



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería Mecánica Eléctrica

**DISPOSITIVO QUE MIDE EL ÍNDICE ULTRAVIOLETA, SOBRE UN  
PACIENTE PARA PREVENIRLO DEL CANCER DE PIEL, APLICANDO  
INTERNET DE LAS COSAS**

**Jaime Ariel Chitay Bautista**

Asesorado por el Ingeniero José Aníbal Silva de los Ángeles

Guatemala, agosto de 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISPOSITIVO QUE MIDE EL ÍNDICE ULTRAVIOLETA, SOBRE UN  
PACIENTE PARA PREVENIRLO DEL CANCER DE PIEL, APLICANDO  
INTERNET DE LAS COSAS**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**JAIME ARIEL CHITAY BAUTISTA**

ASESORADO POR EL INGENIERO JOSE ANIBAL SILVA DE LOS ÁNGELES

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO ELECTRÓNICO**

GUATEMALA, AGOSTO DE 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Guillermo Antonio Puente
EXAMINADOR	Ing. Armando Alonso Rivera
EXAMINADOR	Ing. Erick Carlos Roberto Navarro
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **DISPOSITIVO QUE MIDE EL ÍNDICE ULTRAVIOLETA, SOBRE UN PACIENTE PARA PREVENIRLO DEL CANCER DE PIEL, APLICANDO INTERNET DE LAS COSAS**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica con fecha 5 de abril del 2018.

**Jaime Ariel Chitay Bautista**

Guatemala, 10 de febrero de 2020

Ing. Armando Rivera  
Director Escuela de Ingeniería Mecánica Eléctrica  
Facultad de Ingeniería  
Universidad de San Carlos de Guatemala

Respetable Ingeniero Rivera

Por medio de la presente informo a usted que, como asesor del Trabajo de Graduación del estudiante universitario Jaime Ariel Chitay Bautista, quien se identifica con carné universitario No. 2011-14550, procedí a hacer la revisión de la tesis que contiene cinco capítulos, cuyo título es: "**DISPOSITIVO QUE MIDE EL ÍNDICE ULTRAVIOLETA, SOBRE UN PACIENTE PARA PREVENIRLO DEL CANCER DE PIEL, APLICANDO INTERNET DE LAS COSAS.**", el cual encuentro como satisfactorio.

En tal virtud, LA DOY POR APROBADA, solicitándole dar el trámite correspondiente.

Atentamente

**JOSE ANIBAL SILVA DE LOS ANGELES**

ING. ELECTRÓNICO  
COLEGIADO NO. 5067

Ing José Anibal Silva de los Angeles  
Colegiado 5067



Guatemala, 17 de febrero de 2020

Señor Director  
Armando Alonso Rivera Carrillo  
Escuela de Ingeniería Mecánica Eléctrica  
Facultad de Ingeniería, USAC

Estimado Señor Director

Por este medio me permito dar aprobación al Trabajo de Graduación titulado **DISPOSITIVO QUE MIDE EL ÍNDICE ULTRAVIOLETA, SOBRE UN PACIENTE PARA PREVENIRLO DEL CÁNCER DE PIEL, APLICANDO INTERNET DE LAS COSAS**, desarrollado por el estudiante **Jaime Ariel Chitay Bautista**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Agradecemos.

**ID Y ENSEÑAD A TODOS**


Ing. Julio César Solares Peñate  
Ingeniero de Electrónica





REF. EIME 14. 2020.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto bueno del Coordinador de Área, al trabajo de Graduación del estudiante: JAIME ARIEL CHITAY BAUTISTA titulado; DISPOSITIVO QUE MIDE EL ÍNDICE ULTRAVIOLETA, SOBRE UN PACIENTE PARA PREVENIRLO DEL CÁNCER DE PIEL, APLICANDO INTERNET DE LAS COSAS procede a la autorización del mismo.

  
Ing. Armando Alonso Rivera Carrillo



GUATEMALA, 5 DE MARZO 2020.

DTG. 225.2020.

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Eléctrica, al Trabajo de Graduación titulado: **DISPOSITIVO QUE MIDE EL ÍNDICE ULTRAVIOLETA, SOBRE UN PACIENTE PARA PREVENIRLO DEL CANCER DE PIEL, APLICANDO INTERNET DE LAS COSAS**, presentado por el estudiante universitario: **Jaime Ariel Chitay Bautista**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Inga. Anabela Cordova Estrada  
Decana



Guatemala, septiembre de 2020

AACE/asga



## **ACTO QUE DEDICO A:**

<b>Dios</b>	Que hasta aquí no ha dejado de ayudarme.
<b>Mi madre</b>	Nardaluz Bautista, que me dio el regalo de creer en mí siempre. Gracias a ti, nunca dudé que lograría alcanzar esta meta.
<b>Mi padre</b>	Carlos Humberto Chitay Medrano. Gracias a tu ejemplo y esfuerzo, pude superarme siempre. Eres el mejor.
<b>Mis Hermanos</b>	Ada, Mariela y Sammy Chitay, gracias por su ejemplo de disciplina, creatividad y tranquilidad. Me siento orgulloso de ser su hermano.
<b>Mi hermano</b>	Carlos Misraim Chitay Bautista. Por haber despertado mi interés por la matemática. Nada de esto estuviera pasando sin tu ayuda.
<b>José García</b>	Por guiarme al camino de la ciencia de datos y mostrarme lo increíble que puede ser esta nueva aventura.

**Mis amigos de la  
Universidad**

Nahum, Joshua, Kevin, Alex, Canel, Jaganath, Ronald, Marlon Ramos y Panchoso. Sin ustedes, la universidad hubiera sido muy aburrida, gracias por su amistad y ser mis compañeros de equipo en los proyectos, son brillantes.

**Equipo BAC Credomatic**

Por permitirme salir una hora antes de mi trabajo para que pudiera terminar esta fase importante de mi vida.

**Pueblo de Guatemala**

Gracias por pagar mis estudios, seré un profesional de bien para todo aquel que me necesite.

## **AGRADECIMIENTOS A:**

**Universidad de San de  
Carlos de Guatemala**

Por permitirme entrar a sus aulas y bibliotecas, enseñándome que de nada sirve el conocimiento si no se comparte.

**Escuela de Ingeniería  
Mecánica Eléctrica**

Por guiarme en el camino de la ciencia y tecnología y seguir persiguiendo la innovación.

**Iglesia Nazaret Norte**

Porque me dio la mejor etapa de mi vida, enseñándome a servirle al prójimo y me rodeó de personas increíbles, gracias a todos por acompañarme desde niño.

**Mis amigos**

Levi y Omar, amigos del alma que han estado conmigo, en todas las etapas de mi vida.

**Familia Castillo Zea y  
Álvarez Contreras**

Entré a la universidad con una familia y salí con tres. Gracias por recibirme en sus hogares, proveerme de un lugar donde dormir y estudiar con mis grandes colegas y también grandes amigos.

**Familia Zamora Mérida**

Que me ayudaron en momentos muy difíciles en mi carrera universitaria.

**Ing. José Aníbal Silva**

Por aceptar el reto de estudiar y asesorarme en el mundo del Internet de las Cosas.

**Andrea Maribé  
Lemus Polanco**

No sabía que esta etapa de mi vida se podía convertir en algo tan bello cuando se acompaña de un lindo amor como el tuyo.

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
LISTA DE SÍMBOLOS .....	IX
GLOSARIO .....	XI
RESUMEN.....	XVII
OBJETIVOS.....	XIX
INTRODUCCIÓN .....	XXI
1. CONCEPTOS GENERALES SOBRE EL CÁNCER DE PIEL.....	1
1.1. Tipos de cáncer de piel .....	1
1.1.1. El melanoma.....	1
1.1.2. El carcinoma basocelular.....	1
1.1.3. Carcinomas de células escamosas .....	2
1.2. Signos de alerta que indican la existencia de cáncer de piel.....	2
1.3. Factores de riesgo para cáncer de la piel.....	2
1.4. Antecedentes.....	3
1.5. ¿Qué es el índice UV?.....	7
1.5.1. Escalas del índice Ultravioleta .....	7
1.5.1.1. 0 a 2: bajo.....	8
1.5.1.2. 3 a 5: moderado.....	8
1.5.1.3. 6 a 7: alto.....	9
1.5.1.4. 8 a 10: muy alto .....	9
1.5.1.5. 11 o más: extremo .....	10
1.6. Efectos de la exposición a los rayos UV.....	10
1.6.1. Cáncer de la piel.....	10
1.6.2. Envejecimiento prematuro .....	11

1.6.3.	Cataratas y otras enfermedades de los ojos .....	11
1.6.4.	Daño al sistema inmunológico.....	11
1.7.	La intensidad de la radiación UV .....	11
1.7.1.	La altura del sol .....	12
1.7.2.	La reflexión por el suelo .....	12
1.7.3.	La latitud.....	12
1.7.4.	La nubosidad.....	12
1.7.5.	La altitud.....	13
1.7.6.	El ozono .....	13
2.	CONCEPTOS GENERALES DEL IOT.....	15
2.1.	¿Qué es Internet de las Cosas (IoT)? .....	15
2.2.	Arquitecturas de IoT .....	15
2.2.1.	Arquitectura de tres capas.....	16
2.2.2.	Arquitectura de cinco capas .....	17
2.2.3.	Taxonomía del IoT .....	18
2.2.3.1.	Percepción .....	19
2.2.3.2.	Preprocesamiento .....	20
2.2.3.3.	Comunicación (capa de red) .....	22
2.2.3.3.1.	IoT Network protocol stack.....	24
2.2.3.3.2.	Bluetooth de baja energía (BLE).....	30
2.2.3.3.3.	Wi-Fi de baja potencia...31	
2.2.3.3.4.	Zigbee .....	32
3.	CAPA FÍSICA O DE PERCEPCIÓN .....	35
3.1.	NodeMCU .....	35
3.2.	Datasheet NodeMCU .....	35

3.3.	ESP8266 .....	37
3.4.	Conexión entre ESP8266 y NodeMCU.....	38
3.5.	Instalación de <i>drivers</i> en Windows .....	38
3.5.1.	Conexión .....	39
3.5.2.	Instalación de <i>drivers</i> .....	39
3.5.3.	Instalación de <i>driver</i> equivalente .....	40
3.6.	Putty .....	41
3.6.1.	Instalación de PuTTY .....	42
3.7.	El SI1145.....	43
3.7.1.	Pines de potencia .....	44
3.7.2.	Pines Extra .....	45
3.7.3.	I2C Pins .....	45
3.8.	NodeMCU Flasher .....	46
3.8.1.	Interfaz gráfica NodeMCU Flasher .....	47
3.9.	MicroPython.....	49
3.9.1.	Firmware para la placa ESP8266 .....	50
3.9.2.	Instalación del Firmware al SOC ESP8266 .....	51
3.10.	Python 3 .....	52
3.10.1.	Descarga de Python 3 .....	53
3.10.2.	Instalación de Python 3 .....	53
3.11.	Esptool.py.....	55
3.11.1.	Instalación de Esptool.py.....	55
4.	SERVIDOR WEB .....	57
4.1.	Modelo de un Servidor Web .....	57
4.1.1.	SOAP - Simple Object Access Protocol.....	58
4.1.2.	WSDL - Web Services Description Language .....	58
4.1.3.	UDDI - Universal Description, Discovery and Integration.....	58

4.2.	Arquitectura de un Servidor WEB .....	59
4.3.	API .....	60
4.4.	REST.....	60
4.4.1.	Características de REST .....	61
4.5.	Web <i>Frameworks</i> .....	62
4.5.1.	Full Stack Web Frameworks.....	63
4.5.2.	Django.....	63
4.5.3.	Web <i>Frameworks</i> "ligeros" .....	64
4.5.4.	Flask.....	64
4.6.	Diseño y creación del Backend .....	66
5.	FRONTEND .....	71
5.1.	Front-end Developer .....	71
5.2.	HTML .....	71
5.3.	CSS.....	73
5.3.1.	Sintaxis CSS .....	74
5.3.2.	¿Cómo funciona realmente CSS? .....	76
5.4.	JavaScript .....	77
5.5.	Bootstrap.....	78
5.6.	Componentes Bootstrap.....	79
5.7.	Plantillas Bootstrap.....	80
5.8.	Jinja.....	81
5.9.	Frontend del proyecto .....	81
5.9.1.	Plantilla base del frontend .....	82
5.9.2.	Sección de bienvenida .....	86
5.9.3.	Sección del Dashboard .....	87



CONCLUSIONES .....	89
RECOMENDACIONES .....	91
BIBLIOGRAFÍA.....	93
APÉNDICE.....	95



## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1.	Índice UV.....	7
2.	Arquitectura de IoT.....	16
3.	Taxonomy of research in IoT technologies.....	18
4.	Smart watches .....	19
5.	Puerta de enlace inteligente para el procesamiento .....	22
6.	Diagrama de los pines de la plataforma.....	36
7.	Diagrama de ESP8266.....	37
8.	Manera correcta de conectar el dispositivo .....	39
9.	Descarga de los controladores.....	40
10.	Ubicación del puerto.....	40
11.	Descarga del controlador equivalente .....	41
12.	Muestra del controlador en el ordenador.....	41
13.	Instalación del PuTTY .....	42
14.	Configuración del PuTTY .....	43
15.	Placa de conexiones Adafruit SI1145 - Índice UV/IR/ Sensor Visible...	44
16.	Diagrama de conexión .....	46
17.	Repositorio .....	47
18.	Interfaz gráfica del ESP8266.....	48
19.	Configuración del ESP8266 .....	48
20.	Descarga del firmware .....	50
21.	Configuración del firmware.....	51
22.	Configuración del firmware parte 2.....	52
23.	Descarga de Python 3.....	53

24.	Instalación de Phyton 3.....	54
25.	Adquisición de datos desde la ESP8266 y el SI1145 .....	56
26.	Modelo de servicios web.....	59
27.	Modelo de datos con flask .....	66
28.	Creación de la interfaz .....	67
29.	Muestra del sitio web .....	72
30.	Lenguaje HTML .....	73
31.	Estilo del código CSS .....	74
32.	Muestra de sitio Web .....	75
33.	Función del CSS.....	77
34.	Modelo de JavaScript .....	78
35.	Plantilla base del fronted.....	82
36.	Sección de bienvenida.....	86
37.	Sección del Dashboard.....	87

## TABLAS

I.	Edad y sexo de los casos de cáncer detectados durante el año 2013. Registro Hospitalario del Incan – Guatemala.....	3
II.	Registro hospitalario del Incan - Guatemala. Residencia y sexo de los casos detectados durante el año 2013.....	4
III.	Registro hospitalario del Incan, Guatemala. localizaciones (CIE-10) y sexo de los casos de cáncer detectados durante el año 2013.....	5
IV.	Diagrama interno entre ESP8266 y NodeMCU.....	38
V.	Ejemplo plantillas Boostrap.....	80

## LISTA DE SÍMBOLOS

<b>Símbolo</b>	<b>Significado</b>
<b>MCC</b>	Computacion en la nube movil
<b>NFC</b>	Comunicación de campo cercano
<b>CoAP</b>	Constrained Application Protocol
<b>DODAG</b>	Destination Oriented Directed Acyclic Graph (DODAG)
<b>HTTP</b>	El Protocolo de transferencia de hipertexto
<b>SPF</b>	Factor de protección solar
<b>RFID</b>	Identificación por radiofrecuencia
<b>IUV</b>	Índice Ultra Violeta
<b>API</b>	Interfaz de programación de aplicaciones
<b>IoT</b>	Internet de las Cosas
<b>6LoWPAN</b>	IPv6 over Low power Wireless Personal Area Networks
<b>SQL</b>	Lenguaje de consulta estructurada
<b>XML</b>	Lenguaje de Mercado Extensible
<b>MAC</b>	Media Access Control
<b>MQTT</b>	Message Queue Telemetry Transport
<b>mW</b>	Mili Watt
<b>MQTT-S</b>	MQTT for Sensor Networks
<b>MQTT-SN</b>	MQTT for Sensor networks
<b>NFC</b>	Near field communication
<b>IR</b>	Radiación Infrarroja
<b>RFID</b>	Radio Frequency Identification

<b>REST</b>	Transferencia de estado representacional
<b>Uv</b>	Ultravioleta
<b>MTU</b>	Unidad de transmisión máxima

## GLOSARIO

<b>6LoWPAN</b>	Es un estándar que posibilita el uso de IPv6 sobre redes basadas en el estándar IEEE 802.15.4. Hace posible que dispositivos, como los nodos de una red inalámbrica, puedan comunicarse directamente con otros dispositivos IP.
<b>Backend</b>	Es la parte que se conecta con la base de datos y el servidor que utiliza dicho sitio web, por eso se dice que el backend corre del lado del servidor.
<b>COAP</b>	Es un Protocolo de aplicación de internet especializado para dispositivos restringidos.
<b>Córnea</b>	Membrana transparente en forma de disco abombado, que constituye la parte anterior del globo ocular y se halla delante del iris.
<b>El melanocito</b>	Es una célula dendrítica que deriva de la cresta neural y que migra hacia la epidermis y el folículo piloso durante la embriogénesis. Su principal función es la producción de melanina que tiene importancia en la protección contra los rayos solares.

<b>Firmware</b>	Es el <i>software</i> que tiene directa interacción con el <i>hardware</i> , siendo así el encargado de controlarlo para ejecutar correctamente las instrucciones externas.
<b>Frontend</b>	Consiste en la conversión de datos en una interfaz gráfica para que el usuario pueda ver e interactuar con la información de forma digital usando HTML, CSS y JavaScript.
<b>Hardware libre</b>	Dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público.
<b>IEEE</b>	Institute of Electrical and Electronics Engineers.
<b>INCAN</b>	Instituto de Cancerología.
<b>La melanina</b>	Es un pigmento que se halla en la mayor parte de los seres vivos. En los animales el pigmento se deriva del aminoácido tirosina. La forma más común de melanina es la eumelanina, un polímero negro-marrón de ácidos carboxílicos de dihidroxindol y sus formas reducidas.
<b>LoRaWAN</b>	Es una especificación para redes de baja potencia y área amplia, LPWAN (en inglés, Low Power Wide Area Network), diseñada específicamente para dispositivos de bajo consumo de



alimentación, que operan en redes de alcance local, regional, nacionales o globales.

**MCC/MNC**

Del inglés Mobile Country Code/Mobile Network Code, dos códigos numéricos usados conjuntamente para identificar el país y los operadores de telefonía móvil.

**MQTT**

Es un protocolo de red abierto, ligero, de publicación y suscripción estándar OASIS e ISO que transporta mensajes entre dispositivos.

**NFC**

Near field communication comunicación de campo cercano es una tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos.

**OMS**

Organización Mundial de la Salud.

**PATH**

Es una variable de entorno de los sistemas operativos POSIX y los Microsoft, en ella se especifican las rutas en las cuales el intérprete de comandos debe buscar los programas a ejecutar.

**Protocolo**

Es un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellas para transmitir información por medio de cualquier tipo de variación de una magnitud física.

<b>Real-Time Tracking</b>	Rastreo en tiempo real.
<b>RFID</b>	Identificación por radiofrecuencia es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas o transpondedores RFID.
<b>Sigfox</b>	Es un operador de red global francés fundado en 2009 que construye redes inalámbricas para conectar objetos de baja potencia, como medidores de electricidad y relojes inteligentes, que deben estar continuamente encendidos y emitiendo pequeñas cantidades de datos.
<b>Software libre</b>	Es todo software cuyo código fuente puede ser estudiado, modificado, y utilizado libremente con cualquier fin y redistribuido con cambios o mejoras o sin ellas.
<b>TLS</b>	Es un estándar que mantiene privada la conexión a Internet y verifica que los datos enviados entre dos sistemas (entre dos servidores, o un servidor y un cliente) estén cifrados y no se modifiquen.
<b>Ultravioleta</b>	Radiación electromagnética cuya longitud de onda está comprendida aproximadamente entre los 400 nm y los 15 nm.

**Web Server**

Es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente.

**Zigbee**

Es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (wireless personal área network, WPAN)



## RESUMEN

En ningún caso se pretende que este documento sustituya la opinión o trabajo de un médico profesional.

Se diseñará un prototipo de un dispositivo IoT bajo la arquitectura de tres capas, donde la capa de percepción se encargará de la adquisición de los rayos ultravioleta. Estos datos se transmitirán a través de una capa de red, que será el canal entre el dispositivo IoT y el cliente. Por último, la capa de aplicación, convertirá los datos a información entendible para el usuario final.

La capa de percepción se trabajará con la MSP8266 y el sensor SI1145, la cual también se trabajará con el lenguaje de programación Python 3, pero aplicado a microcontroladores (MicroPython).

En la capa de red, se creará una REST API donde se usará el Microframework Flask, diseñado en Python 3, para crear un canal entre el cliente y el dispositivo IoT.

La información se presentará con el framework Bootstrap, que está compuesto por código HTML, CSS, JavaScript, complementado con Jinja 2 (Python 3) para poderle presentar la información al usuario de forma adecuada. Los consejos médicos preventivos que se presentan en este documento fueron extraídos de la documentación de la Organización Mundial de la Salud (OMS).



# OBJETIVOS

## General

Diseñar un prototipo que mida el índice ultravioleta en un lugar definido para que un paciente pueda tomar las medidas preventivas adecuadas para evitar el cáncer de piel.

## Específicos

1. Encontrar las variables dependientes del índice ultravioleta que puedan perjudicar a las personas.
2. Explorar el costo y beneficio que el Open Source y Open Hardware pueden causar en una sociedad.
3. Diseñar la capa de percepción, capa de red y capa de aplicación que propone el modelo de una arquitectura IoT
4. Crear una interfaz gráfica de los datos adquiridos, para prevenir a los usuarios de los peligros del IUV.





## INTRODUCCIÓN

Al menos 300 casos de cáncer de piel se detectan al año en Guatemala. Esa cifra podría reducirse considerablemente si se practican hábitos de vida saludables y la detección temprana de la enfermedad. Sin embargo, el melanoma es la causa de la mayoría de las muertes en el mundo. Tanto el carcinoma como el melanoma lo causan los rayos tipo ultravioleta que el sol genera, las camas de bronceado o todo material que refleje la luz del sol.

Para medir los rayos ultravioleta la Organización Mundial de la Salud ha creado una escala de índice UV. De esta manera las personas pueden prevenir el cáncer porque se informa la escala a la que se están exponiendo.

Dependiendo la etapa donde se descubra el cáncer de piel así es la tasa de supervivencia. Por eso, todas las sociedades necesitan informarse acerca de cómo prevenir, lo antes posible, el cáncer de piel melanoma o carcinoma.

Con base en lo anterior, se ha propuesto un dispositivo que las personas pueden obtener para informarse sobre el índice ultravioleta al cual están expuestas. De esta manera podrán tomar las medidas necesarias para evitar formar parte de la población con cáncer de piel en el mundo. El Internet de las Cosas es la tecnología que se utilizará para solucionar este problema, gracias a su conexión entre capas con su capacidad de informar a las personas desde cualquier parte del mundo que tenga acceso a internet.



# **1. CONCEPTOS GENERALES SOBRE EL CÁNCER DE PIEL.**

El cáncer de la piel es una afección que consiste en la formación de células malignas (cancerosas) en los tejidos de la piel. El cáncer de piel se puede presentar en cualquier parte del cuerpo, pero es más común en la piel expuesta frecuentemente a la luz solar por los rayos ultravioleta (uv A, uv B), como la cara, el cuello, las manos y los brazos.

## **1.1. Tipos de cáncer de piel**

Existen tres tipos de cáncer de piel: Carcinoma basocelular, Carcinomas de células escamosas y melanoma.

### **1.1.1. El melanoma**

Es un cáncer que se origina en los melanocitos. Los tumores tipo melanoma, usualmente, son de color café o negro. Sin embargo, algunos melanomas no producen melanina, pueden lucir color rosado, café o incluso blanco. El sitio más común en los hombres es en el torso (pecho, espalda). En las mujeres en las piernas, el cuello y el rostro.

### **1.1.2. El carcinoma basocelular**

Es una forma de cáncer de piel de crecimiento lento. El tumor de carcinoma de células basales tiene apariencia de una verruga de color suave, perlado, sin escamas.

### **1.1.3. Carcinomas de células escamosas**

Aparecen en forma de protuberancias crecientes, a menudo de superficie áspera, o planos como manchas rojizas de la piel que crecen lentamente.

### **1.2. Signos de alerta que indican la existencia de cáncer de piel**

- Lunares con bordes irregulares (desiguales, dentados o indistintos).
- Lunares asimétricos (una mitad no coincide con la otra).
- Lunares que cambian de color, aumentan de tamaño o se elevan bruscamente.
- Lunares más grandes como la goma de borrar de un lápiz.
- Lunares que pican, duelen o sangran constantemente.
- Nuevos lunares, manchas o bultos rojos.
- Presencia de costras porque la lesión no cicatriza bien.

### **1.3. Factores de riesgo para cáncer de la piel**

- Piel clara.
- Antecedentes familiares de cáncer de piel.
- Antecedentes personales de cáncer de piel.
- Exposición al sol debido al trabajo o a las actividades recreativas.
- Antecedentes de quemaduras de sol, en especial, durante la infancia.
- Antecedentes de bronceado en camas solares.
- Piel que fácilmente se quema, enrojece o le salen pecas o que bajo el sol se siente adolorida.
- Ojos azules o verdes.
- Cabello rubio o pelirrojo.

- Cierta tipo de lunares o numerosos lunares.

#### 1.4. Antecedentes

La información del crecimiento de casos de cáncer de piel en Guatemala, se refleja a través de la información proporcionada del Instituto de Cancerología.

Tabla I. **Edad y sexo de los casos de cáncer detectados durante el año 2013. Registro Hospitalario del Incan – Guatemala**

GRUPO DE EDAD	SEXO		Total	%		
	Masculino	Femenino				
0 a 4	4	0.1%	-	-	<b>4</b>	<b>0.1%</b>
5 a 9			-	-	<b>0</b>	<b>0.0%</b>
10 a 14	1	0.0%	1	0.0%	<b>2</b>	<b>0.1%</b>
15 a 19	26	0.7%	24	0.7%	<b>50</b>	<b>1.4%</b>
20 a 24	31	0.9%	36	1.0%	<b>67</b>	<b>1.9%</b>
25 a 29	36	1.0%	64	1.8%	<b>100</b>	<b>2.8%</b>
30 a 34	27	0.8%	112	3.2%	<b>139</b>	<b>4.0%</b>
35 a 39	35	1.0%	180	5.1%	<b>215</b>	<b>6.1%</b>
40 a 44	40	1.1%	224	6.4%	<b>264</b>	<b>7.5%</b>
45 a 49	50	1.4%	262	7.5%	<b>312</b>	<b>8.9%</b>
50 a 54	71	2.0%	285	8.1%	<b>356</b>	<b>10.1%</b>
55 a 59	106	3.0%	293	8.3%	<b>399</b>	<b>11.4%</b>
60 a 64	132	3.8%	312	8.9%	<b>444</b>	<b>12.6%</b>
65 a 69	133	3.8%	231	6.6%	<b>364</b>	<b>10.4%</b>
70 a 74	108	3.1%	200	5.7%	<b>308</b>	<b>8.8%</b>
75 y +	219	6.2%	268	7.6%	<b>487</b>	<b>13.9%</b>
<b>Total</b>	<b>1019</b>	<b>29.0%</b>	<b>2,492</b>	<b>71.0%</b>	<b>3,511</b>	<b>100.0%</b>

\* Promedio para el sexo masculino = 59,5 ± 18,3 años

\* Promedio para el sexo femenino = 55,1 ± 15,5 años

Fuente: INCAN. Registro de Cáncer en Guatemala.

Tabla II. **Registro hospitalario del Incan - Guatemala. Residencia y sexo de los casos detectados durante el año 2013**

DEPARTAMENTO	SEXO				TOTAL	
	MASCULINO		FEMENINO		Número	%
	Número	%	Número	%		
Depto. de Guatemala	360	10.3%	819	23.3%	<b>1,179</b>	<b>33.6%</b>
Escuintla	54	1.5%	156	4.4%	<b>210</b>	<b>6.0%</b>
San Marcos	59	1.7%	144	4.1%	<b>203</b>	<b>5.8%</b>
Jutiapa	56	1.6%	120	3.4%	<b>176</b>	<b>5.0%</b>
Santa Rosa	45	1.3%	107	3.0%	<b>152</b>	<b>4.3%</b>
Huehuetenango	47	1.3%	103	2.9%	<b>150</b>	<b>4.3%</b>
Quetzaltenango	33	0.9%	110	3.1%	<b>143</b>	<b>4.1%</b>
El Quiché	46	1.3%	93	2.6%	<b>139</b>	<b>4.0%</b>
Chimaltenango	40	1.1%	97	2.8%	<b>137</b>	<b>3.9%</b>
Sacatepequez	34	1.0%	84	2.4%	<b>118</b>	<b>3.4%</b>
Suchitepequez	30	0.9%	87	2.5%	<b>117</b>	<b>3.3%</b>
El Petén	17	0.5%	71	2.0%	<b>88</b>	<b>2.5%</b>
Izabal	26	0.7%	57	1.6%	<b>83</b>	<b>2.4%</b>
Chiquimula	19	0.5%	60	1.7%	<b>79</b>	<b>2.3%</b>
Retalhuleu	21	0.6%	57	1.6%	<b>78</b>	<b>2.2%</b>
Jalapa	23	0.7%	54	1.5%	<b>77</b>	<b>2.2%</b>
Zacapa	22	0.6%	55	1.6%	<b>77</b>	<b>2.2%</b>
Baja Verapaz	21	0.6%	50	1.4%	<b>71</b>	<b>2.0%</b>
Sololá	19	0.5%	43	1.2%	<b>62</b>	<b>1.8%</b>
El Progreso	9	0.3%	48	1.4%	<b>57</b>	<b>1.6%</b>
Alta Verapaz	17	0.5%	40	1.1%	<b>57</b>	<b>1.6%</b>
Totonicapán	13	0.4%	20	0.6%	<b>33</b>	<b>0.9%</b>
Otros Países	5	0.1%	5	0.1%	<b>10</b>	<b>0.3%</b>
Belize	2	0.1%	7	0.2%	<b>9</b>	<b>0.3%</b>
Centroamérica	1	0.0%	5	0.1%	<b>6</b>	<b>0.2%</b>
<b>Total</b>	<b>1019</b>	<b>29.0%</b>	<b>2,492</b>	<b>71.0%</b>	<b>3,511</b>	<b>100.0%</b>

Fuente: INCAN. Registro de Cáncer en Guatemala.

Tabla III. Registro hospitalario del Incan, Guatemala. localizaciones (CIE-10) y sexo de los casos de cáncer detectados durante el año 2013

CIE-10		SEXO				TOTAL	
		MASCULINO		FEMENINO			
CODIGO	DESCRIPCION	Número	%	Número	%	Número	%
C00-D09	<b>TODAS LAS LOCALIZACIONES</b>	<b>1019</b>	<b>29.0%</b>	<b>2,492</b>	<b>71.0%</b>	<b>3,511</b>	<b>100.0%</b>
C01	Base De La Lengua	4	0.1%	1	0.0%	5	0.1%
C02	Partes No Especificadas Lengua	1	0.0%	11	0.3%	12	0.3%
C03	Encía	3	0.1%	3	0.1%	6	0.2%
C04	Suelo De La Boca	4	0.1%	3	0.1%	7	0.2%
C05	Paladar	10	0.3%	6	0.2%	16	0.5%
C06	Otras Partes De La Boca	6	0.2%	2	0.1%	8	0.2%
C07	Parótida	6	0.2%	13	0.4%	19	0.5%
C08	Otras Partes Glándulas Salivares Mayores	1	0.0%	-	-	1	0.0%
C09	Amígdala	5	0.1%	-	-	5	0.1%
C10	Orofaringe	1	0.0%	1	0.0%	2	0.1%
C11	Nasofaringe	2	0.1%	2	0.1%	4	0.1%
C13	Hipofaringe	5	0.1%	1	0.0%	6	0.2%
C15	Esófago	15	0.4%	14	0.4%	29	0.8%
C16	Estómago	82	2.3%	98	2.8%	180	5.1%
C17	Intestino Delgado	3	0.1%	1	0.0%	4	0.1%
C18	Colon	30	0.9%	43	1.2%	73	2.1%
C19	Unión Recto-Sigmoidea	3	0.1%	2	0.1%	5	0.1%
C20	Recto	18	0.5%	26	0.7%	44	1.3%
C21	Ano y Conducto Anal	6	0.2%	8	0.2%	14	0.4%
C22	Hígado y Vías Biliares Intrahep.	83	2.4%	77	2.2%	160	4.6%
C23	Vesícula Biliar	9	0.3%	18	0.5%	27	0.8%
C24	Otras Partes de Vías Biliares	3	0.1%	7	0.2%	10	0.3%
C25	Páncreas	12	0.3%	26	0.7%	38	1.1%
C26	Otras Partes Aparato Digestivo	6	0.2%	4	0.1%	10	0.3%
C30	Fosa Nasal y Oído Medio	3	0.1%	-	-	3	0.1%
C31	Senos Paranasales	3	0.1%	4	0.1%	7	0.2%
C32	Laringe	15	0.4%	4	0.1%	19	0.5%
C33	Tráquea	1	0.0%	-	-	1	0.0%
C34	Bronquios y Pulmón	25	0.7%	16	0.5%	41	1.2%
C38	Corazón, Mediastino y Pleura	5	0.1%	-	-	5	0.1%
C40	Huesos Extremidades	7	0.2%	15	0.4%	22	0.6%
C41	Otras Partes de Huesos	1	0.0%	2	0.1%	3	0.1%
C43	Melanoma De La Piel	16	0.5%	35	1.0%	51	1.5%
C44	Otros Tumores de la Piel	145	4.1%	187	5.3%	332	9.5%
C47	Sistema Nervioso Autónomo	3	0.1%	5	0.1%	8	0.2%
C48	Peritoneo y Retroperitoneo	4	0.1%	3	0.1%	7	0.2%
C49	Tejidos Blandos	27	0.8%	36	1.0%	63	1.8%
C50	Mama	1	0.0%	413	11.8%	414	11.8%
C51	Vulva	-	-	17	0.5%	17	0.5%
C52	Vagina	-	-	8	0.2%	8	0.2%
C53	Cérvix	-	-	757	21.6%	757	21.6%
D06	Carcinoma In Situ Cérvix	-	-	107	3.0%	107	3.0%
C54	Cuerpo Del Utero	-	-	72	2.1%	72	2.1%

Continuación de la tabla III.

CIE-10		SEXO				TOTAL	
		MASCULINO		FEMENINO			
CODIGO	DESCRIPCION	Número	%	Número	%	Número	%
C55	Utero	-	-	1	0.0%	1	0.0%
C56	Ovario	-	-	92	2.6%	92	2.6%
C58	Placenta	-	-	2	0.1%	2	0.1%
C60	Pene	13	0.4%	-	-	13	0.4%
C61	Próstata	123	3.5%	-	-	123	3.5%
C62	Testículo	66	1.9%	-	-	66	1.9%
C63	Otras Partes Aparato Genital Masculino	2	0.1%	-	-	2	0.1%
C64	Riñón	17	0.5%	24	0.7%	41	1.2%
C66	Uréter	1	0.0%	-	-	1	0.0%
C67	Vejiga Urinaria	16	0.5%	17	0.5%	33	0.9%
C68	Otras Partes Sistema Urinario	-	-	1	0.0%	1	0.0%
C69	Ojos	6	0.2%	2	0.1%	8	0.2%
C71	Encéfalo	20	0.6%	14	0.4%	34	1.0%
C73	Tiroides	14	0.4%	71	2.0%	85	2.4%
C76	Localizaciones Mal Definidas	5	0.1%	11	0.3%	16	0.5%
C77	Ganglio Linfático	1	0.0%	3	0.1%	4	0.1%
C80	Sitio Primario Desconocido	64	1.8%	104	3.0%	168	4.8%
C81	Enfermedad De Hodgkin	10	0.3%	9	0.3%	19	0.5%
C82	Linfoma No Hodgkin Folicular	1	0.0%	4	0.1%	5	0.1%
C83	Linfoma No Hodgkin Difuso	36	1.0%	31	0.9%	67	1.9%
C84	Linfoma de Células T	4	0.1%	7	0.2%	11	0.3%
C85	Linfoma No Hodgkin Otros Tipos	28	0.8%	18	0.5%	46	1.3%
C90	Mieloma Multiple	6	0.2%	7	0.2%	13	0.4%
C91	Leucemia Linfoide	7	0.2%	9	0.3%	16	0.5%
D00	Cavidad Bucal, Esófago y Estómago	-	-	1	0.0%	1	0.0%
D02	Sistema Respiratorio y Oído Medio	1	0.0%	-	-	1	0.0%
D03	Melanoma In Situ	-	-	3	0.1%	3	0.1%
D04	Carcinoma In Situ Piel	2	0.1%	4	0.1%	6	0.2%
D05	Carcinoma In Situ Mama	-	-	4	0.1%	4	0.1%
D07	Carcinoma In Situ Otros Organos Genitales	2	0.1%	5	0.1%	7	0.2%

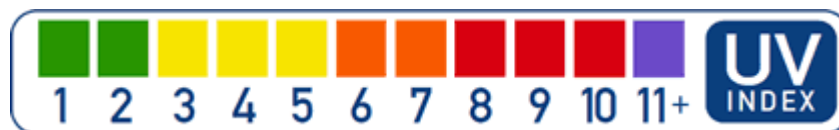
Fuente: INCAN. Registro de Cáncer en Guatemala.



## 1.5. ¿Qué es el índice UV?

Es un indicador de la intensidad de radiación ultravioleta.

Figura 1. Índice UV



Fuente: Trinitykck.org. *Burn Index Chart - Uv Index Chart Time To Burn Mossview*.  
<http://narta.innovations2019.org/burn-index-chart/>. Consulta: 20 de noviembre del 2019

Aunque los rayos UV no son visibles para los ojos humanos, afectan los ojos y la piel. La luz ultravioleta permite producir vitamina D, que es esencial para los humanos. En exceso, puede producir quemaduras solares de piel.

El UV se mide en mW por área, pero es difícil de medir en el área de la cabeza, especialmente si el valor del área está en cm, m o ft. Para facilitar el seguimiento de la seguridad UV, la OMS creó el Índice UV.

El SI1145 está calibrado para transmitir el índice UV en función de la luz visible e IR recibida, pero ¿qué significan esos números?

### 1.5.1. Escalas del índice Ultravioleta

Son un indicador con el objetivo de evitar una exposición perjudicial a la radiación UV.

### **1.5.1.1. 0 a 2: bajo**

Una lectura de Índice UV de 0 a 2 significa un bajo peligro de los rayos UV del sol para la persona promedio.

- Use gafas de sol en días soleados.
- Si se quema fácilmente, cúbrase y use un protector solar SPF 30+ de amplio espectro.
- Tenga cuidado con las superficies brillantes, como arena, agua y nieve, que reflejan los rayos UV y aumentan la exposición.

### **1.5.1.2. 3 a 5: moderado**

Una lectura del Índice UV de 3 a 5 significa un riesgo moderado de daño por la exposición al sol sin protección.

- Permanezca a la sombra cerca del mediodía cuando el sol está más fuerte.
- Si está al aire libre, use ropa protectora, un sombrero de ala ancha y gafas de sol que bloquean los rayos UV.
- Aplique generosamente filtro solar SPF 30+ de amplio espectro cada 2 horas, incluso en días nublados, y después de nadar o sudar.
- Tenga cuidado con las superficies brillantes, como arena, agua y nieve, que reflejan los rayos UV y aumentan la exposición.

### **1.5.1.3. 6 a 7: alto**

Una lectura del Índice UV de 6 a 7 significa un alto riesgo de daño por la exposición al sol sin protección. Se necesita protección contra el daño de la piel y los ojos.

- Reduzca el tiempo de exposición al sol entre las 10 a. m. Y las 4 p. m.
- Si está al aire libre, busque sombra y use ropa protectora, un sombrero de ala ancha y gafas de sol que bloquean los rayos UV.
- Aplique generosamente filtro solar SPF 30+ de amplio espectro cada 2 horas, incluso en días nublados, y después de nadar o sudar.
- Tenga cuidado con las superficies brillantes, como arena, agua y nieve, que reflejan los rayos UV y aumentan la exposición.

### **1.5.1.4. 8 a 10: muy alto**

Una lectura de Índice UV de 8 a 10 significa un riesgo muy alto de daño por exposición al sol sin protección. Tome precauciones adicionales porque la piel y los ojos desprotegidos se dañarán y pueden arder rápidamente.

- Minimice la exposición al sol entre las 10 a.m. y las 4 p.m.
- Si está al aire libre, busque sombra y use ropa protectora, un sombrero de ala ancha y gafas de sol que bloquean los rayos UV.
- Aplique generosamente filtro solar SPF 30+ de amplio espectro cada 2 horas, incluso en días nublados, y después de nadar o sudar.
- Tenga cuidado con las superficies brillantes, como arena, agua y nieve, que reflejan los rayos UV y aumentan la exposición.

### **1.5.1.5. 11 o más: extremo**

Una lectura de índice UV de 11 o más significa riesgo extremo de daño por exposición al sol sin protección. Tome todas las precauciones ya que la piel y los ojos sin protección pueden arder en minutos.

- Intente evitar la exposición al sol entre las 10 a.m. y las 4 p.m.
- Si está al aire libre, busque sombra y use ropa protectora, un sombrero de ala ancha y gafas de sol que bloquean los rayos UV.
- Aplique generosamente filtro solar SPF 30+ de amplio espectro cada 2 horas, incluso en días nublados, y después de nadar o sudar.
- Tenga cuidado con las superficies brillantes, como arena, agua y nieve, que reflejan los rayos UV y aumentan la exposición.

## **1.6. Efectos de la exposición a los rayos UV**

A continuación, se analizarán los efectos que tienen los rayos UV en la piel:

### **1.6.1. Cáncer de la piel**

La mayoría de los cánceres de piel aparecen después de los 50 años, pero el daño que provoca el sol en la piel comienza mucho antes. Por consiguiente, las medidas de protección para prevenir la posterior aparición del cáncer de la piel se deben tomar desde la niñez.

### **1.6.2. Envejecimiento prematuro**

La exposición al sol también causa el envejecimiento prematuro de la piel, una condición denominada envejecimiento por acción de sol, que es diferente del envejecimiento cronológico. Las personas que toman el sol con regularidad muestran señales de esta condición a una edad muy temprana, a menudo antes de los 30 años. El envejecimiento cronológico de la piel suele manifestarse después de los 40 años e incluso más tarde.

### **1.6.3. Cataratas y otras enfermedades de los ojos**

Las cataratas es una enfermedad de los ojos caracterizada por un cambio en la estructura del cristalino, que produce visión borrosa. Las personas que pasan más tiempo al sol pueden desarrollar cataratas más temprano que las demás. Las quemaduras solares en la córnea, los crecimientos en la superficie externa del ojo, las lesiones en la retina y otras enfermedades oculares se atribuyen, con mayor o menor grado de certeza, a la exposición a largo plazo a los rayos ultravioleta.

### **1.6.4. Daño al sistema inmunológico**

La piel es parte del sistema natural de defensa del cuerpo. Cuando la radiación UV suprime la respuesta inmunológica, la capacidad del cuerpo de combatir ciertas enfermedades, entre ellas el cáncer, se reduce.

## **1.7. La intensidad de la radiación UV**

La intensidad de la radiación UV puede ser peligrosa dependiendo de varios factores presentados a continuación.

### **1.7.1. La altura del sol**

Cuanto más alto esté el sol en el cielo, más intensa es la radiación UV. Así, la intensidad de la radiación UV varía según la hora del día y la época del año. Fuera de las zonas tropicales, las mayores intensidades de la radiación UV se producen cuando el sol alcanza su máxima altura, alrededor del mediodía solar durante los meses de verano

### **1.7.2. La reflexión por el suelo**

Diferentes tipos de superficies reflejan o dispersan la radiación UV en diversa medida; por ejemplo, la nieve reciente puede reflejar hasta un 80 % de la radiación UV; la arena seca de la playa, alrededor de un 15 %, y la espuma del agua del mar, alrededor de un 25 %

### **1.7.3. La latitud**

Cuanto más cerca del ecuador, más intensa es la radiación UV

### **1.7.4. La nubosidad**

La intensidad de la radiación UV es máxima cuando no hay nubes, pero puede ser alta incluso con nubes. La dispersión puede producir el mismo efecto que la reflexión por diferentes superficies, aumentando la intensidad total de la radiación UV.

### **1.7.5. La altitud**

A mayor altitud la atmósfera es más delgada y absorbe una menor proporción de radiación UV. Con cada 1000 metros de incremento de la altitud, la intensidad de la radiación UV aumenta en un 10 a 12 %.

### **1.7.6. El ozono**

El ozono absorbe parte de la radiación UV que podría alcanzar la superficie terrestre. La concentración de ozono varía a lo largo del año e incluso del día.





## **2. CONCEPTOS GENERALES DEL IOT**

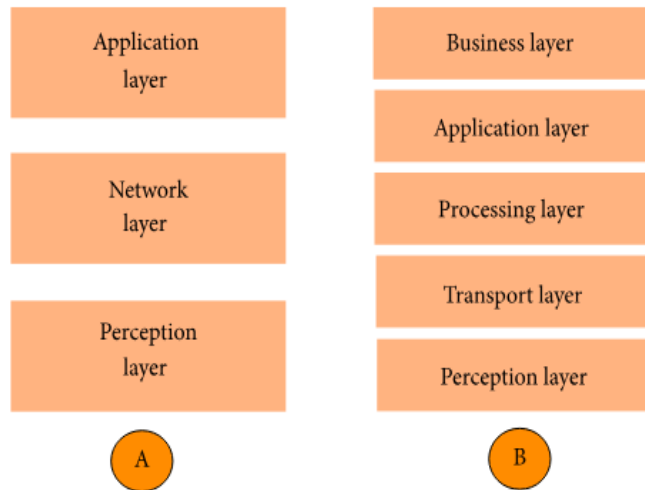
### **2.1. ¿Qué es Internet de las Cosas (IoT)?**

Es un término utilizado para describir un entorno en el que miles de millones de objetos, limitados en términos de recursos ("cosas"), están conectados a internet e interactúan de forma autónoma. Con tantos objetos conectados en soluciones de IoT, el entorno en el que se colocan se vuelve más inteligente. Un software, llamado middleware, desempeña un papel clave, es responsable de la mayor parte de la inteligencia en IoT, integrando datos de dispositivos, lo que les permite comunicarse y tomar decisiones basadas en los datos recopilados.

### **2.2. Arquitecturas de IoT**

De todas las arquitecturas de IoT investigadas, se eligieron las dos más populares.

Figura 2. **Arquitectura de IoT**



Fuente: ResearchGate. *Arquitectura de niebla de una puerta de enlace IoT inteligente*.  
[https://www.researchgate.net/figure/Fog-architecture-of-a-smart-IoT-gateway\\_fig8\\_312957467](https://www.researchgate.net/figure/Fog-architecture-of-a-smart-IoT-gateway_fig8_312957467).  
Consulta: 23 de enero de 2019.

### 2.2.1. **Arquitectura de tres capas**

La arquitectura más básica es una arquitectura de tres capas como se muestra en la figura 1. tiene tres capas que son: la percepción, la de red y las capas de aplicación.

- La capa de percepción es la capa física, que tiene sensores para detectar y recopilar información sobre el entorno. Detecta algunos parámetros físicos o identifica otros objetos inteligentes en el entorno.
- La capa de red es responsable de conectarse a otras cosas inteligentes, dispositivos de red o servidores. Sus características también se utilizan para transmitir y procesar datos de sensores.

- La capa de aplicación es responsable de entregar servicios específicos de la aplicación al usuario. Define varias aplicaciones en las que se puede implementar IoT, por ejemplo, hogares inteligentes, ciudades y salud inteligentes.

La arquitectura de tres capas define la idea principal del Internet de las cosas, pero no es suficiente para la investigación en IoT porque la investigación a menudo se centra en aspectos más finos del Internet de las cosas.

### **2.2.2. Arquitectura de cinco capas**

Las cinco capas son capas de percepción, transporte, procesamiento, aplicación y negocio, ver figura 1. El papel de las capas de percepción, aplicación es el mismo que la arquitectura con tres capas. Se resume la función de las tres capas restantes a continuación.

- La capa de transporte transfiere los datos del sensor desde la capa de percepción a la capa de procesamiento y viceversa a través de redes tales como inalámbrica, 3G, LAN, Bluetooth, RFID y NFC.
- La capa de procesamiento también se conoce como la capa de middleware. Almacena, analiza, procesa grandes cantidades de datos que provienen de la capa de transporte. Puede administrar y proporcionar un conjunto diverso de servicios a las capas inferiores. Emplea muchas tecnologías, como bases de datos, computación en la nube y módulos de procesamiento de *big data*.

- La capa de negocio administra todo el sistema de IoT, incluidas las aplicaciones, los modelos comerciales de beneficios y la privacidad de los usuarios. En ella se resuelven todos los “problemas” de más alto nivel de abstracción, como los modelos de negocio, la privacidad de los datos de usuario, se gestionan las aplicaciones y en general todo el modelo IoT.

### 2.2.3. Taxonomía del IoT

Se propone una taxonomía para la investigación en tecnologías de la IO. Esta taxonomía se basa en los elementos arquitectónicos de IoT como se presenta en la figura 3.

Figura 3. Taxonomy of research in IoT technologies



Figure 3: Taxonomy of research in IoT technologies.

Fuente: ResearchGate. *Taxonomía de la investigación en tecnologías IoT*.  
[https://www.researchgate.net/figure/Taxonomy-of-research-in-IoT-technologies\\_fig9\\_312957467](https://www.researchgate.net/figure/Taxonomy-of-research-in-IoT-technologies_fig9_312957467). Consulta: 23 de enero de 2019.

### 2.2.3.1. Percepción

El primer componente arquitectónico de IoT es la capa de percepción. Recopila datos utilizando sensores, que son los dispositivos más importantes del IoT. Hay varios tipos de sensores utilizados en diversas aplicaciones de IoT. El sensor más genérico disponible en la actualidad es el teléfono inteligente. El teléfono inteligente tiene muchos tipos de sensores integrados como el sensor de ubicación (GPS), sensores de movimiento (acelerómetro, giroscopio), cámara, sensor de luz, micrófono, sensor de proximidad y magnetómetro. Estos están siendo muy utilizados en diferentes aplicaciones de IoT.

Se están empezando a utilizar muchos otros tipos de sensores, como los sensores para medir la temperatura, la presión, la humedad, los parámetros médicos del cuerpo, las sustancias químicas, bioquímicas, señales neuronales. Una clase de sensores que se destacan, son los sensores infrarrojos que preceden a los teléfonos inteligentes. Ahora, se usan ampliamente en muchas aplicaciones de IoT: cámaras IR, detectores de movimiento, medición de la distancia a objetos cercanos, presencia de humo, gases y como sensores de humedad. La investigación se enfoca en el sensor ultravioleta en este documento.

Figura 4. **Smart watches**



Fuente: Fitbit. *Reloj inteligente*. <https://www.pebble.com>. Consulta: 23 de enero de 2019.

### 2.2.3.2. Preprocesamiento

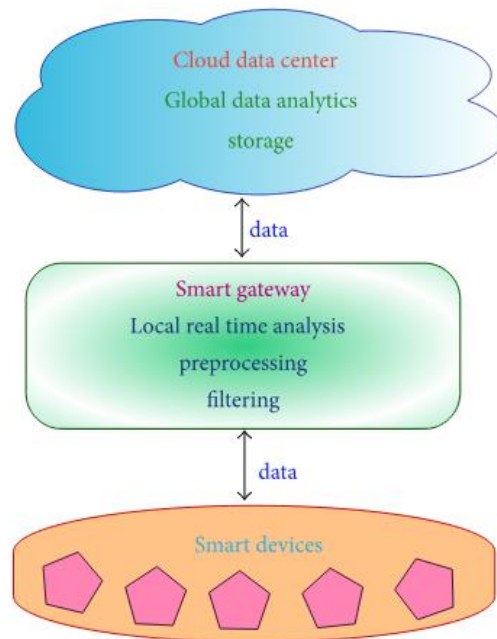
A medida que las cosas inteligentes recopilan una gran cantidad de datos de sensores, se requieren recursos de cómputo, almacenamiento para analizar, almacenar o procesar estos datos. Los recursos de cómputo y almacenamiento más comunes están basados en la nube porque la nube ofrece un manejo masivo de datos, escalabilidad y flexibilidad. Pero esto no será suficiente para cumplir los requisitos de muchas aplicaciones de IoT debido a las siguientes razones.

- **Movilidad:** su ubicación cambiante dificulta la comunicación con el centro de datos de la nube debido a las condiciones cambiantes de la red en diferentes ubicaciones.
- **Actualización en tiempo real:** comunicarse con la nube y obtener respuestas de vuelta lleva su tiempo. Las aplicaciones sensibles a la latencia, que necesitan respuestas en tiempo real, pueden no ser factibles con este modelo. Además, la comunicación puede tener pérdidas debido a los enlaces inalámbricos, lo que puede llevar a datos no confiables.
- **Escalabilidad:** más dispositivos significa más solicitudes a la nube, lo que aumenta la latencia.
- **Restricciones de energía:** la comunicación consume una gran cantidad de energía y los dispositivos IoT funcionan con baterías. Por lo tanto, debe optimizarse la comunicación.

Para resolver el problema de la movilidad, los investigadores han propuesto la computación en nube móvil (MCC). Pero todavía hay problemas asociados con la latencia y la potencia. MCC también tiene problemas de movilidad, como las condiciones de la red que cambian con frecuencia debido a que surgen problemas como el desvanecimiento de la señal y la degradación del servicio.

Como solución a estos problemas, se pueden llevar algunos recursos de cómputo y almacenamiento al borde de la red en lugar de confiar en la nube para todo. Este concepto se conoce como cómputo de niebla. La niebla se puede ver como una nube, que está cerca del suelo. Los datos se pueden almacenar, procesar, filtrar y analizar en el borde de la red antes de enviarlos a la nube a través de medios de comunicación costosos. Los paradigmas de niebla y nube van de la mano. Ambos son necesarios para el rendimiento óptimo de las aplicaciones IoT. Se puede emplear una puerta de enlace inteligente entre las redes subyacentes y la nube para realizar el cómputo de niebla como se muestra en la figura 5.

Figura 5. **Puerta de enlace inteligente para el procesamiento**



Fuente: ResearchGate. *Puerta de enlace inteligente para preprocesamiento*.  
[https://www.researchgate.net/figure/Smart-gateway-for-preprocessing\\_fig4\\_312957467](https://www.researchgate.net/figure/Smart-gateway-for-preprocessing_fig4_312957467).

Consulta: 5 de febrero de 2019.

### 2.2.3.3. **Comunicación (capa de red)**

Como el IoT está creciendo muy rápidamente, hay una gran cantidad de dispositivos inteligentes heterogéneos que se conectan a internet. Los dispositivos IoT funcionan con baterías, con recursos mínimos de cómputo y almacenamiento. Debido a su naturaleza restringida, hay varios desafíos de comunicación involucrados, que son los siguientes:

- Dirección e identificación: dado que millones de cosas inteligentes se conectarán a Internet, tendrán que identificarse a través de una dirección



única, según la cual se comuniquen entre sí. Para esto, se necesita un gran espacio de direccionamiento y una dirección única para cada objeto inteligente.

- Comunicación de baja potencia: la comunicación de datos entre dispositivos es una tarea que consume energía, especialmente, la comunicación inalámbrica. Por lo tanto, se necesita una solución que facilite la comunicación con un bajo consumo de energía.
- Protocolos de enrutamiento con bajo requerimiento de memoria y patrones de comunicación eficientes.
- Comunicación de alta velocidad.
- Movilidad de cosas inteligentes.

Los dispositivos IoT, normalmente, se conectan a Internet a través del stack de IP (Protocolo de Internet). Este stack es muy complejo, requiere una gran cantidad de energía, memoria de los dispositivos conectados. Los dispositivos IoT también pueden conectarse localmente a través de redes no IP, que consumen menos energía, y conectarse a internet a través de una puerta de enlace inteligente. Los canales de comunicación no IP, como Bluetooth, RFID y NFC, son bastante populares, pero están limitados en su alcance (hasta unos pocos metros).

Por lo tanto, sus aplicaciones están limitadas a pequeñas redes de área personal. Las redes de área personal (PAN) se están utilizando ampliamente en aplicaciones de IoT, como dispositivos portátiles conectados a teléfonos inteligentes. Para aumentar el alcance de dichas redes locales, era necesario

modificar la capa de IP para facilitar la comunicación de baja potencia utilizando la capa de IP. Una de las soluciones es 6LoWPAN, que incorpora IPv6 con redes de área personal de bajo consumo. El rango de un PAN con 6LoWPAN es similar a las redes de área local, y el consumo de energía es mucho menor.

Las principales tecnologías de comunicación utilizadas en el mundo de IoT son IEEE 802.15.4, Wi-Fi de baja potencia, 6LoWPAN, RFID, NFC, Sigfox, LoraWAN y otros protocolos propietarios para redes inalámbricas.

#### **2.2.3.3.1. IoT Network protocol stack**

El Grupo de trabajo de ingeniería de internet (IETF) ha desarrollado protocolos alternativos para la comunicación entre dispositivos IoT mediante IP, ya que IP es un estándar flexible y confiable. La Alianza del Protocolo de Internet para Objetos Inteligentes (IPSO) ha publicado varios informes técnicos que describen protocolos, estándares alternativos para las capas de la pila o capa de IP y una capa de adaptación adicional, que se utiliza para la comunicación entre objetos inteligentes.

- Capa física y Enlace (IEEE 802.15.4). El protocolo IEEE 802.15.4 está diseñado para permitir la comunicación entre dispositivos embebidos compactos, económicos de bajo consumo que necesitan una batería de larga duración. Define estándares, protocolos para la capa física, de enlace (MAC) de la capa de IP. Admite la comunicación de baja potencia junto con la comunicación de bajo costo, corto alcance. En el caso de dichos entornos con recursos limitados, se necesita un tamaño de trama pequeño, un ancho de banda bajo y una potencia de transmisión baja.

La transmisión requiere muy poca potencia (máximo un mW), que es solo el uno por ciento de la utilizada en redes Wi-Fi o celulares. Esto limita el rango de comunicación. Debido al rango limitado, los dispositivos tienen que funcionar de manera cooperativa para permitir el enrutamiento multihop en distancias más largas. Como resultado, el tamaño del paquete está limitado a 127 bytes solamente, y la velocidad de comunicación está limitada a 250 kbps. El esquema de codificación en IEEE 802.15.4 ha incorporado redundancia, lo que hace que la comunicación sea robusta, nos permite detectar pérdidas, permite la retransmisión de paquetes perdidos. El protocolo también admite direcciones de enlace cortas de 16 bits para reducir el tamaño del encabezado, los gastos generales de comunicación y los requisitos de memoria.

Los lectores pueden consultar la encuesta de Vasseur et al. para obtener más información sobre diferentes tecnologías físicas y de capa de enlace para la comunicación entre objetos inteligentes.

- Capa de Adaptación. IPv6 se considera el mejor protocolo para la comunicación en el dominio de IoT debido a su escalabilidad y estabilidad. Inicialmente, se pensaba que tales protocolos IP voluminosos no eran adecuados para la comunicación en escenarios con enlaces inalámbricos de baja potencia como IEEE 802.15.4.

6LoWPAN, un acrónimo de IPv6 sobre redes de área personal inalámbricas de baja potencia, es un estándar muy popular para la comunicación inalámbrica. Permite la comunicación mediante IPv6 a través del protocolo IEEE 802.15.4. Esta norma define una capa de adaptación entre la capa de enlace 802.15.4 y la capa de transporte. Los

dispositivos 6LoWPAN pueden comunicarse con todos los demás dispositivos basados en IP en Internet. La elección de IPv6 se debe al gran espacio de direccionamiento disponible en IPv6. Las redes 6LoWPAN se conectan a internet a través de una puerta de enlace (Wi-Fi o Ethernet), que también tiene soporte de protocolo para la conversión entre IPv4 e IPv6, ya que la implementación de internet en la actualidad es principalmente IPv4. Los encabezados de IPv6 no son lo suficientemente pequeños como para caber dentro de la pequeña MTU de 127 bytes del estándar 802.15.4. Por ello, comprimir y fragmentar los paquetes para llevar solo la información esencial es una optimización que realiza la capa de adaptación.

Específicamente, la capa de adaptación realiza las siguientes tres optimizaciones para reducir la sobrecarga de comunicación:

- 6LoWPAN define el encabezado de los paquetes IPv6 para reducir la sobrecarga de IPv6. Algunos de los campos se eliminan porque pueden derivarse de la información a nivel de enlace o pueden compartirse entre paquetes.
- Fragmentación: el tamaño mínimo de MTU (unidad de transmisión máxima) de IPv6 es 1280 bytes. Por otro lado, el tamaño máximo de un marco en IEEE 802.15.4 es de 127 bytes. Por ello, se necesita fragmentar el paquete IPv6. Esto se hace por la capa de adaptación.
- El reenvío de la capa de enlace 6LoWPAN también admite la malla en el enrutamiento, que se realiza en la capa de enlace utilizando direcciones cortas a nivel de enlace en lugar de en la

capa de red. Esta función se puede utilizar para comunicarse dentro de una red 6LoWPAN.

- Capa de red. La capa de red es responsable de enrutar los paquetes recibidos desde la capa de transporte. El grupo de trabajo IETF Routing over Low Power and Lossy Networks (ROLL) ha desarrollado un protocolo de enrutamiento (RPL) para redes Low Power and Lossy Networks (LLN).

Para tales redes, RPL es un protocolo de enrutamiento abierto, basado en vectores de distancia. Describe cómo se construye un gráfico acíclico dirigido orientado hacia el destino (DODAG) con los nodos después de que intercambian vectores de distancia. Se utiliza un conjunto de restricciones y una función objetivo para construir el gráfico con la mejor ruta. La función objetivo y las restricciones pueden diferir con respecto a sus requisitos. Por ejemplo, las restricciones pueden ser evitar los nodos alimentados por batería o preferir enlaces cifrados. La función objetivo puede apuntar a minimizar la latencia o el número esperado de paquetes que deben enviarse.

La realización de este gráfico se inicia desde el nodo raíz. La raíz comienza a enviar mensajes a los nodos vecinos, que luego procesan el mensaje, deciden si unirse o no según las restricciones, la función objetivo. Posteriormente, reenvían el mensaje a sus vecinos. De esta manera, el mensaje se desplaza hasta que se forman los nodos de la hoja y se forma un gráfico. Ahora, todos los nodos en el gráfico pueden enviar paquetes hacia arriba, salto por salto, a la raíz. Podemos realizar un algoritmo de enrutamiento punto a punto de la siguiente manera. Se

envían paquetes a un antepasado común, desde el cual viaja hacia abajo (hacia las hojas) para llegar al destino.

Para administrar los requisitos de memoria de los nodos, los nodos se clasifican en nodos de almacenamiento, no de almacenamiento en función de su capacidad para almacenar información de enrutamiento. Cuando los nodos están en un modo sin almacenamiento, se está construyendo una ruta descendente, la información de la ruta se adjunta al mensaje entrante y se reenvía hasta la raíz. La raíz recibe la ruta completa en el mensaje y envía un paquete de datos junto con el mensaje de ruta al destino, salto a salto. Pero hay una compensación aquí, porque los nodos que no almacenan necesitan más potencia y ancho de banda para enviar información de ruta adicional, no tienen la memoria para almacenar tablas de enrutamiento.

- Capa de transporte. TCP no es una buena opción para la comunicación en entornos de bajo consumo, tiene una gran sobrecarga debido a que es un protocolo orientado a la conexión. Es decir, se prefiere UDP porque es un protocolo sin conexión y tiene una sobrecarga baja.
- Capa de aplicación. La capa de aplicación es responsable del formato y presentación de los datos. La capa de aplicación en internet se basa típicamente en HTTP. Sin embargo, HTTP no es adecuado en entornos con recursos limitados porque es de naturaleza bastante detallada, entonces incurre en una gran sobrecarga de análisis. Se han desarrollado muchos protocolos alternativos para entornos de IoT como CoAP (Protocolo de aplicación restringido) y MQTT (Transporte de telemetría de cola de mensajes).

- Protocolo de aplicación restringido: CoAP se puede considerar como una alternativa a HTTP. Se utiliza en la mayoría de las aplicaciones de IoT. A diferencia de HTTP, incorpora optimizaciones para entornos de aplicación restringidos. Utiliza el formato de datos EXI (Intercambios XML eficientes), que es un formato de datos binarios y es mucho más eficiente en términos de espacio en comparación con el texto simple HTML / XML. Otras características compatibles son la compresión de encabezado, el descubrimiento de recursos, la configuración automática, el intercambio asíncrono de mensajes, el control de congestión y la compatibilidad con mensajes de multidifusión. Hay cuatro tipos de mensajes en CoAP: no confirmable, confirmable, restablecer (nack) y confirmación. Para una transmisión confiable a través de UDP, se utilizan mensajes confiables. La respuesta puede ser superpuesta en el propio reconocimiento. Además, utiliza DTLS (Datagram Transport Layer Security) por motivos de seguridad.
- Transporte de telemetría de cola de mensajes: MQTT es un protocolo de publicación / suscripción que se ejecuta sobre TCP. Fue desarrollado por IBM principalmente como un protocolo cliente / servidor. Los clientes son publicadores / suscriptores y el servidor actúa como un intermediario al que los clientes se conectan a través de TCP. Los clientes pueden publicar o suscribirse a un tema. Esta comunicación se realiza a través del intermediario cuyo trabajo es coordinar las suscripciones, también autenticar al cliente por seguridad. MQTT es un protocolo ligero, que lo hace adecuado para aplicaciones de IoT. Pero debido a que se ejecuta sobre TCP, no se puede usar con todos los tipos de aplicaciones IoT. Además, utiliza texto para los nombres de temas, lo que aumenta su sobrecarga.

- MQTT-S / MQTT-SN es una extensión de MQTT, que está diseñada para dispositivos de bajo consumo y bajo costo. Se basa en MQTT pero tiene algunas optimizaciones para las WSN de la siguiente manera. Los nombres de los temas se reemplazan por ID de temas, lo que reduce los gastos generales de transmisión. Los temas no necesitan registro pues están preregistrados. Los mensajes también se dividen para que solo se envíe la información necesaria. Además, para la conservación de energía, existe un procedimiento fuera de línea para los clientes que están en estado de suspensión. Los mensajes pueden ser buffere.

#### **2.2.3.3.2. Bluetooth de baja energía (BLE)**

Bluetooth Low Energy, también conocido como "Bluetooth Smart", fue desarrollado por Bluetooth Special Interest Group. Tiene un rango relativamente más corto y consume menos energía en comparación con los protocolos de la competencia. La pila del protocolo BLE es similar a la pila utilizada en la tecnología Bluetooth clásica. Tiene dos partes: controlador y host. La capa física y de enlace se implementan en el controlador. El controlador es típicamente un SOC (Sistema en Chip) con una radio. Las funcionalidades de las capas superiores están incluidas en el host. BLE no es compatible con Bluetooth clásico. Veamos las diferencias entre Bluetooth clásico y BLE.

La principal diferencia es que BLE no admite la transmisión de datos. En su lugar, admite la transferencia rápida de paquetes pequeños de datos (el tamaño del paquete es pequeño) con una velocidad de datos de 1 Mbps.

Hay dos tipos de dispositivos en BLE: maestro y esclavo. El maestro actúa como un dispositivo central que puede conectarse a varios esclavos.



Consideremos un escenario de IoT en el que un teléfono o una PC sirven como dispositivos principales y móviles, como un termostato, rastreador de ejercicios, reloj inteligente o cualquier dispositivo de monitoreo que actúe como esclavos. En tales casos, los esclavos deben ser muy eficientes energéticamente. Por lo tanto, para ahorrar energía, los esclavos están en modo de suspensión por defecto y se activan periódicamente para recibir paquetes del maestro.

En el Bluetooth clásico, la conexión está activada todo el tiempo, incluso si no hay transferencia de datos. Además, admite 79 canales de datos (ancho de banda de canal de 1 MHz) y una velocidad de datos de 1 millón de símbolos / s, mientras que BLE admite 40 canales con ancho de banda de canal de 2 MHz (el doble del Bluetooth clásico) y una velocidad de datos de 1 millón de símbolos / s. BLE es compatible con los requisitos de ciclo de trabajo bajo, su tamaño de paquete es pequeño, el tiempo necesario para transmitir el paquete más pequeño es tan pequeño como 80 s. La pila de protocolos BLE también admite comunicación basada en IP. Un experimento realizado por Siekkinen et al. registró el número de bytes transferidos por Joule para mostrar que BLE consume mucha menos energía en comparación con los protocolos de la competencia como Zigbee. La eficiencia energética de BLE es 2,5 veces mejor que Zigbee.

#### **2.2.3.3.3. Wi-Fi de baja potencia**

La alianza Wi-Fi ha desarrollado recientemente "Wi-Fi HaLow", que se basa en el estándar IEEE 802.11ah. Consume menos energía que un dispositivo Wi-Fi tradicional y también tiene un alcance mayor. Por esta razón, este protocolo es adecuado para las aplicaciones de Internet de las cosas. El alcance de Wi-Fi HaLow es casi el doble que la de Wi-Fi tradicional.

Igual que otros dispositivos Wi-Fi, los dispositivos que admiten Wi-Fi HaLow también admiten conectividad IP, lo cual es importante para las aplicaciones de IoT. Veamos las especificaciones del estándar IEEE 802.11ah. Este estándar fue desarrollado para lidiar con los escenarios de redes de sensores inalámbricos, donde los dispositivos tienen limitaciones de energía, requieren una comunicación de largo alcance. IEEE 802.11ah opera en la banda de subgigahertz (900 MHz). Debido a la frecuencia relativamente baja, el rango es más amplio, las ondas de mayor frecuencia sufren una mayor atenuación. Es posible ampliar el rango (actualmente 1 km) reduciendo aún más la frecuencia. Sin embargo, la tasa de datos también será menor y, por lo tanto, la compensación no está justificada. IEEE 802.11ah también está diseñado para soportar grandes redes en forma de estrella, donde muchas estaciones están conectadas a un solo punto de acceso.

#### **2.2.3.3.4. Zigbee**

Se basa en el estándar de protocolo de comunicación IEEE 802.15.4 y se utiliza para redes de área personal o PAN. El estándar IEEE 802.15.4 tiene MAC, físicas de baja potencia y ya se ha explicado. Zigbee fue desarrollado por la alianza Zigbee, que funciona para soluciones de comunicación confiables, de bajo consumo energético y baratas. El rango de comunicación del dispositivo Zigbee es muy pequeño (10 a 100 metros). Los detalles de la red y las capas de aplicación también están especificados por el estándar Zigbee. A diferencia de BLE, la capa de red aquí proporciona el enrutamiento multihop.

Hay tres tipos de dispositivos en una red Zigbee: FFD (Dispositivo totalmente funcional), RFD (Dispositivo funcional reducido) y un coordinador de Zigbee. Un nodo FFD también puede actuar como un enrutador. Zigbee es

compatible con topologías de estrella, árbol y malla. El esquema de enrutamiento depende de la topología. Otras características de Zigbee son el descubrimiento y el mantenimiento de rutas, el soporte para nodos que se unen o salen de la red, direcciones cortas de 16 bits y enrutamiento multihop.

El marco para la comunicación y el desarrollo distribuido de aplicaciones lo proporciona la capa de aplicación. La capa de aplicación consta de objetos de aplicación (APO), subcapa de aplicación (APS) y un objeto de dispositivo Zigbee (ZDO). Los APO se extienden por los nodos de la red. Estas son piezas de software que controlan algunos dispositivos de hardware subyacentes (ejemplos: conmutador, transductor). El dispositivo, los servicios de administración de red son provistos por el ZDO, que luego son utilizados por los APO. Los servicios de transferencia de datos son proporcionados por la subcapa de aplicación a los APO y ZDO. Es responsable de la comunicación segura entre el objeto de la aplicación.



### **3. CAPA FÍSICA O DE PERCEPCIÓN**

La capa de percepción es la capa física, esta posee sensores para detectar y recopilar información sobre el entorno. Detecta algunos parámetros físicos o identifica otros objetos inteligentes en el entorno.

En este capítulo se presentarán los sensores utilizados para la adquisición de rayos ultravioleta.

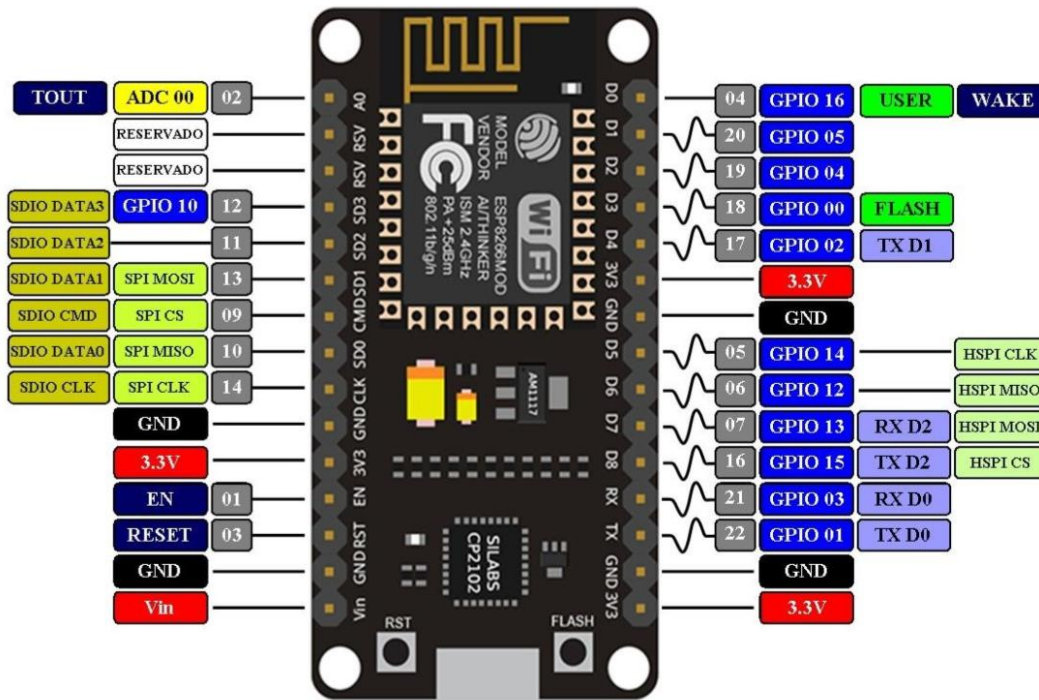
#### **3.1. NodeMCU**

Es una plataforma IoT de código abierto. Incluye el firmware que se ejecuta en el SoC Wi-Fi ESP8266 de Espressif Systems y el hardware que se basa en el módulo ESP-12. El término "NodeMCU" se refiere al firmware en lugar de a los kits de desarrollo. El firmware utiliza el lenguaje Lua. Se basa en el proyecto eLua y se basa en el SDK no operativo de Espressif para el ESP8266. Utiliza muchos proyectos de código abierto, como lua-cjson, y spiffs.

#### **3.2. Datasheet NodeMCU**

A continuación, se muestra el diagrama de los pines de la plataforma que se usará como firmware de recopilación y procesamiento de datos de la capa de recepción.

Figura 6. Diagrama de los pines de la plataforma



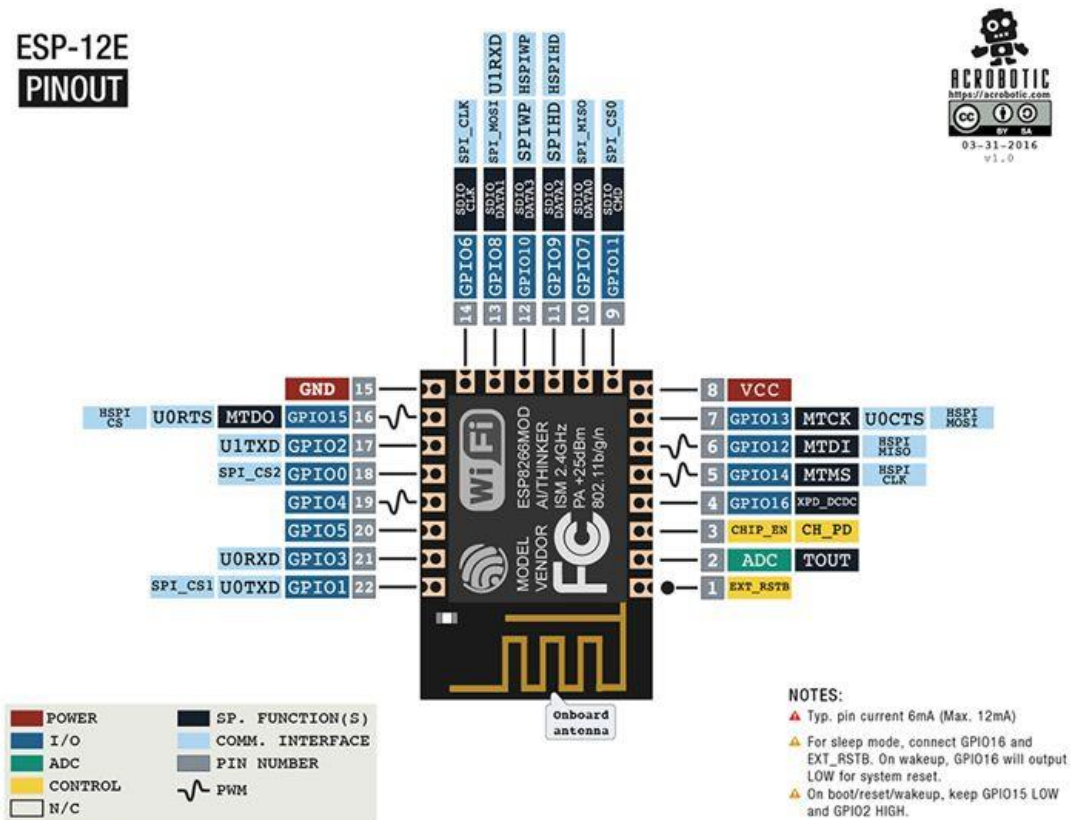
Vin	Alimentación Externa (de 5V a 10V)
3.3V	Alimentación Interna (desde la placa a dispositivos)
GND	Tierra (GND Ground)
GPIO	Pin de entrada/salida +3.3V (GPIO General Purpose Input/Output)
ADC	Pin de salida analógica (El rango es entre 0V y 1V dividido en 1023 intervalos).
SPI	Bus SPI (Serial Peripheral Interface)
HSPI	Bus HSPI (Hardware-Serial Peripheral Interface)
SDIO	Pines para inicio del ESP8266 desde una tarjeta SD
TX/RX	Comunicación Serie Tx/Rx Los pines GPIO01 y GPIO02 están conectados al puerto MicroUSB a través del convertidor UART.

Fuente: elaboración propia.

### 3.3. ESP8266

Es un chip de bajo costo Wi-Fi con una pila TCP/IP completa y un microcontrolador, fabricado por Espressif, una empresa ubicada en Shanghai, China. Este chip está ensamblado en el SOC NodeMCU.

Figura 7. Diagrama de ESP8266



Fuente: Electronilab. *NodeMCU – Board de desarrollo con módulo ESP8266 WiFi y Lua.*  
<https://electronilab.co/tienda/nodemcu-board-de-desarrollo-con-esp8266-wifi-y-lua/>. Consulta: 5 de febrero de 2019.

### 3.4. Conexión entre ESP8266 y NodeMCU

Sabiendo que la ESP8266 está implementada dentro de la NodeMCU, a continuación, se muestra el diagrama interno de las conexiones de la tarjeta.

Tabla IV. Diagrama interno entre ESP8266 y NodeMCU

Pin NodeMCU	Pin ESP8266
D0	GPIO16
D1	GPIO5
D2	GPIO4
D3	GPIO0
D4	GPIO2
D5	GPIO14
D6	GPIO12
D7	GPIO13
D8	GPIO15
D9 / RX	GPIO3
D10 / TX	GPIO1
D11 / SD2	GPIO9
D12 / SD3	GPIO10

Fuente: elaboración propia.

### 3.5. Instalación de *drivers* en Windows

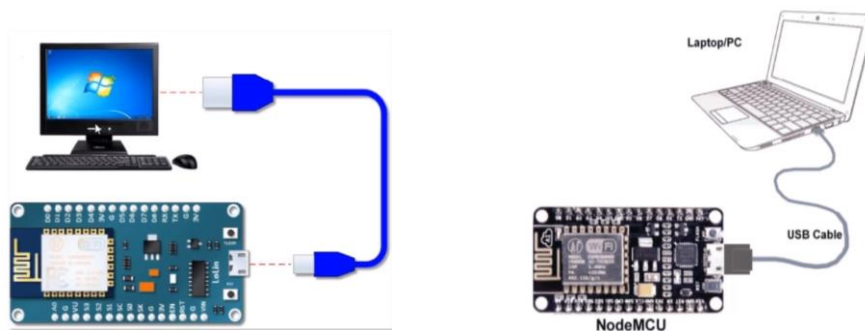
Los *drivers* sirven para que el sistema operativo reconozca y permita trabajar con los diferentes dispositivos. Para esto, el ordenador debe reconocer la tarjeta NodeMCU con la cual se trabajará.



### 3.5.1. Conexión

Para que el *driver* pueda instalarse en la PC, primero se conecta el dispositivo al ordenador, como se muestra en las siguientes imágenes.

Figura 8. **Manera correcta de conectar el dispositivo**



Fuente: Electronilab. *NodeMCU – Board de desarrollo con módulo ESP8266 WiFi y Lua.*  
<https://electronilab.co/tienda/nodemcu-board-de-desarrollo-con-esp8266-wifi-y-lua/>. Consulta: 5 de febrero de 2019.

### 3.5.2. Instalación de *drivers*

Para acceder a los *drivers* de la Node MCU se debe tomar en cuenta el sistema operativo que utilizará (para este casi será Windows 10) se deberá acceder el siguiente enlace:

- <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>

Figura 9. **Descarga de los controladores**

Download for Windows 10 Universal (v10.1.7)

Platform	Software	Release Notes
 Windows 10 Universal	<a href="#">Download VCP (2.3 MB)</a>	<a href="#">Download VCP Revision History</a>

Fuente propia

Figura 10. **Ubicación del puerto**



Fuente: elaboración propia.

### 3.5.3. **Instalación de *driver* equivalente**

Si no se puede instalar el *driver* anterior, se descargará su equivalente en el siguiente *link*:

- <https://www.geekfactory.mx/download/driver-ch340-windows/>

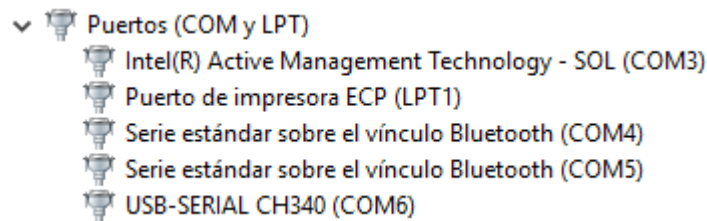
Figura 11. **Descarga del controlador equivalente**

Driver CH340 Windows  
by Admin | Nov 23, 2017 | 39 comments

Version	1117
Download	128896
File Size	237.62 KB
Create Date	23 noviembre, 2017
Download	

Fuente: elaboración propia.

Figura 12. **Muestra del controlador en el ordenador**



Fuente: elaboración propia, empleando PC local.

### 3.6. Putty

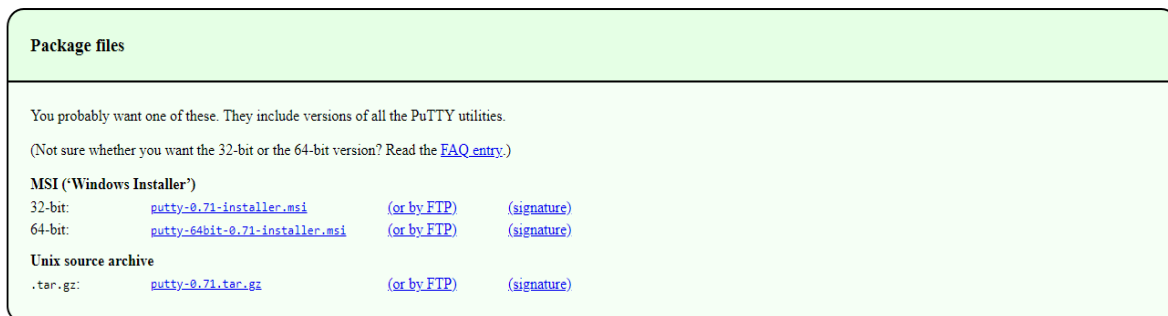
Es un cliente SSH, Telnet, rlogin, y TCP raw con licencia libre. Disponible originalmente solo para Windows, ahora también está disponible en varias plataformas Unix, se está desarrollando la versión para Mac OS clásico, Mac OS X. Otras personas han contribuido con versiones no oficiales para otras plataformas, como Symbian para teléfonos móviles. Es software beta escrito, mantenido principalmente por Simon Tatham, open source y licenciado bajo la Licencia MIT.

### 3.6.1. Instalación de PuTTY

Descargar la opción según se adecúe a su ordenador e instalarlo. Acceder al enlace a continuación:

- <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Figura 13. Instalación del PuTTY



The screenshot shows a section titled "Package files" with a light green background. Below the title, it says "You probably want one of these. They include versions of all the PuTTY utilities." and "(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)". There are two sections: "MSI ('Windows Installer')" and "Unix source archive".

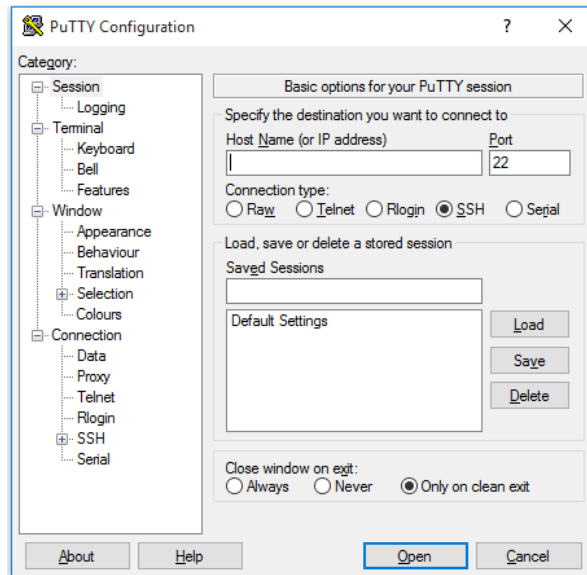
Package type	File name	FTP link	Signature link
MSI ("Windows Installer")	32-bit: <a href="#">putty-0.71-installer.msi</a>	<a href="#">(or by FTP)</a>	<a href="#">(signature)</a>
	64-bit: <a href="#">putty-64bit-0.71-installer.msi</a>	<a href="#">(or by FTP)</a>	<a href="#">(signature)</a>
Unix source archive	.tar.gz: <a href="#">putty-0.71.tar.gz</a>	<a href="#">(or by FTP)</a>	<a href="#">(signature)</a>

Fuente: PuTTY. *Download PuTTY: latest release (0.73)*.

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>. Consulta: 23 de marzo de 2019.

Cuando lo haya instalado correctamente, se debe dirigir a la pestaña de inicio de Windows y escribir "Putty". Al ejecutarlo, debe aparecer la figura 14.

Figura 14. Configuración del PuTTY



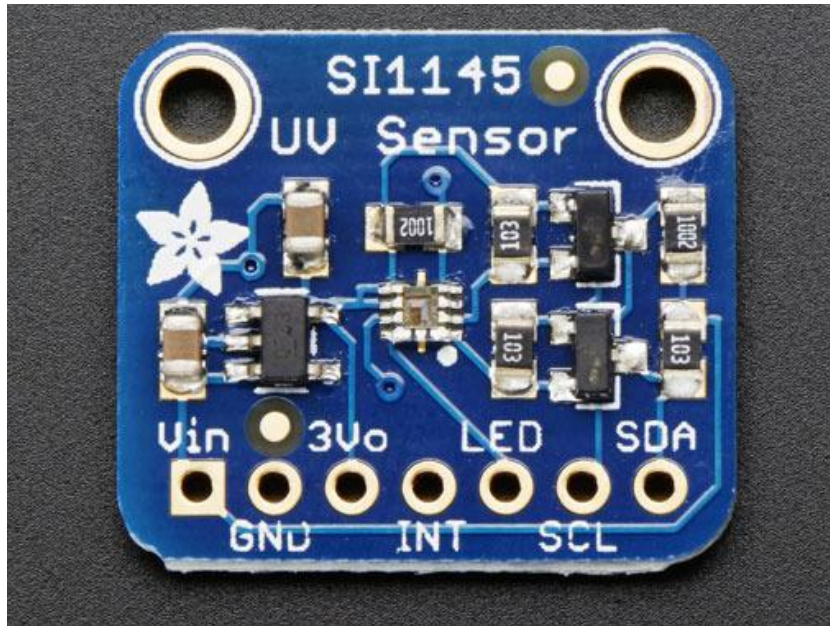
Fuente: PuTTY. *Download PuTTY: latest release (0.73)*.

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>. Consulta: 23 de marzo de 2019.

### 3.7. El SI1145

Es un nuevo sensor de SiLabs con un algoritmo de detección de luz calibrado que puede calcular el índice UV. No contiene un elemento sensor de UV real, sino que lo aproxima a la luz visible e IR del sol. Es un sensor digital que funciona con I2C, por lo que cualquier microcontrolador puede usarlo. El sensor también tiene elementos de detección IR e IR individuales para que pueda medir casi cualquier tipo de luz: si necesita precisión Lux medida de verificación del TSL2561. Si la persona tiene conocimientos avanzados, puede conectar un led IR al pin del led y también usar la capacidad del sensor de proximidad básico que está en el SI1145.

Figura 15. **Placa de conexiones Adafruit SI1145 - Índice UV/IR/ Sensor Visible**



Fuente: Adafruit. *SI1145 Digital UV Index / IR / Visible Light Sensor*.  
<https://www.adafruit.com/product/1777>. Consulta: 24 de marzo de 2019.

El SI1145 es un sensor I2C, por lo que usa dos terminales.

### 3.7.1. Pines de potencia

Puede alimentar el SI1145 de 3 a 5 VDC, hay un regulador de nivel incorporado para bajar el voltaje de manera segura a 3,3 V.

- Vin: esta es la entrada al regulador de voltaje. Potencial de 3 a 5 VDC. Tiene protección de polaridad inversa.
- GND: este es el pin de señal de tierra y potencial que se debe conectar al pin de tierra del microcontrolador

- 3Vo: esta es la salida del regulador a bordo. Si necesita 3,3 VDC, puede extraer hasta 100 mA de este pin.

### **3.7.2. Pines Extra**

Se trata de pines a los que no es necesario conectarse, pero pueden usarse si desea utilizar las funciones más avanzadas del chip.

- INT - esto es la salida de interrupción del chip, con un pull-up de 10 K a 3,3 V. Puede consultar este pin para ver cuándo se realiza el siguiente conjunto de conversiones. Esta biblioteca no lo usa.
- Led: si desea probar el uso de las capacidades de proximidad del SI1145, puede vincular un led IR a este pin. Cualquier led IR funcionará. Es un pin de colector abierto, así que conecte el pin de cátodo (-) a este pin, y el pin de ánodo (+) del led a 3 a 5 VDC. Puede consumir hasta 300 mA en una espiga, así que ten en cuenta que si usas este pin.

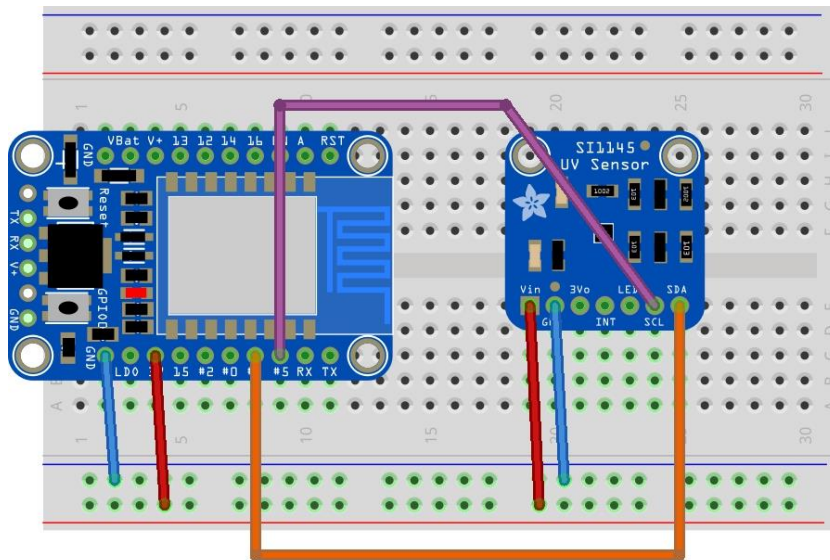
### **3.7.3. I2C Pins**

Si desea leer los datos del sensor, debe usar estos pines. Este sensor utiliza I2C clásico para que cualquier microcontrolador con capacidad I2C pueda usarlo. Estos pines están desplazados en el nivel para que pueda usar la lógica de 3 o 5 V de forma segura. Hay 10 K pullups en ambos pines. Puede compartir los pines i2c con muchos otros sensores siempre que no usen la misma dirección: el SI1145 usa la dirección de 7 bits 0x60.

- SCL: este es el pin del reloj i2c, conéctelo al pin maestro del reloj I2C del microcontrolador.

- SDA: este es el pin de datos i2c, conéctelo al pin maestro de datos del microcontrolador I2C.

Figura 16. Diagrama de conexión



Fuente: ESP8266 Learning. *Si1145 sensor and ESP8266 example*.

<http://www.esp8266learning.com/si1145-sensor-and-esp8266-example.php>. Consulta: 25 de marzo de 2019.

### 3.8. NodeMCU Flasher

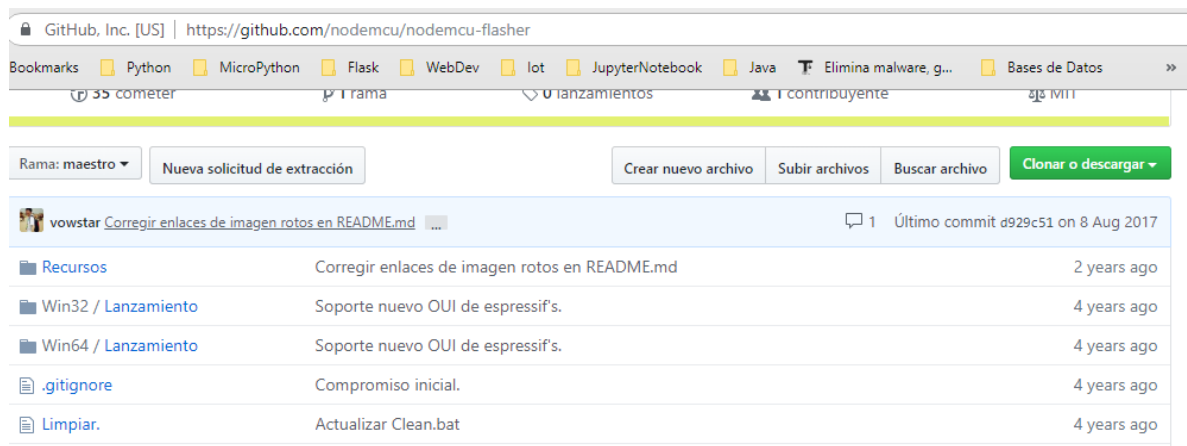
Es un programador de firmware para NodeMCU Devkit v 0,9 y puede usarlo para programar su propia tarjeta ESP8266. Debe establecer los pines GPIO0 en low para que NodeMCU Flasher pueda programar el firmware automáticamente.

Este archivo con extensión .exe lo puede descargar del siguiente repositorio: <https://github.com/nodemcu/nodemcu-flasher>, para este proyecto de



graduación se usará la versión para un Windows 10 de 64 bits, por lo que se usará el archivo Win64/lanzamiento.

Figura 17. Repositorio

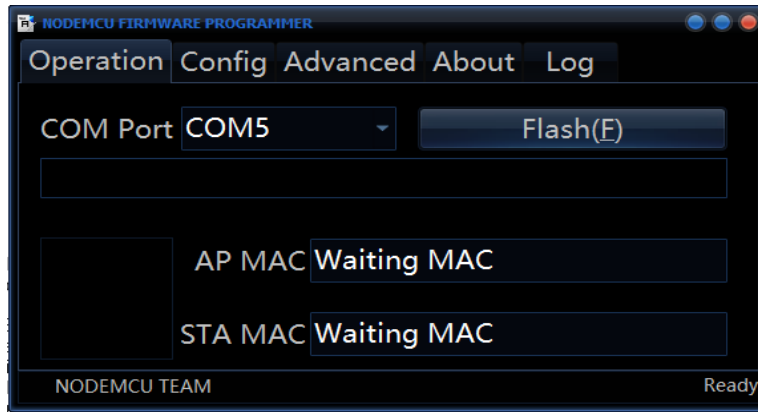


Fuente: GitHub. *Nodemcu-flasher*. [github.com/nodemcu/nodemcu-flasher](https://github.com/nodemcu/nodemcu-flasher). Consulta: 26 de marzo de 2019.

### 3.8.1. Interfaz gráfica NodeMCU Flasher

Al descargar y ejecutar el archivo ESP8266Flasher.exe debe mostrarse una interfaz gráfica como se presenta a continuación:

Figura 18. Interfaz gráfica del ESP8266



Fuente: elaboración propia, empleando PC local.

Ir a la parte de configuración para poder instalarle el lenguaje de programación que se usará para que pueda procesar la información que se ingresará (siguiente capítulo) a través del sensor SI1145.

Figura 19. Configuración del ESP8266



Fuente: elaboración propia, empleando PC local.

### 3.9. MicroPython

MicroPython es un pequeño pero eficiente interprete del Lenguaje de Programación Python 3 que incluye un subconjunto mínimo de librerías y que, además, está optimizado para que pueda correr en microcontroladores, ambientes restringidos.

Con MicroPython tienes la posibilidad de escribir códigos más simples, en lugar de usar Lenguajes de Programación de más bajo nivel como C o C++, que es el que utiliza Arduino, por ejemplo.

Hay algunas características que MicroPython tiene y es lo que lo hace único y diferente de otros sistemas embebidos:

- Tiene una REPL Interactiva (Read-Eval-Print).

Que es un pequeño programa que Lee e interpreta los comandos del usuario, los evalúa y después imprime el resultado. Esto permite conectar alguna tarjeta (microcontrolador que soporte Python) y esta tiene que ejecutar el código sin necesidad de compilar ni cargar el programa.

- Muchas librerías

Así como el Lenguaje de Programación Python cuenta con un sin fin de librerías para la ejecución de tareas, MicroPython también cuenta con bastantes paquetes para ahorrar trabajo. Es posible ejecutar análisis de datos JSON desde un servicio web, búsqueda de texto en expresiones regulares o hasta levantar un *socket* dentro de una red tan solo con las funciones ya precargadas.

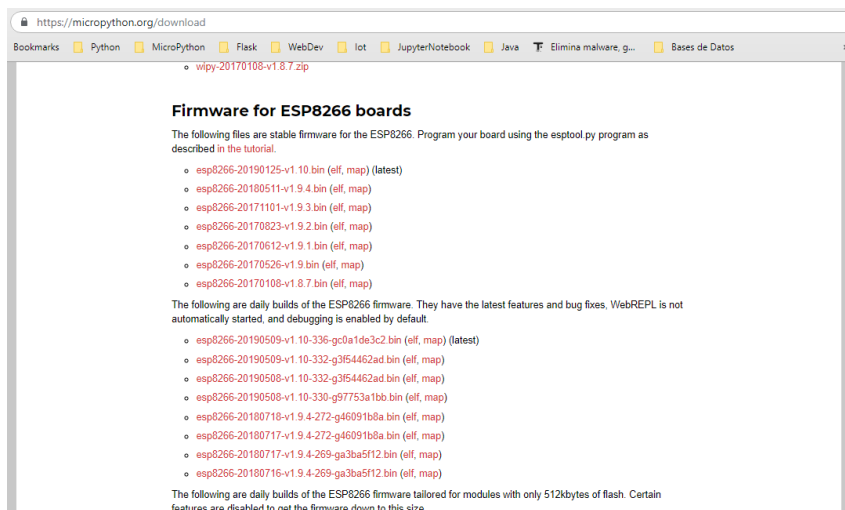
- Extensibilidad

Para los usuarios avanzados de MicroPython, pueden extender de Python a funciones de más bajo nivel como C o C++, pudiendo mezclar códigos que requieran de ejecución más rápida a bajo nivel con MicroPython que es de más alto nivel.

### 3.9.1. Firmware para la placa ESP8266

Se ha elegido MicroPython como lenguaje de programación para procesar la capa de percepción, por lo que se descargarán de su página oficial sus archivos de instalación para la NodeMCU, se debe acceder al siguiente *link* de descargas: <http://micropython.org/download> luego se debe ubicar en la sección de la ESP8266 para descargar la última versión del software que, en este caso, será la esp8266-20190125-v1.10.bin

Figura 20. Descarga del firmware

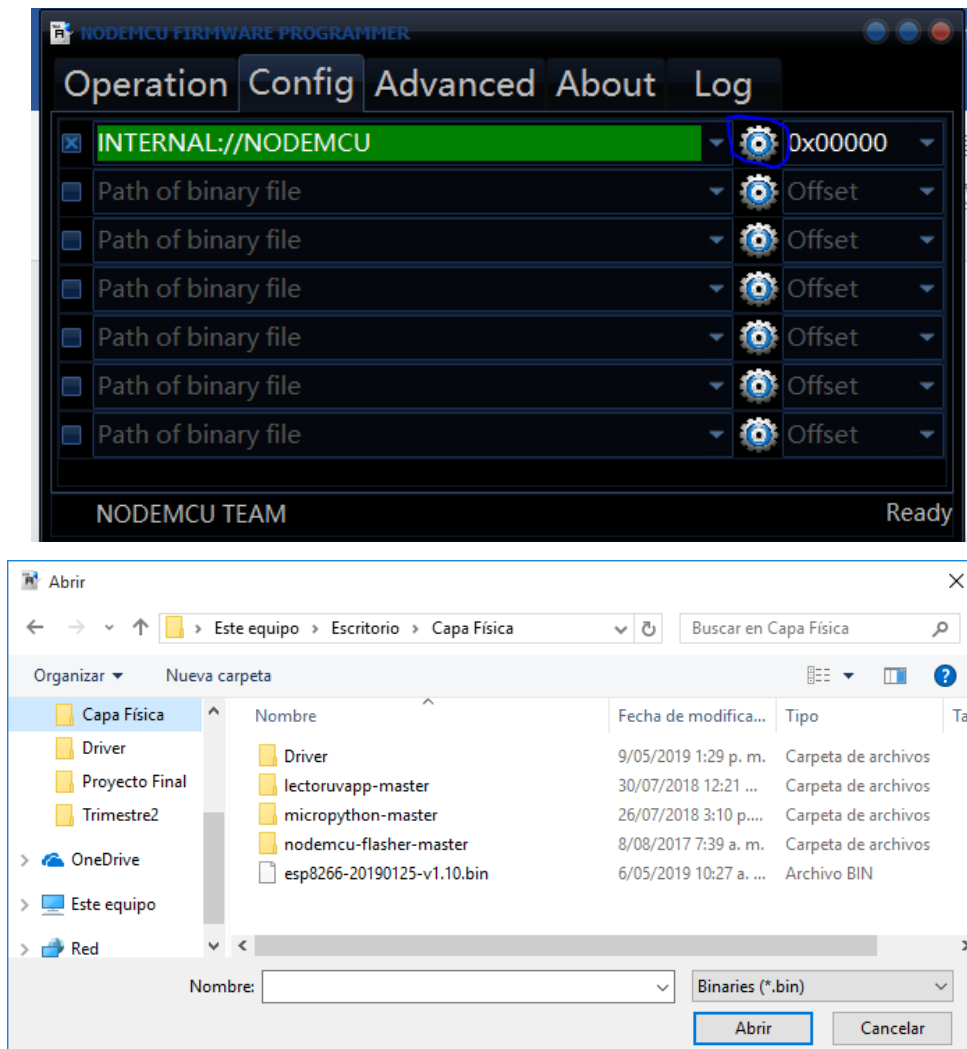


Fuente: elaboración propia, empleando Pc local.

### 3.9.2. Instalación del Firmware al SOC ESP8266

Para ingresar el firmware al SOC debe abrir el NodeMCU Flasher, ingresar a la sección de configuración e ir a la parte de ajustes para ingresar el archivo descargado en la sección anterior llamado esp8266-20190125-v1.10.bin.

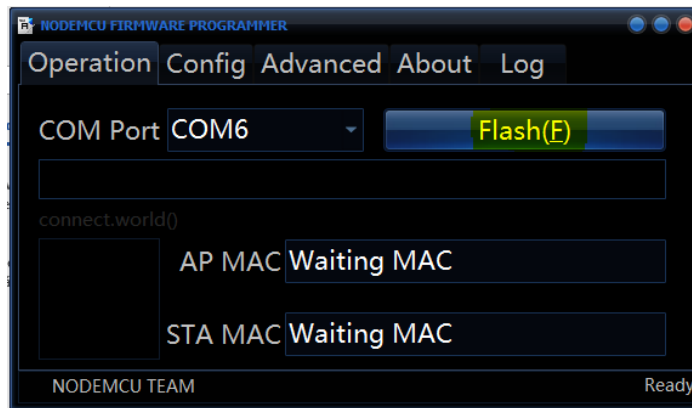
Figura 21. Configuración del firmware



Fuente: elaboración propia, empleando Pc local.

Luego ingresar al *flash* el archivo .bin y almacenarlo en la memoria *flash* de la MSP8266

Figura 22. **Configuración del firmware parte 2**



Fuente: elaboración propia, empleando Pc local.

### 3.10. Python 3

Python es un lenguaje de programación interpretado, su filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, soporta programación orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

### 3.10.1. Descarga de Python 3

Para encontrar las últimas versiones de Python adecuadas a su ordenador, ingresar el siguiente *link*: <https://www.python.org/downloads/> e instalar el archivo ejecutable en su ordenador.

Figura 23. Descarga de Python 3



Fuente: elaboración propia, empleando Pc local.

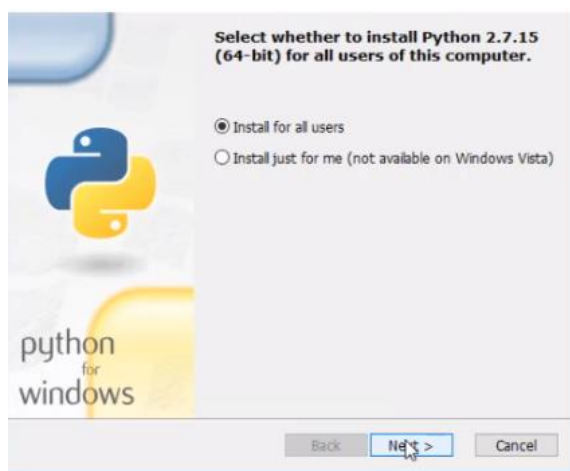
### 3.10.2. Instalación de Python 3

Luego de descargar el archivo de la última versión de Python debe abrir el archivo y ejecutarlo, como en las siguientes imágenes de ejemplo.

- Se instalará Python para todos los usuarios.
- Se agrega Python al PATH.

- Se elige la segunda opción: “Entire feature Will be installed on local hard drive”.

Figura 24. Instalación de Python 3



(1)



(2)



(3)



(4)

Fuente: elaboración propia, empleando Pc local.



### 3.11. Esptool.py

Una utilidad basada en Python, de código abierto, independiente de la plataforma, para comunicarse con el cargador de arranque ROM en los chips Espressif ESP8266 y ESP32.

Esptool.py fue iniciado por Fredrik Ahlberg (@ themadinventor ) como un proyecto comunitario no oficial. Ahora, también es apoyado por Espressif. El mantenedor primario actual es Angus Gratton (@ projectgus ).Esptool.py es software libre bajo una licencia GPLv2.

#### 3.11.1. Instalación de Esptool.py

Esta sección es para los usuarios que no quisieran utilizar NodeMCU flasher publicada en la sección 3.8. Para instalar Esptool.py ya sea en Windows o Linux, puede acceder a los siguientes *links*:

- <https://github.com/espressif/esptool>
- <https://docs.micropython.org/en/latest/esp8266/tutorial/intro.html?highlight=esptool>

Figura 25. Adquisición de datos desde la ESP8266 y el SI1145

```
main.py 797 Bytes Edit
1 import machine
2 import si1145
3 import time
4 import ujson
5 import urequests
6
7
8 class Medicion():
9     def __init__(self,uv,ir,visible):
10         self.uv = uv
11         self.ir = ir
12         self.visible = visible
13
14     def __repr__(self):
15         return '{"uv": %f , "ir": %f , "visible": %f }' % (self.uv, self.ir, self.visible)
16
17 i2c = machine.I2C(
18     sda=machine.Pin(0),
19     scl=machine.Pin(2))
20 sensor = si1145.SI1145(i2c=i2c)
21
22 lecturas = []
23
24 for i in range(10):
25     lectura = Medicion(sensor.read_uv,sensor.read_ir,sensor.read_visible)
26     lecturas.append(lectura)
27     time.sleep(1)
28
29 print(lecturas)
30
31 urequests.post('http://www.lectoruv.com/api/v1.0/envio',json=lecturas, headers={'Content-Type': 'application/json'})
32
33 with open('data.json', 'w') as outfile:
34     outfile.write(str(lecturas))
```

Fuente: elaboración propia, empleando Pc local.

## **4. SERVIDOR WEB**

También llamado servidor HTTP, se le define como un programa informático que procesa una aplicación del lado del servidor, este programa informático debe ceder al cliente de forma bidireccional o unidireccional una conexión síncrona o asíncrona, cediendo al cliente una respuesta a su necesidad en cualquier tipo de lenguaje (Java, Python, Windows o Linux). El código recibido debe ir procesado por un navegador web. El protocolo que usa un servidor web para transmitir sus datos generalmente es el HTTP, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador.

Las aplicaciones escritas en varios lenguajes de programación que funcionan en plataformas diferentes pueden utilizar web services para intercambiar información a través de una red. La interoperabilidad, por ejemplo, entre Java y Python o Windows y Linux se debe al uso de estándares abiertos.

La base de comunicación entre web services es por tanto el lenguaje XML y el protocolo HTTP.

### **4.1. Modelo de un Servidor Web**

Los webs services estandarizados funcionan con los siguientes componentes:

#### **4.1.1. SOAP - Simple Object Access Protocol**

Es un protocolo escrito en XML para el intercambio de información entre aplicaciones. Es un formato para enviar mensajes, diseñado especialmente para servir de comunicación en Internet, pudiendo extender los HTTP headers. Es una forma de definir qué información se envía y cómo mediante XML. Básicamente es un protocolo para acceder a un Web Service.

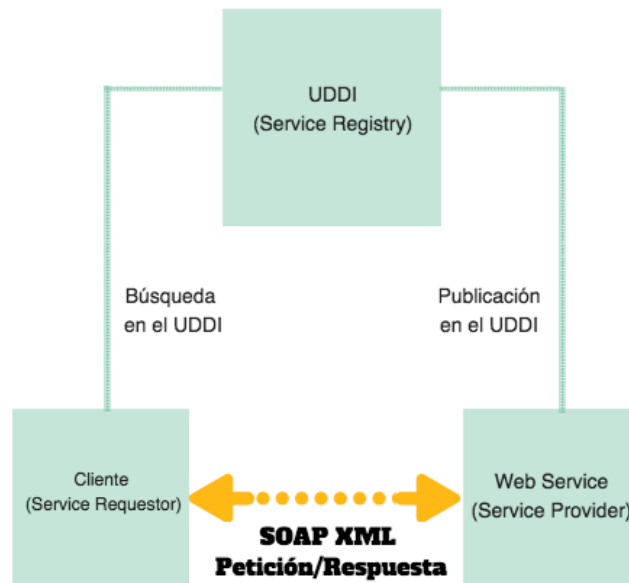
#### **4.1.2. WSDL - Web Services Description Language**

Es un lenguaje basado en XML para describir los servicios web y cómo acceder a ellos. Es el formato estándar para describir un web service, y fue diseñado por Microsoft e IBM. WSDL es una parte integral del estándar UDDI, y es el lenguaje que éste utiliza.

#### **4.1.3. UDDI - Universal Description, Discovery and Integration**

Es un estándar XML para describir, publicar y encontrar servicios web. Es un directorio donde las compañías pueden registrar y buscar servicios web. Es un directorio de interfaces de servicios web descritos en WSDL que se comunican mediante SOAP.

Figura 26. **Modelo de servicios web**



Fuente: Diego Lázaro. *Introducción a los Web Services*. <https://diego.com.es/introduccion-a-los-web-services>. Consulta: 13 de septiembre de 2019.

#### 4.2. **Arquitectura de un Servidor WEB**

- Service Discovery: centraliza servicios web en un directorio común de registro y proveer una funcionalidad sencilla para publicar y buscar. UDDI se encarga del Service Discovery.
- Service Description. Uno de los aspectos más característicos de los webs services es que se autodescriben. Esto significa que una vez que se ha localizado un Web Service proporcionará información sobre que operaciones soporta y cómo activarlo. Esto se realiza a través del Web Services Description Language (WSDL).
- Service Invocation. Invocar a un Web Service implica pasar mensajes entre el cliente y el servidor. SOAP (*Simple Object Access Protocol*)

especifica cómo se deberían formatear los mensajes request para el servidor, y cómo el servidor debería formatear sus mensajes de respuesta.

- Transport. Todos estos mensajes han de ser transmitidos de alguna forma entre el servidor y el cliente. El protocolo elegido para ello es HTTP (HyperText Transfer Protocol). Se pueden utilizar otros protocolos, pero HTTP es actualmente el más usado.

### **4.3. API**

La interfaz de programación de aplicaciones, conocida también por la sigla API, en inglés, application programming interface,<sup>1</sup> es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas, generalmente, en las bibliotecas de programación

Una API representa la capacidad de comunicación entre componentes de software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software.

### **4.4. REST**

En la actualidad, se usa en el sentido más amplio para describir cualquier interfaz entre sistemas que utilice directamente HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato (XML,

JSON, entre otros) sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes, como por ejemplo SOAP. Es posible diseñar sistemas de servicios web de acuerdo con el estilo arquitectural REST de Fielding y también es posible diseñar interfaces XMLHTTP de acuerdo con el estilo de llamada a procedimiento remoto (RPC), pero sin usar SOAP. Estos dos usos diferentes del término REST causan cierta confusión en las discusiones técnicas, aunque RPC no es un ejemplo de REST.

#### **4.4.1. Características de REST**

- Protocolo cliente/servidor sin estado: cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla. Aunque esto es así, algunas aplicaciones HTTP incorporan memoria caché. Se configura lo que se conoce como protocolo cliente-caché-servidor sin estado: existe la posibilidad de definir algunas respuestas a peticiones HTTP concretas como cacheables, con el objetivo de que el cliente pueda ejecutar en un futuro la misma respuesta para peticiones idénticas. De todas formas, que exista la posibilidad no significa que sea lo más recomendable.
- Las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro: POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).
- Los objetos en REST siempre se manipulan a partir de la URI. Es la URI y ningún otro elemento el identificador único de cada recurso de ese sistema REST. La URI nos facilita acceder a la información para su modificación o borrado, o, por ejemplo, para compartir su ubicación exacta con terceros.

- Interfaz uniforme: para la transferencia de datos en un sistema REST, este aplica acciones concretas (POST, GET, PUT y DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Esto facilita la existencia de una interfaz uniforme que sistematiza el proceso con la información.
- Sistema de capas: arquitectura jerárquica entre los componentes. Cada una de estas capas lleva a cabo una funcionalidad dentro del sistema REST.
- Uso de hipermedios: hipermedia es un término acuñado por Ted Nelson en 1965. Es una extensión del concepto de hipertexto. Ese concepto llevado al desarrollo de páginas web permite que el usuario puede navegar por el conjunto de objetos a través de enlaces HTML. En el caso de una API REST, el concepto de hipermedia explica la capacidad de una interfaz de desarrollo de aplicaciones de proporcionar al cliente y al usuario los enlaces adecuados para ejecutar acciones concretas sobre los datos.

#### **4.5. Web Frameworks**

Un *Web Framework* es una aplicación o paquete que facilita la programación web. Las aplicaciones del lado del servidor (por ejemplo: un catálogo, búsqueda y visualización de contenido, sitio de reserva u otros sitios web interactivos) usan un *framework*, para manejar los detalles de las solicitudes de red en la web (web network request), visualización de página (page display) y entrada / salida de base de datos, mientras libera al programador para proporcionar solo la lógica de cómo funcionará la aplicación.



#### 4.5.1. Full Stack Web Frameworks

Una aplicación web está formada por varias capas de componentes, que están configurados para funcionar juntos (es decir, integrados en una "pila"). Dichos componentes pueden incluir:

- Autenticación e identificación de usuarios a través de cookies.
- Manejo de entrada de datos desde formularios y URL.
- Lectura y escritura de datos hacia y desde almacenamiento persistente (por ejemplo, bases de datos).
- Visualización de plantillas con datos dinámicos insertados.
- Proporcionando estilo a plantillas (por ejemplo, css).
- Que proporciona funcionalidad de página web dinámica, como con AJAX.

La expresión full stack developer es utilizada por las escuelas y los reclutadores, se refiere a un desarrollador, que es competente en todas estas áreas.

#### 4.5.2. Django

Django es un *framework* de aplicaciones web gratuito y de código abierto (open source) escrito en Python. Como se mencionaba anteriormente, un *framework* web es un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente.

Cuando se construye un sitio en la web, se necesita un conjunto de componentes similares: autenticación de usuarios (registrarse, iniciar sesión, cerrar sesión), un panel de administración para el sitio web, formularios, una forma de subir archivos, entre otros.

Creado por desarrolladores experimentados, se ocupa de gran parte de la molestia del desarrollo web, por lo que puede concentrarse en escribir su aplicación sin necesidad de reinventar la rueda. Es gratis y de código abierto.

### **4.5.3. Web Frameworks "ligeros"**

Un framework "liviano" proporciona la funcionalidad básica necesaria para una aplicación del lado del servidor, pero permite al usuario agregar otros componentes de la pila como desee. Por lo general, es más fácil comenzar con estas aplicaciones porque requieren menos configuración e instalación. Flask es un marco liviano popular con muchos valores predeterminados convenientes que nos permite iniciar nuestra aplicación web rápidamente.

### **4.5.4. Flask**

Es un Micro framework escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. Se basa en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD.

- WSGI: Web Server Gateway Interface (WSGI) se ha adoptado como estándar para el desarrollo de aplicaciones web Python. WSGI es una especificación para una interfaz universal entre el servidor web y las aplicaciones web.
- Werkzeug: es un kit de herramientas WSGI que implementa solicitudes, objetos de respuesta y otras funciones de utilidad. Esto permite construir un marco web sobre él. El marco Flask usa Werkzeug como una de sus bases.

- jinja2: es un popular motor de plantillas para Python. Un sistema de plantillas web combina una plantilla con un determinado origen de datos para representar páginas web dinámicas.
- Instalación: para la instalación de flask, ingresar al *link* siguiente: <https://flask.palletsprojects.com/en/1.1.x/installation/#install-flask> en caso no le sea útil, debe buscar como instalar flask en S.O Windows de la temporada.
- Se presentará una dirección para crear las primeras REST APIs en FLASK.
- <https://blog.nearsoftjobs.com/crear-un-api-y-una-aplicaci%C3%B3n-web-con-flask-6a76b8bf5383>.
- Librerías con Flask.
  - SQLite 3 con Flask: En Flask puede implementar fácilmente la apertura de conexiones de bases de datos a pedido y cerrarlas cuando el contexto muere (usualmente cuando finaliza el request). <https://flask.palletsprojects.com/en/1.1.x/patterns/sqlite3/>.
  - Plotly con flask: plotly.py es una biblioteca gráfica interactiva basada en navegador para Python. Construido sobre plotly.js , plotly.py es una biblioteca de gráficos declarativa de alto nivel. plotly.js se envía con más de 30 tipos de gráficos, incluidos gráficos científicos, gráficos 3D, gráficos estadísticos, mapas SVG, gráficos financieros y más.
  - NumPy con flask: es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituye una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.
  - Traceback con flask: este módulo proporciona una interfaz estándar para extraer, formatear e imprimir rastros de pila de

programas Python. Imita exactamente el comportamiento del intérprete de Python cuando imprime un seguimiento de la pila. Esto es útil cuando desea imprimir trazas de pila bajo el control del programa, como en un "contenedor" alrededor del intérprete.}

#### 4.6. Diseño y creación del Backend

- Creación del Modelo de Datos con flask

En esta sección, se presentará el código que toma la información, del archivo generado por la capa física (SI1145) y la convierte en una base de datos estructurada.

Figura 27. **Modelo de datos con flask**

```
...
datamodel.py
proposito: crea modelo de datos para aplicacion de lectura de sensor uv
autor: Ariel Chitay

...
import sqlite3
import os

DB_NAME = 'lecturas.db'

def crear_db():
    '''Crea modelo de datos de aplicacion de sensor uv'''
    if os.path.exists(DB_NAME):
        os.remove(DB_NAME)

    conn = sqlite3.connect(DB_NAME)
    c = conn.cursor()
    # Create table
    c.execute('CREATE TABLE lecturas
              (uv real, ir real, visible real)')

    # Save (commit) the changes
    conn.commit()
    # We can also close the connection if we are done with it.
    # Just be sure any changes have been committed or they will be lost.
    conn.close()

if __name__ == '__main__':
    crear_db()
```

Fuente: elaboración propia, empleando PC local.

- REST API

En esta sección se creará la interfaz para poder conectar la ESP8266 a nuestro servidor web, para que se puedan almacenar sus datos y mostrarlos en una página web.

Figura 28. Creación de la interfaz

```
import flask
import sqlite3
import os
#Librerias para graficar y manejar matematica
import plotly.offline as py # es para trabajarlo independiende de plotly offline
import plotly.graph_objs as go
import numpy as np
import traceback

app = flask.Flask(__name__) # a Flask object

DB_NAME = 'lecturas.db'

QUERY_INSERT = """INSERT INTO lecturas (uv,ir,visible) VALUES ({} , {} , {})"""
QUERY_SELECT = """SELECT uv, ir, visible FROM lecturas """

def generar_grafica(x,y):
    """Funcion que genera la grafica en plotly"""
    trace0 = go.Scatter(
        x = x,
        y = y,
        # mode = 'Lines',
        name = 'uv'
    )

    data = [trace0]

    #Esto es para el Layout (presentacion)
    layout = go.Layout(
        title='Indice UV',
        yaxis=dict(title='Indice UV'),
        xaxis=dict(title='Numero de muestra'),
    )

    fig = go.Figure(data=data, layout=layout)

    return py.plot(fig, output_type='div')
```

Continuación de la figura 28.

```
@app.route('/')
def hello_html():
    return flask.render_template('index.html')    # found in templates/response.html

@app.route('/api/v1.0/envio', methods = ['POST'])
def envio():
    status = ''
    error = ''
    try:
        data = flask.request.get_json()
        print(data)
        conn = sqlite3.connect(DB_NAME)
        c = conn.cursor()
        for lectura in data:
            print(str(lectura))
            c.execute(QUERY_INSERT.format(lectura['uv'],
            lectura['ir'],
            lectura['visible']))
        conn.commit()
        status = 'ok'
    except Exception as e:
        status = 'fail'
        error = traceback.format_exc()
    finally:
        conn.close()
    return flask.jsonify({'status': status, 'error' : error})
```

Continuación de la figura 28.

```
@app.route('/consulta')
def consulta():
    conn = sqlite3.connect(DB_NAME)
    c = conn.cursor()
    c.execute(QUERY_SELECT)
    lecturas = []

    for row in c:
        print(row)
        lecturas.append({"uv":row[0],"ir":row[1],"visible":row[2]})

    y = np.array([lectura['uv'] for lectura in lecturas]) #numpy
    N = len(y)
    x = np.linspace(1, N, N)

    conn.close()
    return flask.render_template('consulta.html',
                                mediciones=lecturas,
                                grafica=flask.Markup(generar_grafica(x,y)),
                                )

if __name__ == '__main__':
    app.run()
# app.run(debug=True, host='0.0.0.0', port=5000) # app starts serving in
```

Fuente: elaboración propia, empleando PC local.





## **5. FRONTEND**

Es la parte de un sitio web que interactúa con los usuarios, por eso decimos que está del lado del cliente. Backend es la parte que se conecta con la base de datos y el servidor que utiliza dicho sitio web, por eso decimos que el backend corre del lado del servidor. Estos dos conceptos explican a grandes rasgos cómo funciona una página web.

### **5.1. Front-end Developer**

Es la persona encargada de programar el navegador de una web, es decir, es el encargado de traducir las definiciones de diseño y estilo visual realizadas en etapas previas a códigos HTML semántico. Es una de las piezas fundamentales en un proyecto digital.

Aunque normalmente a un Front-end Developer se le asocien principios de diseño y de estructura de páginas web, como buen programador que es debe ser consciente que su trabajo se realizará en el lado cliente, en muchos casos en el navegador. Esta persona ha de tener conocimientos y tener en cuenta la usabilidad, legibilidad de la web o app, el diseño y la estructura de páginas web, aplicaciones para procurar que el usuario pueda tener una buena experiencia de usuario final.

### **5.2. HTML**

HTML (Lenguaje Marcado de HyperTexto) define el significado y la estructura del contenido web.

"Hipertexto" se refiere a enlaces de páginas web entre sí. Los enlaces son un aspecto fundamental de la Web. Al cargar contenido en internet y vincularlo a páginas creadas por otras personas, se convierte en un participante activo en la World Wide Web.

Un elemento HTML se separa de otro texto en un documento mediante "etiquetas", pueden escribirse en mayúsculas, minúsculas o una mezcla. Por ejemplo, la etiqueta `<title>` puede ser escrito como `<Title>`, `<TITLE>` o en cualquier otra forma.

En la figura 26 el navegador muestra un sitio web y en la figura 27, cómo se ve en lenguaje HTML.

Figura 29. **Muestra del sitio web**

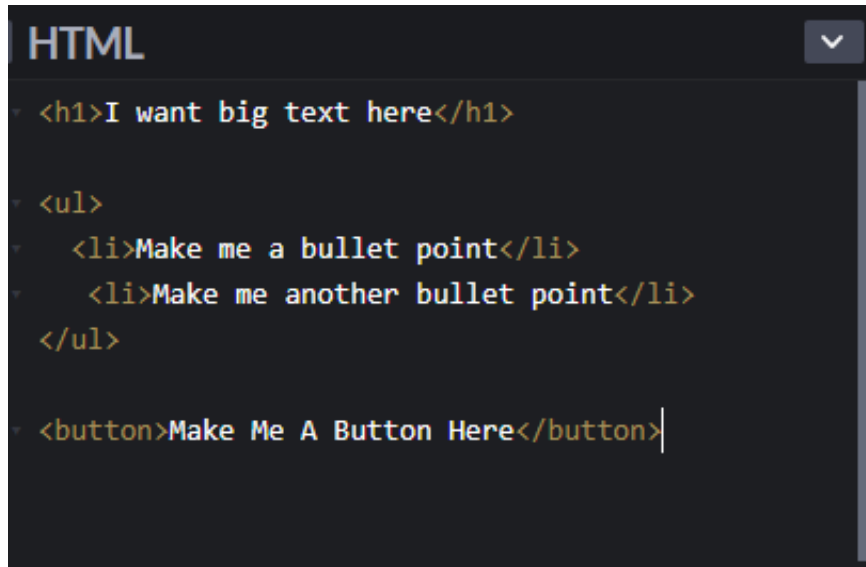
# I want big text here

- Make me a bullet point
- Make me another bullet point

**Make Me A Button Here**

Fuente: elaboración propia, empleando PC local.

Figura 30. Lenguaje HTML

A screenshot of a code editor window titled "HTML". The editor has a dark background and shows the following HTML code: 

```
<h1>I want big text here</h1>

<ul>
  <li>Make me a bullet point</li>
  <li>Make me another bullet point</li>
</ul>

<button>Make Me A Button Here</button>
```

Fuente: elaboración propia, empleando PC local.

### 5.3. CSS

CSS es un lenguaje para especificar cómo se presentan los documentos a los usuarios, cómo se diseñan, se presentan, entre otros. A diferencia de HTML que lo que está viendo son los estilos predeterminados del navegador: estilos muy básicos que el navegador aplica al HTML para asegurarse de que sean básicamente legibles incluso si el autor de la página no especifica un estilo explícito.

Un documento suele ser un archivo de texto estructurado con un lenguaje de marcado: HTML es el lenguaje de marcado más común, pero también puede encontrar otros lenguajes de marcado como SVG o XML .

CSS puede usarse para un estilo de texto de documento muy básico, por ejemplo, cambiar el color y el tamaño de los encabezados y enlaces. Se puede usar para crear un diseño, por ejemplo, convertir una sola columna de texto en un diseño con un área de contenido principal y una barra lateral para información relacionada. Incluso se puede usar para efectos como la animación.

### 5.3.1. Sintaxis CSS

CSS es un lenguaje basado en reglas: usted define reglas que especifican grupos de estilos que deben aplicarse a elementos particulares o grupos de elementos en su página web. Por ejemplo, "Quiero que el encabezado principal de mi página se muestre como texto rojo grande".

El siguiente código lograría el estilo que se mostró en la sección de HTML.

Figura 31. Estilo del código CSS

```
CSS
h1 {
  color: purple;
}

button {
  border: 2px solid #34495e;
  background-color: #3498db;
  color: white;
}
```

Fuente: elaboración propia, empleando PC local.

Figura 32. **Muestra de sitio Web**

**I want big text here**

- Make me a bullet point
- Make me another bullet point

Make Me A Button Here

Fuente: elaboración propia, empleando PC local.

La regla en CSS se abre con un selector. Esto selecciona el elemento HTML que vamos a diseñar. En este caso, estamos diseñando encabezados de nivel uno ( <h1>).

Luego se tiene un conjunto de llaves { }. Dentro de ellas habrá una o más declaraciones que tomarán la forma de pares de propiedades y valores. Cada par especifica una propiedad de los elementos que estamos seleccionando, luego un valor que nos gustaría dar a la propiedad.

Antes de los dos puntos, se tiene la propiedad, y después de los dos puntos, el valor. Las propiedades CSS tienen diferentes valores permitidos, dependiendo de qué propiedad se esté especificando. En el ejemplo, la propiedad está en color, que puede tomar varios valores de color. También está la propiedad font-size. Esta propiedad puede tomar varias unidades de tamaño como valor.

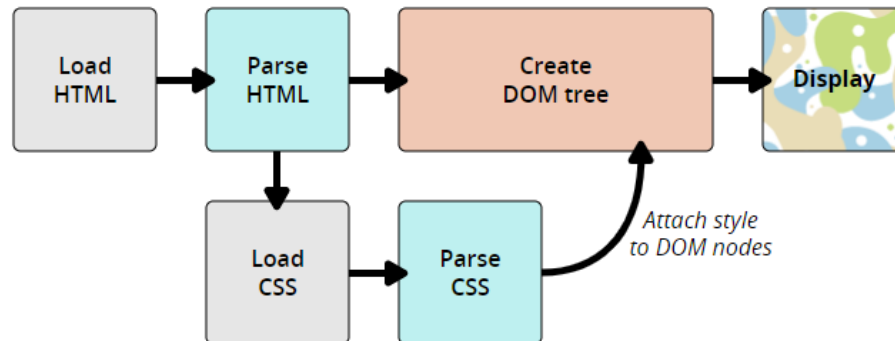
Una hoja de estilo CSS contendrá muchas de esas reglas, escritas una tras otra.

### 5.3.2. ¿Cómo funciona realmente CSS?

Cuando un navegador muestra un documento, debe combinar el contenido del documento con su información de estilo. Procesa el documento en varias etapas, que se muestran a continuación. Esta es una versión muy simplificada de lo que sucede cuando un navegador carga una página web, y que diferentes navegadores manejarán el proceso de diferentes maneras. Pero esto es más o menos lo que sucede.

- El navegador carga el HTML (por ejemplo, lo recibe de la red).
- Convierte el HTML en un DOM (Modelo de Objeto de Documento). El DOM representa el documento en la memoria de la computadora.
- El navegador luego recupera la mayoría de los recursos que están vinculados por el documento HTML, como imágenes y videos incrustados y CSS vinculado.
- El navegador analiza el CSS obtenido y clasifica las diferentes reglas por sus tipos de selector en diferentes "cubos", por ejemplo, elemento, clase, ID, etc. En función de los selectores que encuentra, determina qué reglas se deben aplicar a qué nodos en el DOM y les asigna estilo según sea necesario (este paso intermedio se denomina árbol de renderizado).
- El árbol de renderizado se presenta en la estructura en la que debería aparecer después de que se le hayan aplicado las reglas.
- La visualización de la página se muestra en la pantalla (esta etapa se llama pintura).

Figura 33. **Función del CSS**



Fuente: W3C. *Cascading Style Sheets*. <https://www.w3.org/Style/CSS/#specs>. Consulta: 23 de marzo de 2020.

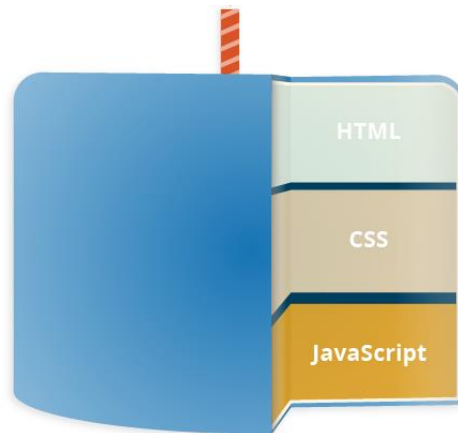
#### 5.4. **JavaScript**

JavaScript es un lenguaje interpretado, orientado a objetos, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, como node.js, Apache CouchDB y Adobe Acrobat.

El estándar de JavaScript es ECMAScript. Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1.

Cada vez más, una página web desarrolla más acciones, además de mostrar información estática como mostrar actualizaciones de contenido en el momento, interactuar con mapas, animaciones gráficas 2D/3D. Se presenta el Modelo de Javascript en la figura 34.

Figura 34. **Modelo de JavaScript**



Fuente: W3C. *Cascading Style Sheets*. <https://www.w3.org/Style/CSS/#specs>. Consulta: 23 de marzo de 2020.

## 5.5. **Bootstrap**

Bootstrap es un marco gratuito de código abierto HTML , CSS y JavaScript para construir rápidamente sitios web receptivos.

Inicialmente, Bootstrap se llamaba Twitter Blueprint, se desarrolló en Twitter. Admite un diseño receptivo, presenta plantillas de diseño predefinidas que puede usar de inmediato o personalizar para sus necesidades con su propio código. Tampoco debe preocuparse por la compatibilidad con otros navegadores, ya que Bootstrap es compatible con todos los navegadores modernos y las versiones más recientes de Internet Explorer.



## 5.6. Componentes Bootstrap

Los componentes más destacados de Bootstrap son sus componentes de diseño, ya que afectan a toda una página web. El componente de diseño básico se llama "Contenedor", ya que todos los demás elementos de la página se colocan en él. Los desarrolladores pueden elegir entre un contenedor de ancho fijo y un contenedor de ancho fluido. Mientras que el último siempre llena el ancho de la página web, el primero usa uno de los cuatro anchos fijos predefinidos, dependiendo del tamaño de la pantalla que muestra la página:

- Menor que 576 píxeles
- 576–768 píxeles
- 768–992 píxeles
- 992–1 200 píxeles
- Más grande que 1 200 píxeles

Una vez que un contenedor está en su lugar, otros componentes de diseño de Bootstrap implementan un diseño de cuadrícula CSS a través de la definición de filas y columnas.

Una versión precompilada de Bootstrap está disponible en forma de un archivo CSS y tres archivos JavaScript que se pueden agregar fácilmente a cualquier proyecto. Sin embargo, la forma en bruto de Bootstrap permite a los desarrolladores implementar más personalizaciones y optimizaciones de tamaño. Esta forma cruda es modular, lo que significa que el desarrollador puede eliminar componentes innecesarios, aplicar un tema y modificar los archivos Sass sin compilar.

## 5.7. Plantillas Bootstrap

Existen diversidades de páginas web con plantillas (gratuitas) escritas en Bootstrap. A continuación, se dará una fracción de ejemplo de un documento con código y su respectivo fragmento de resultado.

Tabla V. Ejemplo plantillas Bootstrap

```
<!DOCTYPE html>
<html lang="en">

  <title>SB Admin 2 - Charts</title>

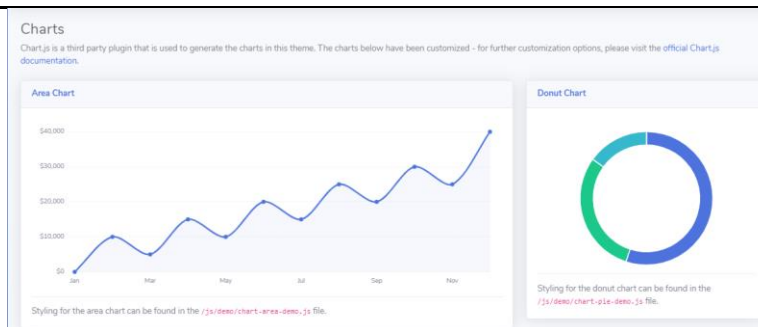
  <!-- Custom fonts for this template-->
  <link href="vendor/fontawesome-free/css/all.min.css" rel="stylesheet"
type="text/css">
  <link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,4
00i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">

  <!-- Custom styles for this template-->
  <link href="css/sb-admin-2.min.css" rel="stylesheet">

</head>

<body id="page-top">

  <!-- Page Wrapper -->
  <div id="wrapper">
```



Fuente: Star Bootstrap. *SB Admin 2*. <https://bit.ly/2ZJNeMK>. Consulta: 25 de marzo de 2020.

## **5.8. Jinja**

Es un motor de plantillas web (web template engine) para el lenguaje de programación Python. Está licenciado como software libre (Licencia BSD) creada por Armin Ronacher . Es similar al motor de plantillas de Django , pero proporciona expresiones similares a Python al tiempo que garantiza que las plantillas se evalúen en un entorno limitado . Es un lenguaje de plantilla basado en texto, por lo tanto, se puede utilizar para generar cualquier marcado, así como el código fuente.

El motor de plantillas Jinja permite la personalización de etiquetas, filtros, pruebas y globales. Además, a diferencia del motor de plantillas Django, Jinja permite que el diseñador de plantillas invoque funciones con argumentos sobre objetos. Jinja es el motor de plantillas predeterminado de Flask y también lo usan Ansible<sup>1</sup> y Trac .

Si quiere saber más acerca de jinja, acceda a su documentación a través de este link: <https://jinja.palletsprojects.com/en/2.10.x/> o a una explicación básica a través de este link: <https://pythonista.io/cursos/py201/introduccion-a-jinja-2>.

## **5.9. Frontend del proyecto**

A continuación, se presentará el código que presenta la visualización de la conexión entre la capa física, la capa de red y la de negocio.

### 5.9.1. Plantilla base del frontend

En esta plantilla se programó el código base, se trabajará con código HTML, CSS, Bootstrap, Flask y Jinja2. Este código servirá como plantilla base para las demás secciones de código del Frontend.

Figura 35. Plantilla base del frontend

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="icon" href="../../../../favicon.ico">

    <title>Lector de Indice UV</title>

    <!-- Bootstrap core CSS -->
    <link href="../../static/bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom styles for this template -->
    <link href="dashboard.css" rel="stylesheet">
  </head>

  <body>
    <nav class="navbar navbar-dark sticky-top bg-dark flex-md-nowrap p-0">
      <a class="navbar-brand col-sm-3 col-md-2 mr-0" href="#">Lector UV</a>
      <input class="form-control form-control-dark w-100" type="text" placeholder="Search" aria-label="Search">
      <ul class="navbar-nav px-3">
        <li class="nav-item text-nowrap">
          <a class="nav-link" href="#">Sign out</a>
        </li>
      </ul>
    </nav>
```

Continuación de la figura 35.

```
<div class="container-fluid">
  <div class="row">
    <nav class="col-md-2 d-none d-md-block bg-light sidebar">
      <div class="sidebar-sticky">
        <ul class="nav flex-column">
          <li class="nav-item">
            <a class="nav-link active" href="/">
              <span data-feather="home"></span>
              Inicio <span class="sr-only">(current)</span>
            </a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/consulta">
              <span data-feather="bar-chart-2"></span>
              Reportes
            </a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">
              <span data-feather="file"></span>
              Recomendacion
            </a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">
              <span data-feather="shopping-cart"></span>
              Productos
            </a>
          </li>
        </ul>
      </div>
    </nav>
  </div>
</div>
```

Continuación de la figura 35.

```

<li class="nav-item">
  <a class="nav-link" href="#">
    <span data-feather="users"></span>
    Usuarios
  </a>
</li>
</ul>

<h6 class="sidebar-heading d-flex justify-content-between align-items-center px-3 mt-4 mb-1 text-muted">
  <span>Reportes Guardados</span>
  <a class="d-flex align-items-center text-muted" href="#">
    <span data-feather="plus-circle"></span>
  </a>
</h6>
<ul class="nav flex-column mb-2">
  <li class="nav-item">
    <a class="nav-link" href="#">
      <span data-feather="file-text"></span>
      Current month
    </a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">
      <span data-feather="file-text"></span>
      Last quarter
    </a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">
      <span data-feather="file-text"></span>
      Last quarter
    </a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">
      <span data-feather="file-text"></span>
      Social engagement
    </a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">
      <span data-feather="file-text"></span>
      Year-end sale
    </a>
  </li>
</ul>
</div>
</nav>

<main role="main" class="col-md-9 ml-sm-auto col-lg-10 pt-3 px-4">
  <div class="d-flex justify-content-between flex-wrap flex-md-nowrap align-items-center pb-2 mb-3 border-bottom">
    <h1 class="h2">Lector UV</h1>
    <div class="btn-toolbar mb-2 mb-md-0">
      <div class="btn-group mr-2">
        <button class="btn btn-sm btn-outline-secondary">Share</button>
        <button class="btn btn-sm btn-outline-secondary">Export</button>
      </div>
      <button class="btn btn-sm btn-outline-secondary dropdown-toggle">
        <span data-feather="calendar"></span>
        This week
      </button>
    </div>
  </div>
</div>

```

Continuación de la figura 35.

```
{% block content %}
{% endblock %}

</main>
</div>
</div>

<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DRtIkvYIK3UENzmM7FCkRr/rE9/Qpg6AAGGjWFPVNVN/AOPqFF93h7P6S5KKN"
crossorigin="anonymous"></script>
<script>window.jQuery || document.write('<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DRtIkvYIK3UENzmM7FCkRr/rE9/Qpg6AAGGjWFPVNVN/AOPqFF93h7P6S5KKN"
crossorigin="anonymous"></script>')</script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR63jkx2y39wvcPRoID1nY8jwDR1qz0/6GkZo7l" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-C6300/3e37c3de644b901305724e69a743229797f3e0d46711a994fbc58d18" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js" integrity="sha384-JZR63jkx2y39wvcPRoID1nY8jwDR1qz0/6GkZo7l" crossorigin="anonymous"></script>

<!-- Icons -->
<script src="https://unpkg.com/feather-icons/dist/feather.min.js"></script>
<script>
  feather.replace()
</script>

<!-- Graphs -->
<script src="https://cdn.jsdelivr.net/npm/chart.js@2.7.1/dist/chart.min.js"></script>
<script>
  var ctx = document.getElementById("myChart");
  var myChart = new Chart(ctx, {
    type: 'line',
    data: {
      labels: ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"],
      datasets: [{
        data: [15339, 21345, 18483, 24003, 23489, 24092, 12034],
        lineTension: 0,
        backgroundColor: 'transparent',
        borderColor: '#007bff',
        borderWidth: 4,
        pointBackgroundColor: '#007bff'
      }]
    },
    options: {
      scales: {
        yAxes: [{
          ticks: {
            beginAtZero: false
          }
        }]
      },
      legend: {
        display: false,
      }
    }
  });
</script>
</body>
</html>
```

Fuente: elaboración propia, empleando <https://gitlab.com/ariel91/lectoruvapp/-/blob/master/templates/base.html>.

## 5.9.2. Sección de bienvenida

En esta sección se explican las estadísticas y las razones por las cuales se ha desarrollado este proyecto. Se utiliza Jinja2 para extender la bienvenida, con el documento mostrado anteriormente.

Figura 36. Sección de bienvenida

```
1  {% extends "base.html" %}
2
3  {% block content %}
4
5  <div class="container">
6
7      <!-- content goes in the body -->
8          <div class="jumbotron">
9              <div class="container">
10                 <h1>Dispositivo que mide indice UV </h1>
11                 <a class="btn btn-primary btn-lg" href="/consulta" role="button">Reportes
12                 &raquo;</a>
13                 </div>
14             </div>
15             <h2>Introduccion</h2>
16
17             <p class="text-justify" >Aproximadamente 91,270 nuevos casos de cáncer de
18             piel tipo melanoma estarán siendo detectados en el 2018, conformando solo
19             el 1% de los casos de cáncer de piel (American Cancer Society, January 4,
20             2018), sin embargo, el melanoma es la causa de la mayoría de las muertes en
21             el mundo. Tanto el Carcinoma como el Melanoma lo causan los rayos tipo
22             ultravioleta que el sol genera, las camas de bronceado o todo material que
23             refleje la luz del sol.</p>
24
25             <p class="text-justify">Para poder tener una medición de los rayos
26             ultravioleta la Organización Mundial de la Salud ha creado una escala de
27             indice UV para saber informar a las personas la escala a la que se está
28             exponiendo y así se pueda prevenir.</p>
29
30             <p class="text-justify">Según American Cancer Society (2018) dependiendo la
31             etapa donde se descubra el cáncer de piel así es la tasa de supervivencia,
32             por lo cual todas las sociedades necesitan información que les haga saber
33             cómo prevenirse lo antes posible den cáncer de piel melanoma o carcinoma.</p>
34
35             <p class="text-justify">
36             Sabiendo esto se ha propuesto una pulsera que puedan portar las personas
37             para que puedan estar informados sobre el indice ultra violeta al cual
38             están siendo expuestos y puedan tomar las medidas necesarias para no
39             llegar a formar parte de la población con cáncer de piel en el mundo.
40             Siendo el Internet de las Cosas la tecnología que se utilizara para
41             solventar este problema y ahorrar dinero a las personas que no tienen lo
42             suficiente para ir a un dermatólogo.
43
44             </p>
45         </div>
46     </div>
47
48 {% endblock %}
```

Fuente: elaboración propia, empleando <https://gitlab.com/ariel91/lectoruvapp/-/tree/master>.



### 5.9.3. Sección del Dashboard

En esta sección se presentarán las visualizaciones generadas por el sensor ultravioleta SI1145, los datos adquiridos y procesados del Backend y la vista que generarán los documentos que crearán el frontend.

Figura 37. Sección del Dashboard

```
1  {% extends "base.html" %}
2
3  {% block content %}
4      {{grafica}}
5  <div class="container">
6
7      <!-- content goes in the body -->
8      <h1>Medicion de Rayos Ultra Violeta</h1>
9      <div class="table-responsive">
10         <table class="table table-striped table-sm">
11             <thead class="thead-dark">
12                 <tr>
13                     <th>UV</th>
14                     <th>IR</th>
15                     <th>Visible</th>
16                 </tr>
17             </thead>
18             {% for medicion in mediciones %}
19                 <tr>
20                     <td>{{medicion.uv}}</td>
21                     <td>{{medicion.ir}}</td>
22                     <td>{{medicion.visible}}</td>
23                 </tr>
24             {% endfor %}
25         </table>
26     </div>
27 </div>
28 </div>
29 {% endblock %}
30
```

Fuente: elaboración propia, empleando <https://gitlab.com/ariel91/lectoruvapp/-/tree/master>.



## CONCLUSIONES

1. La intensidad del índice de radiación UV puede variar, según la altura del sol, reflexión por el suelo, latitud, altitud, ozono y nubosidad que exista en el ambiente.
2. El Open Source y el Open Hardware son las tecnologías adecuadas para una sociedad que necesita ser informada, son de acceso gratuito y eleva la calidad de vida del ciudadano.
3. La capa de percepción debe ser única y exclusivamente para adquisición de datos, la capa de transporte se debe utilizar en su API REST el protocolo HTTP, las visualizaciones en la capa de aplicación deben mostrar datos de manera limpia, formateada y fácil de entender.
4. Los mensajes preventivos al IUV deben ser sencillos y fáciles de comprender a través de una interfaz web, para que el receptor pueda ponerlos en práctica y adoptar hábitos saludables.



## RECOMENDACIONES

1. Todas las fuentes de información preventivas deben ser extraídas de INCAN Guatemala, American Cancer Society, Organización Mundial de la Salud enfocado al cáncer de piel. Este documento es netamente informativo, acuda a su médico si tiene problemas en su piel.
2. Los costos del proyecto se pueden reducir en gran manera, si se usa Python 3 en todas sus áreas de desarrollo y sistema operativo Linux.
3. Tener conocimiento básico de HTML, CSS, Flask, Jinja2, JavaScript, Micropython, SQL, Python 3, AWS (Opcional), Heroku, Microcontroladores.
4. Los mensajes o recomendaciones que contiene esta tesis son netamente preventivos, proporcionados por la OMS y recomendados por un médico salubrista. No se pretende sustituir a ningún profesional en el área, la persona que presente problemas en su piel, debe consultar con un médico.
5. Tomar en cuenta que las pruebas en un servidor local conllevan a un cambio en el código de programación, al momento de montar el servicio en la nube. No se debe usar computación en la niebla (Fog Computing) ya que el chip propuesto no soportaría el procesamiento. La seguridad de la REST API creada dependerá del tipo de dato que quiera transferir a la nube.



## BIBLIOGRAFÍA

1. BBVA API\_Market. *API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos.* [en línea]. <<https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>>. [Consulta: 20 de noviembre del 2019].
2. Buena vida. *Campaña busca prevenir y diagnosticar de manera oportuna el cáncer de piel.* [en línea]. <<https://www.prensalibre.com/guatemala/comunitario/buscan-prevenir-y-diagnosticar-de-manera-oportuna-el-cancer-de-piel/>>. [Consulta: 23 de noviembre del 2019].
3. CELIS, Ricardo. *¿Qué es frontend y backend?.* [en línea]. <<https://platzi.com/blog/que-es-frontend-y-backend/>>. [Consulta: 25 de noviembre del 2019].
4. Diego Lázaro. *Introducción a los webs services.* [en línea]. <<https://diego.com.es/introduccion-a-los-web-services>>. [Consulta: 25 de noviembre del 2019].
5. GARCÍA, Ylenia. *Qué es un Front end Developer.* [en línea]. <<https://www.iebschool.com/blog/funciones-front-end-developer-analitica-usabilidad/>>. [Consulta: 25 de noviembre del 2019].

6. IBM. *Web Services Conceptual Architecture (WSCA 1.0)*. [en línea]. <<https://www.csd.uoc.gr/~hy565/docs/pdfs/papers/wsca.pdf>>. [Consulta: 26 de noviembre del 2019].
7. Informe de los casos de cáncer registrados en el instituto de Cancerología y hospital. “Dr. Bernardo del Valle S.” INCAN durante el año 2011, por el registro de Cáncer de INCAN – Guatemala.
8. MANTILLA PERDOMO, Ana Estefanía. *Implementación de un prototipo de cobro y recarga prepago de tarifas para el servicio de transporte público urbano mediante una hmi controlada por nodeMCU*. [en línea]. <[https://i2.wp.com/www.esploradores.com/wp-content/uploads/2016/08/PINOUT-NodeMCU\\_1.0-V2-2.jpg](https://i2.wp.com/www.esploradores.com/wp-content/uploads/2016/08/PINOUT-NodeMCU_1.0-V2-2.jpg)>. [Consulta: 26 de noviembre del 2019].
9. MDN web docs. *HTML: Hypertext Markup Language*. [en línea]. <<https://developer.mozilla.org/en-US/docs/Web/HTML>>. [Consulta: 26 de noviembre del 2019].
10. Juan Gabriel Gomilla. *What is CSS*. [en línea]. <[https://developer.mozilla.org/enUS/docs/Learn/CSS/First\\_steps/What\\_is\\_CSS](https://developer.mozilla.org/enUS/docs/Learn/CSS/First_steps/What_is_CSS)>. [Consulta: 26 de noviembre del 2019].



## APÉNDICE

### Apéndice 1. Instaladores e información para usuarios LINUX

Instalación de Micropython en sistema operativo Linux	<a href="http://docs.micropython.org/en/latest/esp8266/tutorial/intro.html#intro">http://docs.micropython.org/en/latest/esp8266/tutorial/intro.html#intro</a>
Adquisición del Firmware de Micropython	<a href="http://micropython.org/download#esp8266">http://micropython.org/download#esp8266</a>
Activación del REPL desde Linux	<a href="http://docs.micropython.org/en/latest/esp8266/tutorial/repl.html">http://docs.micropython.org/en/latest/esp8266/tutorial/repl.html</a>
Activación del WebREPL desde Linux	<a href="http://docs.micropython.org/en/latest/esp8266/tutorial/repl.html">http://docs.micropython.org/en/latest/esp8266/tutorial/repl.html</a>
Uso del WebREPL	<a href="http://micropython.org/webrepl/">http://micropython.org/webrepl/</a>
Instalación de Python 3 en Linux	<a href="https://www.python.org/downloads/source/">https://www.python.org/downloads/source/</a>
Instalación de Flask en Linux	<a href="https://flask.palletsprojects.com/en/1.1.x/installation/#install-flask">https://flask.palletsprojects.com/en/1.1.x/installation/#install-flask</a>
Como usar Heroku y Python 3	<a href="https://devcenter.heroku.com/articles/getting-started-with-python#set-up">https://devcenter.heroku.com/articles/getting-started-with-python#set-up</a>
Uso de SQL3 y Python	<a href="https://docs.python.org/2/library/sqlite3.html">https://docs.python.org/2/library/sqlite3.html</a>
Plantillas de Bootstrap gratis	<a href="https://getbootstrap.com/docs/4.4/examples/">https://getbootstrap.com/docs/4.4/examples/</a>
Instalación de Jinja2 y documentación	<a href="https://jinja.palletsprojects.com/en/2.11.x/intro/#installation">https://jinja.palletsprojects.com/en/2.11.x/intro/#installation</a>

Fuente: elaboración propia.

