones, En esta prueba las mediciones de potencia se hicieron variando la temperatura desde 100 a 0°C en intervalos de 10°. Como podemos los comportamientos en cada una de las mediciones son casi exactamente los mismos.

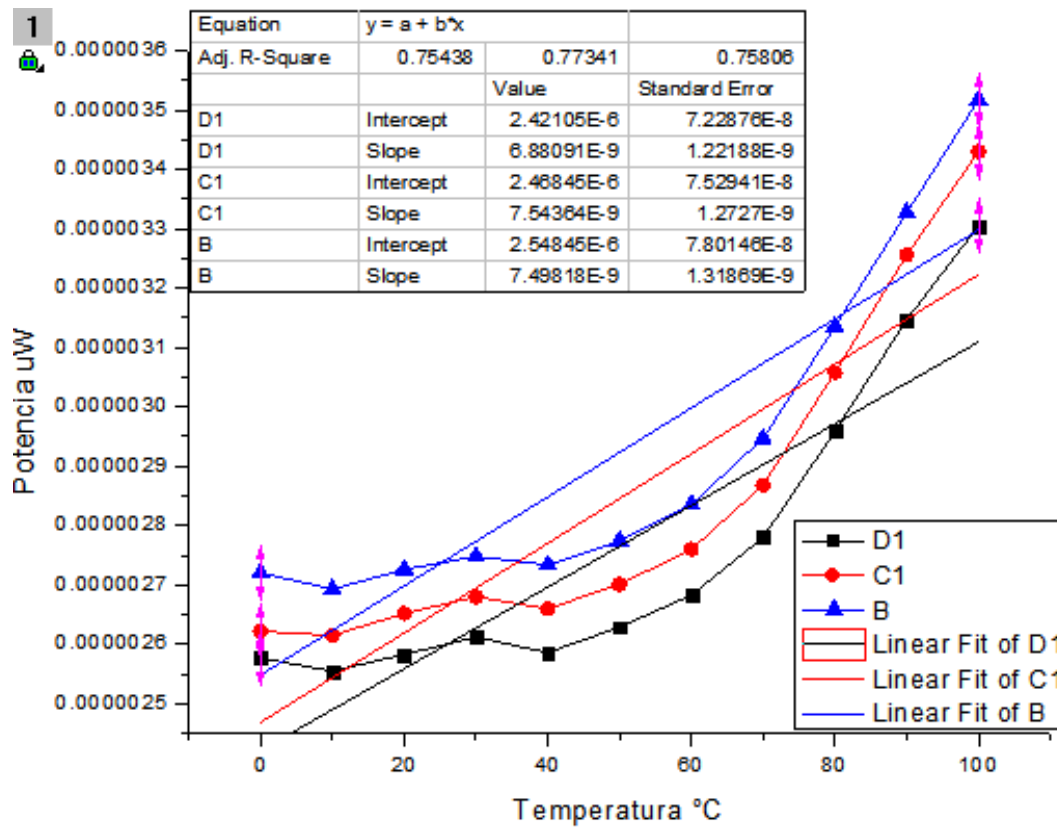


Figura 50 – Potencia 100 a 0°C

La siguiente gráfica (Fig.51) implica el voltaje receptado en medida que la temperatura varía en esta ocasión se mide de 0 a 100 °C. las pruebas fueron replicadas en 3 ocasiones, obteniendo lo siguiente.

Como podemos observar, ocurre un comportamiento similar que la grafica de la figura 49.

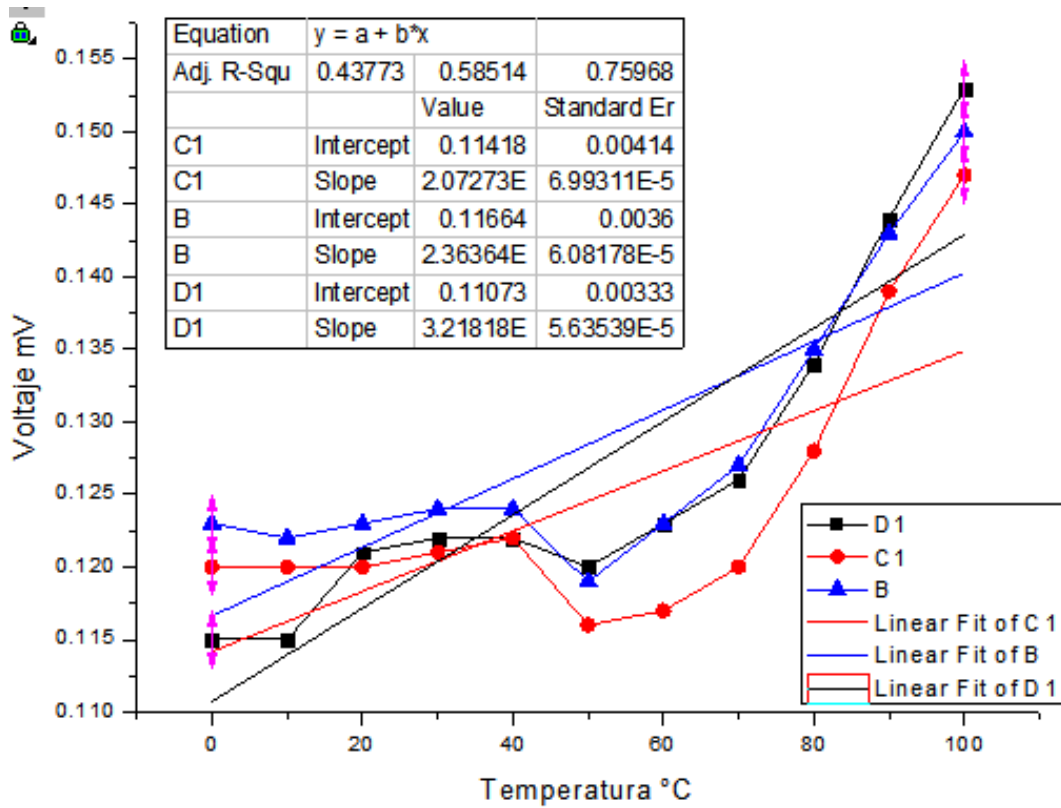


Figura 51 - Voltaje 0 a 100 °C

Aquí podemos apreciar la última gráfica, comparando el voltaje en variación a la temperatura, en esta ocasión las mediciones fueron 3 veces desde 100 a 0 °C. Y los resultados son similares a los obtenidos en la potencia de la gráfica de la figura 50.

Como conclusión en la caracterización decimos que en base al comportamiento similar entre el voltaje y la potencia medidas, podemos establecer correspondencia sabiendo que a determinado voltaje medido, podremos comparar la potencia medida y con esto arrojar la temperatura a la cual se alcanza esa determinada potencia.

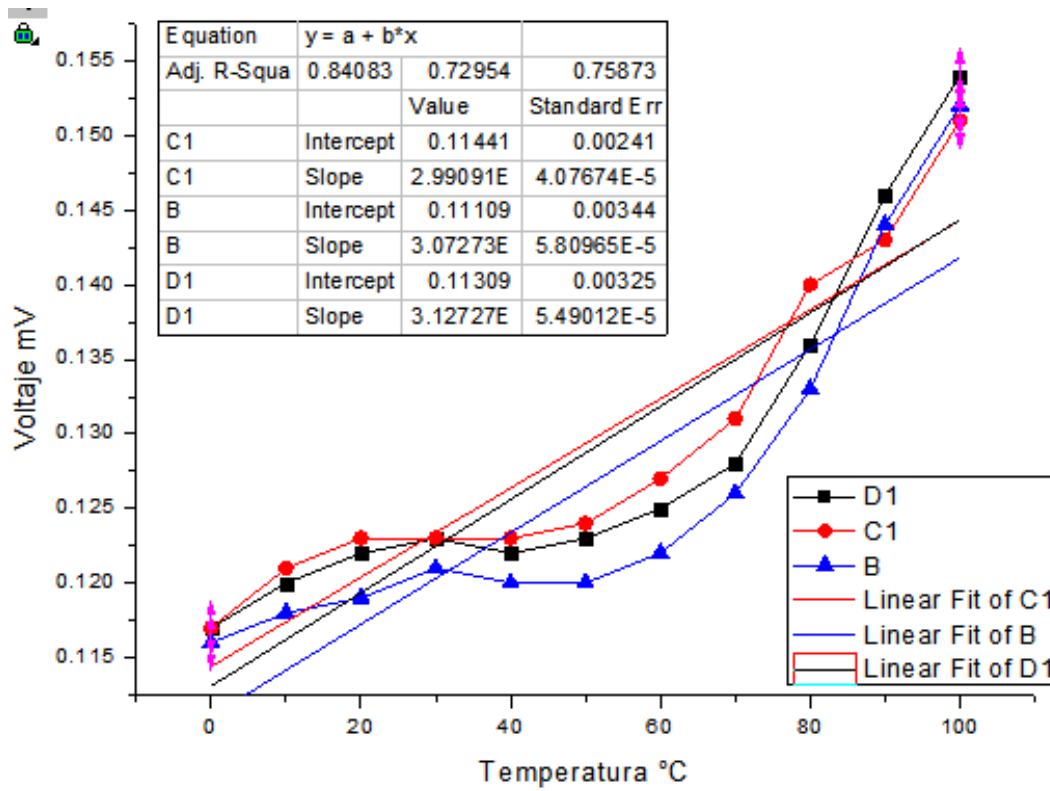


Figura 52 - Voltaje 100 a 0 °C

## 4.2. Sensores de temperatura

Dentro de los sensores de temperatura con los cuales se realizaron pruebas están el DTH11, TP-100 y los de fibra óptica. A continuación, se explica el uso, programación y medición de cada uno de ellos.

### 4.2.1. Módulo DTH-11

Es un sensor digital de temperatura y humedad, básico y de costo ultra bajo. Utiliza un sensor de humedad capacitivo y un termistor para medir el aire circundante, y emite una señal digital en el pin de datos (como se observa en la fig. 49 no necesita pin de entrada analógica). Es bastante simple de usar, pero requiere una sincronización cuidadosa para capturar datos. El único inconveniente real de este sensor es que solo puede obtener datos nuevos una vez cada 2 segundos.



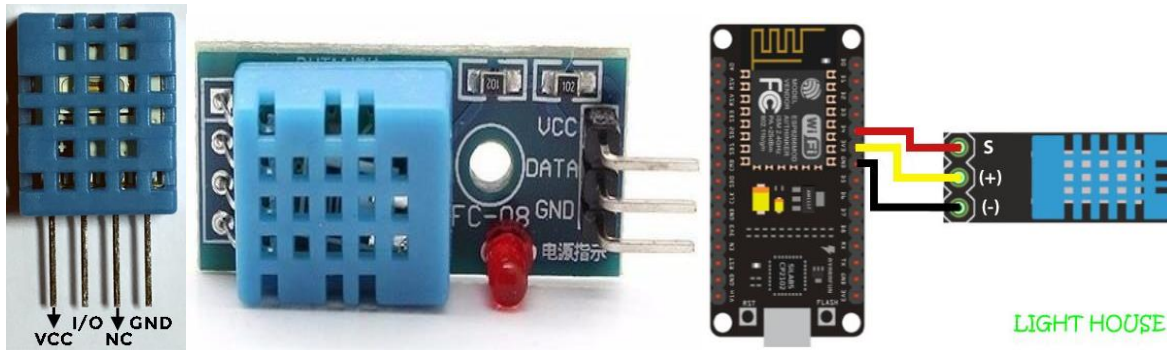


Figura 53 - Dht11

Los pines de la versión con PCB del DHT11 son:

- GND: conexión con tierra.
- DATA: transmisión de datos.
- VCC: alimentación.

Las características del DHT11 son realmente escasas, especialmente en rango de medición y precisión.

- Medición de temperatura entre 0 a 50, con una precisión de 2°C
- Medición de humedad entre 20 a 80%, con precisión del 5%.
- Frecuencia de muestreo de 1 muestras por segundo (1Hz).

## Programación

Para programar este sensor no fue requerido más que cables hembra – hembra, el código es el siguiente que mostramos en las figuras 50-52.

En la sección inicial del código declaramos nuestra librería, la variable y el pin donde se reciben los datos del sensor y declaramos como objeto el tipo de sensor a utilizar en el cual está inmerso nuestro pin del sensor.

## DHT11 §

```
#include <DHT11.h>
int pin=D4; //Seleccionar pin para el sensor
DHT11 dht11(pin); //Se selecciona el DHT11 (hay otros DHT)
```

Figura 54 – Programación DHT11

Inicializamos en el setup la comunicación con el monitor serial a 115200 baudios

```
void setup() {
  Serial.begin(115200); //Se inicia la comunicación serial
}
```

Figura 55 – Programación DHT11

En el loop declaramos como variables flotantes a la temperatura (temp), la humedad (hum) y un error (err) que nos indicará si la comunicación está establecida.

Condicionamos al error, el cual hará la lectura de nuestro objeto declarado, asignándole a nuestras variables de temperatura y humedad los valores leídos. Para luego ser impresos en el monitor serial.

En caso de no haber comunicación con el sensor será impreso un comentario en el monitor serial: "Error Num:" así como el error leído.

```

void loop() {
float temp, hum;
int err;
if((err=dht11.read(hum, temp))==0)
{
Serial.print("Temperatura: ");
Serial.print(temp);
Serial.print(" Humedad: ");
Serial.print(hum);
Serial.println();
}
else
{
Serial.println();
Serial.print("Error Num :");
Serial.print(err);
Serial.println();
}
delay(5000);
}

```

Figura 56 – Programación DHT11

#### 4.2.2. Sensor TH100PT

El TH100PT es un dispositivo detector de temperatura (RTD por sus siglas en inglés resistive temperature detector) mediante su capacidad resistiva, diseñado a base de substrato de óxido de Aluminio.



Figura 57 - Sensor TH100PT

#### Especificaciones

- Resistencia de 100Ω a 0°C.
- Repetibilidad de ±0.1°C.
- Estabilidad de ±0.1°C / año.
- Rango de temperatura de -70 a 400°C.
- Dimensiones 0.08" x 0.20" x 0.05" (2.0 mm x 5.0 mm x 1.3 mm)

Para poder programar este sensor fue necesario de un convertidor RTD, en este caso se utilizó el MAX31865.

## MAX31865

Es un convertidor de resistencia a digital fácil de utilizar y optimizado para su uso con detectores de temperatura con resistencia de platino (RTD) (fig. 54). Una resistencia externa establece la sensibilidad para el RTD utilizado y un ADC delta-sigma de precisión convierte el ratio de la resistencia del RTD con la resistencia de referencia en una forma digital. Las entradas del MAX31865 están protegidas contra fallos de sobretensión de hasta  $\pm 50$  V. Detección programable incluida de estados de circuito abierto y cortocircuito en el RTD y el cable.



Figura 58 - MAX31865

## Características

- Conversión simple de la resistencia del RTD de platino al valor digital
- Controla el RTD de platino de  $100 \Omega$  a  $1 \text{ k}\Omega$  (a  $0 \text{ }^\circ\text{C}$ ) (de PT100 a PT1000)
- Compatible con conexiones de sensores de 2, 3 y 4 cables
- Tiempo de conversión: 21 ms máx.
- Resolución ADC de 15 bits

## Programación

Para la programación de este sensor solo fueron requeridos macho – hembra. , el código es el siguiente que mostramos en las figuras 55-57.

En la parte inicial del código incluimos la librería necesaria para el MAX31, declaramos al MAX31 como objeto especificando los pines de conexión. Definimos un valor de resistencia de referencia (430) y uno nominal (100), puesto que estamos trabajando con el TH100PT.

```
TH100PT
#include <Adafruit_MAX31865.h>
// Use software SPI: CS, DI, DO, CLK
Adafruit_MAX31865 max = Adafruit_MAX31865(7, 8, 12, 13);
// use hardware SPI, just pass in the CS pin
//Adafruit_MAX31865 max = Adafruit_MAX31865(10);

// The value of the Rref resistor. Use 430.0 for PT100 and 4300.0 for PT1000
#define RREF      430.0
// The 'nominal' 0-degrees-C resistance of the sensor
// 100.0 for PT100, 1000.0 for PT1000
#define RNOMINAL 100.0
```

Figura 59 - Programación TH100PT

En el setup inicializamos la comunicación con el monitor serial a 115200 baudios e imprimimos un mensaje de prueba, inicializando nuestro MAX31 estableciendo que nuestro convertidor solo usará 2 cables.

```
void setup() {
  Serial.begin(115200);
  Serial.println("Adafruit MAX31865 PT100 Sensor Test!");

  max.begin(MAX31865_2WIRE); // set to 2WIRE or 4WIRE as necessary
}
```

Figura 60 - Programación TH100PT

En el loop declaramos una variable tipo flotante de la temperatura (temp) la cual tendrá el valor de la lectura de la temperatura según las resistencias nominal de referencia.

La cual será impresa en el monitor serial junto a un mensaje. Se realizarán mediciones cada 120000 ms (2 minutos).

```
void loop() {  
  //uint16_t rtd = max.readRTD();  
  float temp = (max.temperature(RNOMINAL, RREF));  
  Serial.print("Temperature = "); Serial.println(temp);  
  
  Serial.println();  
  delay(120000);  
}
```

Figura 61 - Programación TH100PT

### 4.2.3. Sensores De Temperatura De Fibra Óptica

Los sensores de temperatura de fibra óptica fabricados en el laboratorio de Optomecatrónica se basan a grandes rasgos en el dopaje de fibras ópticas estrechas (FOE) o Tapers, el dopaje de estos se basa en el funcionamiento requerido.

#### Estructuras Taper o fibras estrechadas

La fibra óptica estrechada (FOE) o Taper es objeto de estudio en el área de la óptica aplicada desde hace más de dos décadas, dispositivos tales como acopladores, multiplexores en longitud de onda, concentradores de haz, dispositivos acusto-ópticos o sensores son las principales aplicaciones conocidas de este tipo de estructura óptica. Los dispositivos de FOE pueden ser aprovechados como sensor de campo evanescente, principalmente sensores químicos donde se aprovecha la zona más estrecha de la estructura para depositar sustancias cuyas propiedades de absorción óptica cambia en función del parámetro que se desea medir.

Sensores de pH, humedad, concentración de gases entre otros han sido reportados con anterioridad. También, se pueden fabricar sensores de deformación de materiales basándose en el estirado de FOE multimodo embebida en un material.

La siguiente figura (58) muestra la estructura de un Taper donde se distinguen: la (zona I) o de contracción, la (zona II) o central, también conocida como cintura del Taper y la (Zona III) o de expansión.

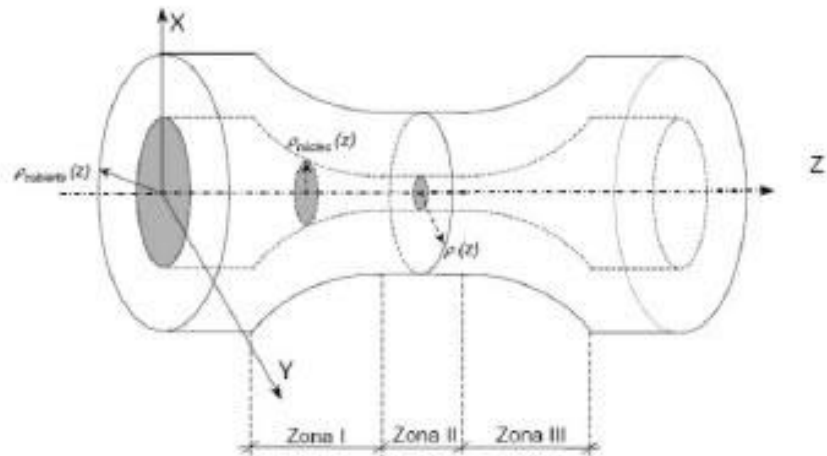


Figura 62 - Estructura por zonas de una fibra óptica estrechada (Taper)

## CAPÍTULO 5. RESULTADOS

En el capítulo anterior se revisaron las características de la placa y también se ha preparado el entorno de Arduino para empezar a programar el NodeMCU. A continuación, se explica el código utilizado con el cual se realizaron las pruebas pertinentes a la investigación.

Este código trata principalmente en la lectura de las variables provenientes de los sensores de temperatura establecidos en la placa, los cuales se enviaron a través de internet a un servidor de MATLAB® el cual lleva por nombre ThingSpeak™.

### 5.1. ThingSpeak

ThingSpeak es una plataforma abierta de aplicaciones, diseñada para permitir conectar personas con objetos. Se caracteriza por ser una plataforma Open Source con una API para almacenar y recuperar datos de los objetos usando el protocolo HTTP sobre Internet o vía LAN (Local Area Network) (Fig.59).

Se trata de una plataforma basada en Ruby on Rails 3.0 (RoR), este es un framework de aplicaciones web de código abierto basado en Ruby, cuya arquitectura está basada en el Modelo Vista Controlador (MVC). Se caracteriza por su simplicidad a la hora de programar aplicaciones del mundo real, escribiendo menos código y con una configuración mucho más sencilla que otros frameworks. Otra de las características que hacen de RoR un framework perfecto para el desarrollo de aplicaciones es que permite el uso de meta programación, haciendo que su sintaxis sea más legible y llegue a un gran número de usuarios.

La aplicación incluye todo lo necesario para poder empezar a trabajar, desde una aplicación web en la que podremos gestionar usuarios, gestionar claves de API, gestión de canales y cartografía.

“**ThingSpeak™** es un servicio de plataforma de análisis de IoT que le permite agregar, visualizar y analizar flujos de datos en vivo en la nube. ThingSpeak proporciona visualizaciones instantáneas de los datos



publicados por sus dispositivos en ThingSpeak. Con la capacidad de ejecutar código MATLAB® en ThingSpeak, puede realizar el análisis y el procesamiento - en línea - de los datos a medida que se incorporan”. (ThingSpeak, 2019).

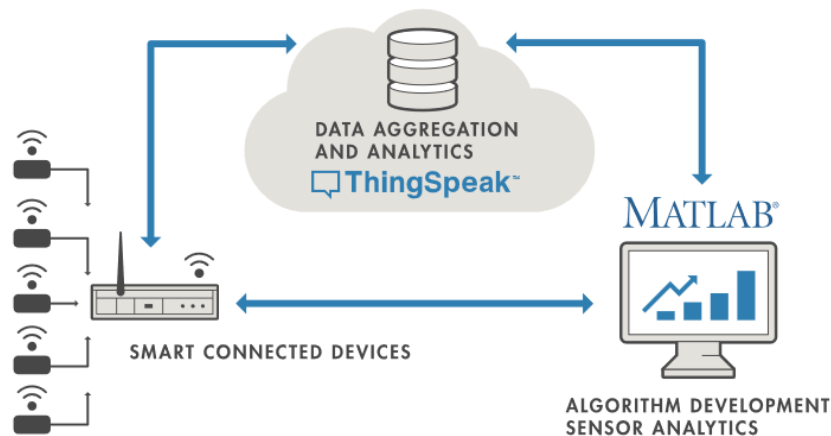


Figura 63 - ThingSpeak

### 5.1.1. Características principales

- API: A la hora de desarrollar cualquier proyecto es encontrar un API disponible de forma sencilla para que el desarrollador tenga los mecanismos necesarios para el desarrollo de la aplicación. En este caso, ThingSpeak dispone de una API la cual está disponible en GitHub para su descarga en un servidor propio. Es totalmente abierta, por lo que también se puede modificar su código fuente original y así contribuir a la comunidad con nuevas características, un principio básico en toda plataforma Open Source.
- Canales: La forma que tiene esta plataforma de almacenar y publicar los datos es a través de los “Channels” (Canales). Su creación es muy simple y en un par de clicks y rellenando una serie de datos lo tendremos disponible sin mayor complicación.
- Plugins: Para extender la funcionalidad del sitio también se nos brinda la oportunidad de desarrollar plugins. Estos nos ofrecen la posibilidad de crear aplicaciones de forma nativa en nuestra plataforma ThingSpeak. Soporta HTML,

CSS y JavaScript como lenguajes de programación. Al igual que los canales los plugins pueden ser público o privados según sean nuestras necesidades. Por último, destacar que ofrece la posibilidad de usar Google Gauge Visualization, gracias al cual nos ofrece la posibilidad de visualizar los datos de una forma rápida y amigable, con un nivel de personalización muy amplio.

- Integración Uno de los puntos fuertes en cualquier plataforma IoT, es que permita una amplia integración con diversos dispositivos Hardware y software. En este caso ThingSpeak permite la integración de su plataforma con:
  - Arduino
  - Raspberry Pi
  - IoBridge / RealTime.io
  - Electric Imp
  - Móviles / Aplicaciones web
  - Redes Sociales
  - Análisis de datos con MATLAB

### **5.1.2. Apps en ThingSpeak**

Sin duda las aplicaciones que encontramos en ThingSpeak son un complemento perfecto para nuestros proyectos, dotándolos en muchos casos de unas funcionalidades muy interesantes, actualmente tienen las siguientes aplicaciones en catálogo.

ThingTweet: Esta aplicación actúa como proxy permitiendo a nuestro dispositivo enviar actualizaciones de status a Twitter. Esto se consigue gracias a una llamada a la API de ThingTweet.

Tweet Control: Esta aplicación se pone a escuchar los hashtags de Twitter y luego permite controlar cualquier cosa que puedas imaginar. Cualquier cosa que se pueda conectar a través de ThingHTTP también se puede conectar a Twitter.

ThingHTTP: Con ThingHTTP permite a un microcontrolador o dispositivo a bajo nivel conectarse a un web service a través de Internet o la propia red. Soporta los métodos GET, POST, PUT y DELETE, SSL y Autenticación Básica.

React: Básicamente lo que realiza esa app, es ejecutar una acción cuando se cumple una cierta condición para los datos en nuestro canal. Por ejemplo, podríamos encender nuestra cafetera según llegamos a casa del trabajo, creando un React de Geolocalización.

TalkBack: Permite que cualquier dispositivo pueda actuar sobre los comandos en cola. Por ejemplo, podremos controlar una puerta añadiendo comandos tales como OPENDOOR, CLOSEDOR usando web interface o API. El funcionamiento podría ser el siguiente si alguien se acerca a la puerta esta se abre, si después de un par de minutos no se detecta movimiento esta se cierra.

TimeControl: Con TimeControl a una determinada hora, podremos ejecutar un ThingHTTP o un ThingTweet, e incluso añadir un nuevo comando a TalkBack.

Ya conociendo un poco de la plataforma ThingSpeak partimos a la programación del NodeMCU. La siguiente programación esta diseñada para leer el voltaje en la resistencia de carga situada en la placa donde se sitúa el dispositivo OPF474 (detector).

## **5.2. Programación**

En las figuras 60-63 podemos ver que el código inicia con las librerías correspondientes a al sensor DHT11 esto debido a que el uso del RTD TH100PT trajo complicaciones a la hora de compilar el código, se declaró la librería wifi del ESP, así como también la apiKey otorgada al crear el canal en ThingSpeak, se declaró la red wifi así como su contraseña, se declaró al servidor “api.thingspeak.com”, y también al pin de datos de lectura de temperatura sin olvidarse del objeto DHT11 en el que va incluido el pin anterior.

```

ThingSpeak$
#include <ESP8266WiFi.h>
#include <DHT11.h>
String apiKey = "S56Y4K06Y1N8T18D";
const char* ssid = "LaboratorioOptomecatronica";
const char* password = "2017opto";
const char* server = "api.thingspeak.com";
int pin=D4;
DHT11 dht11(pin);
WiFiClient client;

```

Figura 64 - Programación STFO

En el setup se estableció la comunicación con el monitor serial a 115200 baudios y un pequeño retardo para evitar traslapes, se inicializa el wifi de la placa, dejando un pequeño mensaje donde ha conectado con la dirección y la contraseña de la misma.

Otro pequeño retardo para después mandar a imprimir un mensaje donde la conexión ha sido exitosa.

```

void setup() {
  Serial.begin(115200);
  delay(10);
  //pinMode(D4, INPUT);
  WiFi.begin(ssid, password);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
}

```

Figura 65 - Programación STFO

Dentro del loop se utilizó el código ya dado para el sensado de temperatura con el DHT11

Agregando además en una variable flotante con la que leeremos el voltaje arrojado por la placa del OPF472, esta lectura se hizo con utilizando el pin análogo A0. Se asignará

a la variable declarada el valor de la lectura, está siendo multiplicada por el voltaje nominal de la placa, todo esto dividido entre la resolución del convertidor ADC de la misma.

Para después ser mandado a imprimir un texto de voltaje de salida, junto a la variable leída, ajustando 5 cifras después del punto.

```
void loop() {
  float temp, hum;
  int err;
  if((err=dht11.read(hum, temp))==0)
  {
    Serial.print("Temperatura: ");
    Serial.print(temp);
    Serial.print(" Humedad: ");
    Serial.print(hum);
    Serial.println();
  }
  else
  {
    Serial.println();
    Serial.print("Error Num :");
    Serial.print(err);
    Serial.println();
  }

  float val = analogRead(A0);
  val = (val*3.3000)/1023;
  Serial.print("Voltaje Entrada:");
  Serial.println(val,5);
  //delay(2000);
}
```

Figura 66 - Programación STFO

Lo siguiente en el código no es más que condicionantes para el envío de datos de la plataforma ThingSpeak. Donde fueron recibidos los datos de las lecturas que obtuvimos.

Dejando un pequeño retardo, pues en cuestión no es necesaria la velocidad de lectura de una variable tan inestable como es la temperatura, entonces es por eso que se dejó un lapso de 120000 ms (2 minutos).

```

if (client.connect(server,80)){
  String postStr = apiKey;
  postStr += "&field1=";
  postStr +=String((int)temp);
  postStr += "&field2=";
  postStr +=String((int)hum);
  postStr += "&field3=";
  postStr +=String((float)val);
  postStr += "\r\n\r\n";
  client.print("POST /update HTTP/1.1\n");
  client.print("Host: api.thingspeak.com\n");
  client.print("Connection: close\n");
  client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
  client.print("Content-Type: application/x-www-form-urlencoded\n");
  client.print("Content-Length: ");
  client.print(postStr.length());
  client.print("\n\n");
  client.print(postStr);
  Serial.println("§ sent to Thingspeak");
}
client.stop();
Serial.println("Waiting...");
delay(120000);
}

```

Figura 67 - Programación STFO

### 5.3. Grafica de datos en tiempo real

Habiendo ya programado el código se avanzó a la parte de las pruebas, se utilizo un controlador de temperatura para variar la temperatura a la que se sometió el sensor de fibra óptica. En las figuras se explica.

El control de temperatura utilizado en el desarrollo de las mediciones fue el Quantum TC125 y el qpod (Fig.64).



Figura 68 - Quantum TC125 y qpod

Conforme la temperatura dentro del qpod fuese variando la trasmisión de las variables: voltaje, temperatura y humedad iba a ir variando en su graficación.

Podemos ver que el trasmisor esta ahora conectado a voltaje en configuración de 1.8 v y 100 mA. (Fig.65)

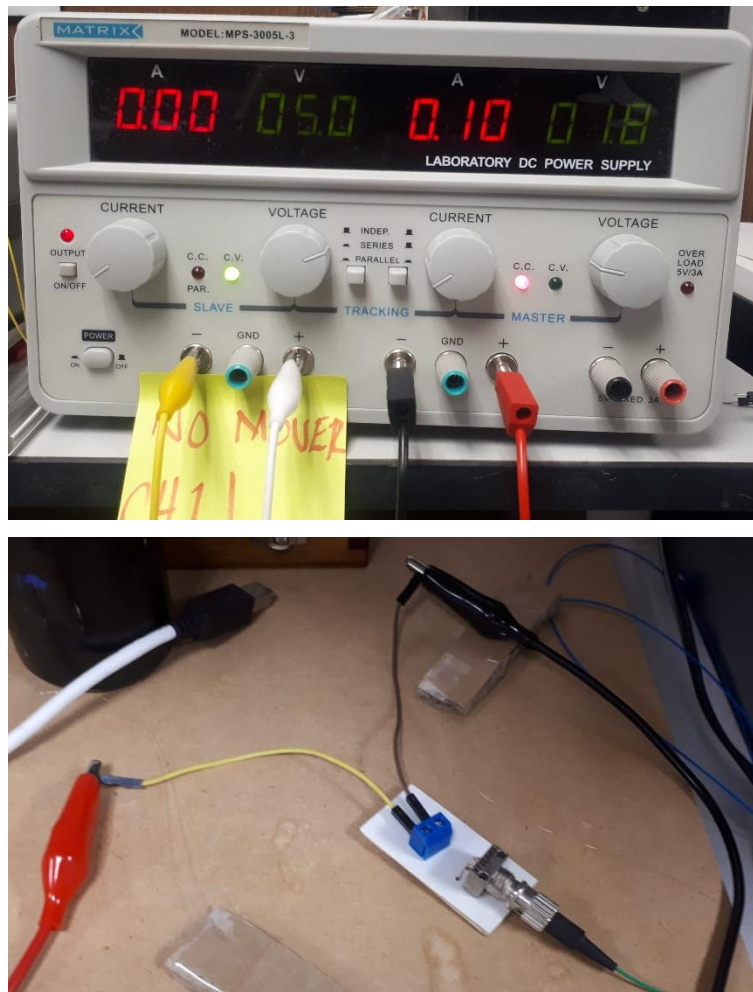


Figura 69 - Trasmisor en operación

El Controlador se mantuvo variando manualmente de 0 a 100 °C y de 100 a 0 °C en intervalos de 10° como se aprecia en la figura 66. Esto debido a que el control automático era demasiado rápido que era difícil apreciar con buena precisión los valores de nuestras variables en voltaje.



Figura 70 - Variaciones de Temp.

En la siguiente figura (Fig. 67) podemos apreciar el sensor de fibra, la fuente de luz y el dispositivo receptor, esto antes de ser colocado en el qpod.

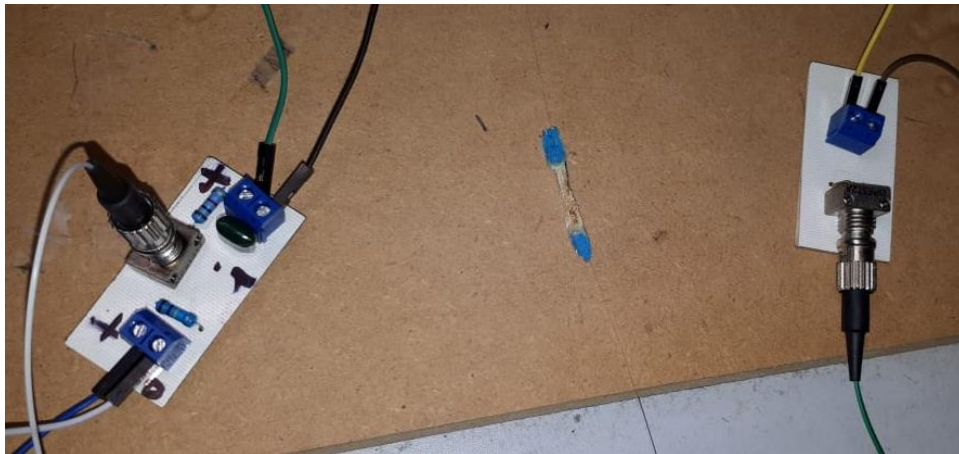


Figura 71 - Sensor de Temperatura

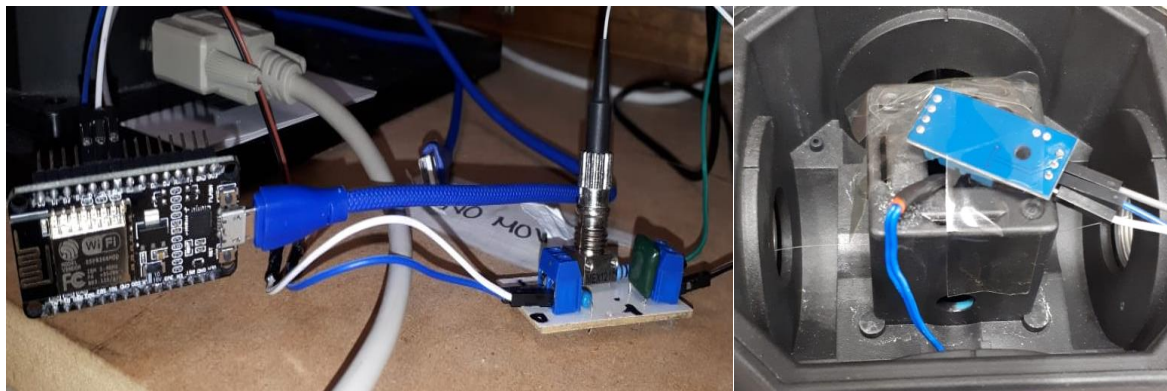
El sensor de fibra es introducido al qpod (Fig.68) con esto variando sus propiedades de trasmisión, estas variaciones en la trasmisión podrán ser observadas cuando el voltaje sea graficado en el servidor ThingSpeak.





*Figura 72 - Sensor dentro del qpod*

En la figura 69 se puede observar la placa ESP realizando la lectura análoga del voltaje en la resistencia de carga de la placa del receptor OPF. También se puede observar el sensor DTH-11 al borde del qpod. Así como también se observa al sensor TH100PT, el cual esta censado la temperatura en otro punto dentro del controlador.



*Figura 73 - Sensado de temperatura*

Teniendo ya todo lo antes mencionado, será necesario regresar al canal adquirido en ThingSpeak para poder corroborar el envío de datos y su posterior graficación.

(Los datos de la transmisión de voltaje rondaban el orden de mili Volts, por lo tanto fue un poco difícil el acople de la señal).

Gráfica de la temperatura. (Fig.70)

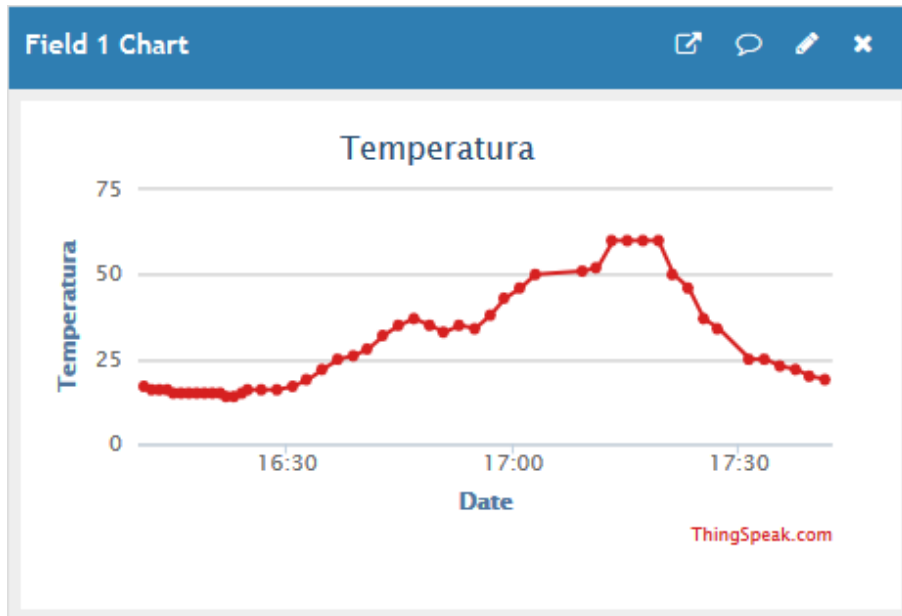


Figura 74 – Temperatura

Gráfica de humedad (Fig. 71)

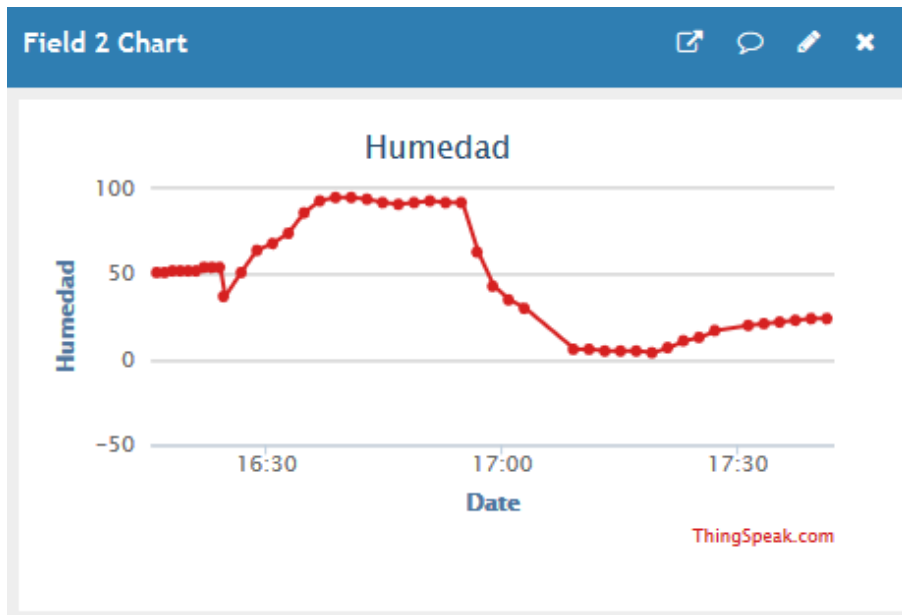


Figura 75 - Humedad

## Gráfica de voltaje

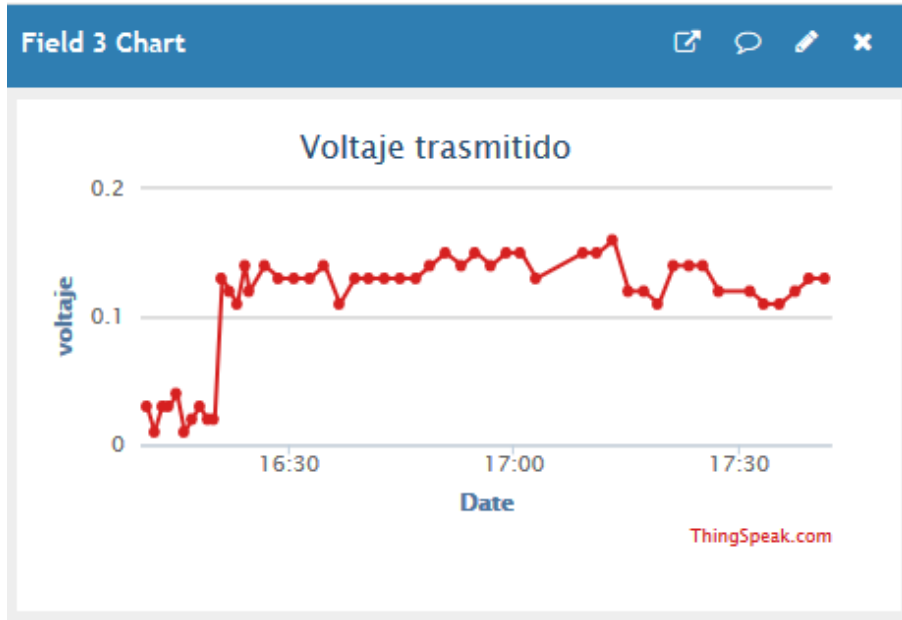


Figura 76 - voltaje

La Figura 73 muestra una toma general del canal de ThingSpeak

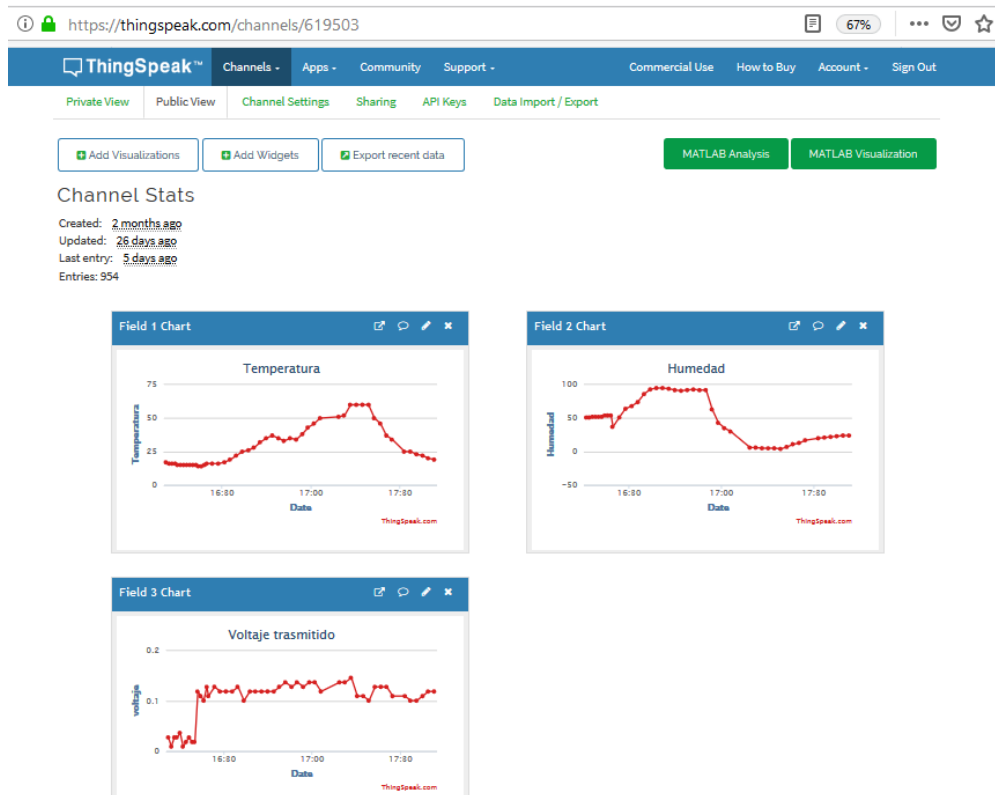


Figura 77 - Channel ThingSpeak

## **Conclusión**

A lo largo de la investigación se han ido realizando las tareas específicas requeridas para cada etapa, como fueron la caracterización de los sensores de fibra óptica en potencia y voltaje, que gracias a esto pudimos establecer un cierto criterio de comportamiento con el que podemos saber a grandes rasgos los valores de voltaje o potencia lumínica que corresponden a un valor en la temperatura.

Gracias a esta investigación pudimos ver como el gradiente de temperatura afecta las mediciones en nuestros sensores, los cuales eran de distinta calidad, esto también afecta a nuestro criterio de evaluación pues caemos en la incertidumbre de no saber cual sensor pueda estar censando correcta o incorrectamente.

A lo largo de esta investigación pudimos cumplir con los objetivos planteados al inicio.

El envío de datos como temperatura, humedad y voltaje a través del internet se ha llevado a cabo, aunque con un cierto tiempo de desfase puesto que la aplicación web en la que nos basamos necesita de ello, esto no afecta nuestras mediciones, pues al ser la temperatura una variable muy difícil de controlar, nuestro censado de la misma es a determinados lapsos de tiempo.

Podemos terminar diciendo que el internet de las cosas ya está aquí y que los sensores de fibra no parecen quedarse estancados. Pues con este reporte estamos acreditando la factible posibilidad de combinar ambos mundos.

## REFERENCIAS

- Arenas, G. (2007). Aplicaciones de un sensor de fibra óptica a la medición no invasiva de desplazamientos y vibraciones en el rango micro y sub-micrométrico. *IV Conferencia Panamericana de END Buenos Aires - Octubre 2007*. Buenos Aires, Argentina.
- Aristizábal, V. (2007). INTRODUCCIÓN A LA TECNOLOGÍA DE FIBRAS ÓPTICAS. *Tecno-Lógicas*, 143-145.
- Barrio, M. (2018). *INTERNET DE LAS COSAS*. Madrid: Reus.
- Bernis, L. (s.f.). *Servidores web*.
- Chomycz, B. (s.f.). *Instalaciones de fibra óptica fundamentos, técnicas y aplicaciones*. Mc Graw Hill.
- Chwalisz, M. (2018). *thingspeak Documentation*.
- Díaz, N. (2005). *Desarrollo de sensores de fibra Óptica para el control In-situ De Parámetros físicos del medio Acuático*. Madrid.
- García, I. (2017). *Desarrollo de un sensor de fibra óptica*. Bilbao.
- Loureiro, R. (2015). ESTUDIO PLATAFORMAS IOT. En R. Loureiro, *ESTUDIO PLATAFORMAS IOT* (págs. 5-7).
- LTDA., C. E. (11 de Abril de 2018). *Componentes Electrónicas LTDA*. Obtenido de Componentes Electrónicas LTDA: <https://www.compelect.com.co/>
- Martínez, A. (2013). *Sensores de fibra óptica para la detección de compuestos orgánicos volátiles basados en xerogeles silíceos híbridos*. Navarra.
- Ning, D. (2017). Developing and Deploying Analytics for IoT Systems. *MATLAB CONFERENCE 2017*. Australia.
- Valle, L. d. (2017). *Programa Fácil*. Obtenido de Programa Fácil: <https://programarfacil.com/>
- Yoany Rodríguez, e. a. (2013). Desarrollo de estructuras de fibra óptica estrechada (TAPER) para la detección de vibraciones mecánicas. *Revista Ciencia y Tecnología*.