

## ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



### TRABAJO FIN DE GRADO

**Interfaz VLC-Sigfox o VLC-LoRa con plataforma de  
gestión de sensores.**

**Titulación: Grado en Ingeniería en Tecnologías de la  
Telecomunicación**

**Mención: Sistemas de Telecomunicación**

**Autor: Ángel Caamaño Lourido**

**Tutores: D. José Alberto Rabadán Borges  
D. Julio Francisco Rufo Torres**

**Fecha: 15 de enero de 2019**



## ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



### TRABAJO FIN DE GRADO

**Interfaz VLC-Sigfox o VLC-LoRa con plataforma  
de gestión de sensores.**

### HOJA DE FIRMAS

**Alumno/a**

Fdo.: Ángel Caamaño Lourido

**Tutor/a**

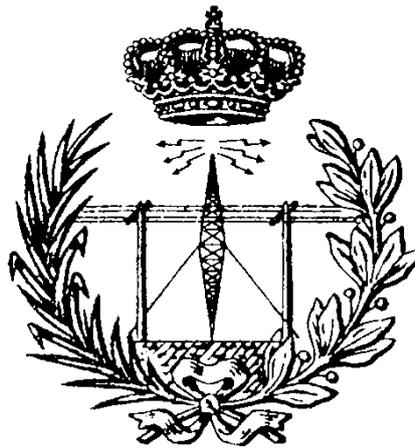
**Tutor/a**

Fdo.: D. José Alberto Rabadán Borges

Fdo.: D. Julio Francisco Rufo Torres

**Fecha: 15 de enero del 2019**

# ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



## TRABAJO FIN DE GRADO

**Interfaz VLC-Sigfox o VLC-LoRa con plataforma de  
gestión de sensores.**

## HOJA DE EVALUACIÓN

Calificación: \_\_\_\_\_

**Presidente**

Fdo.:

**Vocal**

Fdo.:

**Secretario/a**

Fdo.:

**Fecha:**

**Enero 2019**



*Mientras estemos entre los humanos permítannos ser humanos.*

*Séneca.*



# Agradecimientos

*A mis tutores por todo el esfuerzo y ayuda prestada que han hecho posible este proyecto.*

*Gracias a mis hermanos por estar ahí dispuestos siempre a ayudar y a aconsejar.*

*Y, por último, a mis padres por su apoyo incondicional y sus fuerzas inquebrantables.*

*A todos ellos, gracias.*



# Índice General

I.	Memoria.....	15
<b>1.</b>	<b>Introducción.....</b>	<b>16</b>
1.1.	Antecedentes. ....	16
1.2.	Peticionario. ....	17
1.3.	Objetivos del trabajo. ....	17
1.4.	Estructura de la memoria .....	18
<b>2.</b>	<b>Capítulo II. Tecnología VLC. Comunicaciones por luz visible.....</b>	<b>20</b>
2.1.	Estandarización de VLC .....	22
2.2.	Configuraciones .....	22
2.3.	Modulaciones.....	23
2.4.	Características .....	27
2.5.	Ventajas y desventajas .....	28
2.6.	Comparativa con RF .....	29
2.7.	Aplicaciones.....	30
<b>3.</b>	<b>Capítulo III. Tecnología LoRa. Long-Range.....</b>	<b>33</b>
3.1.	LoRaWAN .....	34
3.2.	Características .....	34
3.3.	Ventajas y desventajas .....	34
3.4.	Comparativa con SigFox.....	35
3.5.	Aplicaciones.....	35
<b>4.</b>	<b>Capítulo IV. Hardware y software utilizado. ....</b>	<b>37</b>
4.1.	Hardware empleado .....	37
4.2.	Software utilizado .....	44
<b>5.</b>	<b>Capítulo V. Implementación del sistema.....</b>	<b>53</b>
5.1.	Enlace VLC.....	53
5.2.	Enlace LoRa.....	57
5.3.	Aplicación web .....	63
5.4.	Implementación final .....	70
<b>6.</b>	<b>Capítulo VI. Conclusiones.....</b>	<b>73</b>
<b>7.</b>	<b>Capítulo VII. Bibliografía. ....</b>	<b>75</b>
II.	Pliego de condiciones .....	78
III.	Presupuesto .....	82
	Anexos .....	91
1.	Instalación de Node.JS.....	92
2.	Instalación de MongoDB .....	94

3. Instalación de Angular CLI.....	97
4. Instalación de Python.....	97
5. Instalación de Arduino IDE.....	99
6. Instalación de Visual Studio Code.....	101
7. Instalación de Postman.....	102
8. Contenido del CD-ROM.....	104

# Índice de figuras

Figura 1. Localización del espectro visible. ....	21
Figura 2. Enlace básico de telecomunicación. ....	22
Figura 3. Representación del LED blanco basado en fósforo. ....	22
Figura 4. Elementos básicos de un enlace VLC. ....	23
Figura 5. (a) configuración LOS; (b) NLOS; (c) enlace difuso. ....	24
Figura 6. Esquema de modulaciones. ....	25
Figura 7. Modulación On-Off Keying. ....	26
Figura 8. Ejemplo de PPM. ....	26
Figura 9. Gamut de un LED RGB. ....	28
Figura 10. Esquema de LoRa. ....	34
Figura 11. Ordenador portátil. ....	38
Figura 12. Arduino Uno R3. ....	39
Figura 13. Esquema del Arduino Uno R3. ....	40
Figura 14. Arduino Mega 2560 R3. ....	41
Figura 15. Esquema del Arduino Mega 2560 R3. ....	42
Figura 16. Lámpara VLC. Transmisor. ....	42
Figura 17. Receptor VLC. ....	43
Figura 18. Esquema del Receptor VLC. ....	43
Figura 19. Dispositivo GAMMA-868. ....	44
Figura 20. Esquema del GAMMA-868. ....	45
Figura 21. Logo de MongoDB. ....	46
Figura 22. Algunas características de MongoDB. ....	47
Figura 23. Logo de Node.js. ....	48
Figura 24. Logo de Express.JS. ....	49
Figura 25. Logo de Angular.JS. ....	49
Figura 26. Logo de Python. ....	51
Figura 27. Ventana de Arduino IDE. ....	52
Figura 28. Logo de Visual Studio Code. ....	53
Figura 29. Ventana de Visual Studio Code. ....	53
Figura 30. Logo de Postman. ....	54
Figura 31. Ventana de Postman. ....	54
Figura 32. Esquema del sistema. ....	55
Figura 33. Enlace VLC en funcionamiento. ....	56
Figura 34. Configuración del Arduino Mega para receptor VLC. Variables y setup(). ....	57
Figura 35. Configuración del Arduino Mega para receptor VLC. Loop(). ....	58
Figura 36. Esquemático de la lámpara VLC. ....	58
Figura 37. Esquemático del receptor VLC. ....	59
Figura 38. Configuración del módulo LoRa. Transmisor (Arduino Mega). ....	60
Figura 39. Arduino Mega. Transmisor LoRa. ....	61
Figura 40. Código de Arduino Mega. Envío de datos por LoRa. ....	62
Figura 41. Configuración del módulo LoRa. Receptor (Arduino Uno). ....	63
Figura 42. Arduino Uno. Receptor LoRa. ....	64
Figura 43. Código de Arduino Uno. Recepción de datos por LoRa. ....	65
Figura 44. Esquema de la aplicación web. ....	65
Figura 45. Esquema de "emisor". ....	66
Figura 46. Esquema de "emisorRand". ....	66
Figura 47. Código de emisor.py. ....	67
Figura 48. Código de emisorRand.py. ....	68
Figura 49. Diagrama UML. ....	69

Figura 50. Esquema de la API.....	70
Figura 51. Esquema del Front-End.....	71
Figura 52. Recepción de los datos por el puerto serie. ....	72
Figura 53. Datos de la base de datos. ....	73
Figura 54. Página web. ....	74
Figura 55. Los datos mostrados.....	75
Figura 56. Descarga de Node.JS.....	94
Figura 57. Opciones en la instalación (I).....	95
Figura 58. Opciones en la instalación (II). ....	96
Figura 59. Descarga de MongoDB.....	97
Figura 60. Instalación de MongoDB. ....	98
Figura 61. Términos y condiciones de MongoDB. ....	99
Figura 62. Descarga de Python.....	100
Figura 63. Instalación de Python. ....	101
Figura 64. Descarga de Arduino IDE. ....	102
Figura 65. Términos y condiciones de Arduino IDE. ....	102
Figura 66. Instalación de Visual Studio Code.....	104
Figura 67. Descarga de Postman. ....	105
Figura 68. Ventana inicial de Postman.....	105

## Índice de tablas

Tabla 1. Ventajas y desventajas.....	30
Tabla 2. Equipo hardware.....	82
Tabla 3. Equipo software.....	82
Tabla 4. Coeficientes reductores para trabajo tarifado según el COITT. ....	86
Tabla 5. Precios y costes de amortización hardware. ....	88
Tabla 6. Precios y costes de amortización del software. ....	88
Tabla 7. Precios y costes de la ejecución del trabajo más la amortización. ....	89
Tabla 8. Coste final de redacción del trabajo. ....	90
Tabla 9. Coeficientes para el cálculo del visado. ....	90
Tabla 10. Cálculo total P para el cálculo del visado (Presupuesto base).....	90
Tabla 11. Presupuesto total sin impuestos.....	91
Tabla 12. Presupuesto total. ....	91

## Acrónimos

<b>VLC</b>	<i>Visible Light Communications</i>
<b>IoT</b>	<i>Internet of Things</i>
<b>LED</b>	<i>Light-Emitting Diode</i>
<b>TFG</b>	<i>Trabajo Fin de Grado</i>
<b>ULPGC</b>	<i>Universidad Las Palmas de Gran Canaria</i>
<b>EITE</b>	<i>Escuela de Ingeniería de Telecomunicación y Electrónica</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>M.E.A.N.</b>	<i>Mongo, Express, Angular, Node.js</i>
<b>LASER</b>	<i>Light Amplification by Stimulated Emission of Radiation</i>
<b>IEEE</b>	<i>Institute of Electrical and Electronics Engineers</i>
<b>OOK</b>	<i>On-off keying</i>
<b>VPPM</b>	<i>Variable Pulse Position Modulation</i>

# **Parte I**

## **Memoria**

# 1. Introducción

## 1.1. Antecedentes

Durante la última década, el consumo de datos ha ido incrementándose de forma vertiginosa provocando la búsqueda de nuevas tecnologías para poder adaptarse a su continuo crecimiento. Una de las consecuencias del crecimiento es la necesidad de implementar nuevas soluciones más eficientes y con un menor consumo energético. Con este propósito nacen tecnologías como las comunicaciones ópticas inalámbricas (*VLC*) y otras como *LoRa* de bajo consumo.

La comunicación por luz visible (*VLC*, siglas en inglés *Visible Light Communications*) es una modalidad de las comunicaciones ópticas que se basan en transmitir datos a frecuencias de entre 400 y 800 THz en espacio abierto.

Como la iluminación de estado sólido avanza constantemente, la potencialidad de la tecnología *VLC* también se incrementa. Actualmente, todos tenemos en nuestras casas iluminación basada en *LEDs* debido a unas características mucho mayores que la iluminación incandescente. Las principales ventajas son: una alta eficiencia energética, mayor vida útil, su tamaño reducido, resistencia a la rotura, temperaturas de funcionamiento, encendido inmediato, alta capacidad de conmutación, compatibilidad con sistemas de control, amplio espectro de temperaturas de color, entre otras.

Su implementación no requiere grandes esfuerzos de sustitución o nueva incorporación, todos tenemos o podemos tener una iluminación basada en *LEDs*. Esto aumenta su atractivo como solución a problemas de cobertura o a la saturación del espectro. Las características más interesantes de esta tecnología son:

- Capacidad de proporcionar altas velocidades de transmisión.
- Gran robustez frente a condiciones climáticas como aplicación en algunos entornos donde el espectro radioeléctrico se encuentra saturado.
- Carencia de restricciones legales.

Por lo tanto, utilizar *VLC* en interiores, por ejemplo, es una solución adecuada por todas sus grandes ventajas.

La tecnología *LoRa* fue diseñada específicamente para establecer las comunicaciones en el mundo del Internet de las Cosas (*IoT* por sus siglas en inglés, *Internet of Things*). Una tecnología inalámbrica al igual que *Wifi* y *Bluetooth* entre otras, con unas características más afines a proyectos de *IoT*. Dichas características son:

- Muy bajo consumo.
- Baja transferencia de datos.
- Alta tolerancia a las interferencias.

LoRa permite crear redes de sensores de forma rápida y sin un alto coste. Pueden alimentarse con una pequeña batería llegando a lograr una autonomía de unos 10 años [1] con un radio aproximadamente de 16 kilómetros.

## 1.2. Peticionario

Actúa como peticionario del presente Trabajo Fin de Grado (TFG) la Escuela de Ingeniería de Telecomunicación y Electrónica (EITE) de la Universidad de Las Palmas de Gran Canaria (ULPGC) como requisito indispensable para la obtención del título de Graduado en Ingeniería en Tecnologías de la Telecomunicación, tras haber superado con éxito las asignaturas especificadas en el Plan de Estudios.

## 1.3. Objetivos del trabajo

El objetivo principal del proyecto es el diseño e implementación de una interfaz basada en *VLC-LoRa* capaz de proporcionar información de un entorno, comunicarse con una API sencilla y mostrar la información obtenida en una página web logrando así fomentar y abrir un nuevo camino a futuras implementaciones dentro de estas tecnologías.

Este sistema estará compuesto de un enlace *VLC* sencillo que transmitirá tres códigos distintos, un enlace *LoRa* que estarán conectados en dos Arduinos diferentes. El receptor *LoRa* entregará los datos a una API que, posteriormente, serán expuestos en una página web realizada con *Angular*. Una vez validada la implementación se dará por concluido la finalidad de la interfaz y el objetivo del trabajo.

Antes de la implementación del sistema se realizará un estudio del arte de las tecnologías involucradas (*VLC* y *LoRa*) mostrando sus beneficios para las diferentes aplicaciones que puedan llevarse a cabo.

## 1.4. Estructura de la memoria

El presente documento se divide en cuatro partes diferenciadas: Memoria, Pliego de condiciones, Presupuesto y por último Anexo. Con respecto a la memoria, su organización está relacionada con el estudio y desarrollo de las diferentes partes del sistema que se ha llevado a cabo a lo largo de la implementación del sistema. Se ha estructurado en seis capítulos descritos a continuación:

**Capítulo 1.** En este apartado se expone la introducción del trabajo exponiendo un breve preámbulo de las tecnologías VLC y LoRa, enseñando algunas de sus ventajas al igual que algunas de sus posibles aplicaciones. Además, se indican los objetivos por los cuales se ha llevado a cabo el proyecto. Así mismo se añade un apartado donde se muestra una visión global de la estructura del documento.

**Capítulo 2.** En este apartado se conglera un análisis que se ha llevado a cabo sobre la tecnología basada en la comunicación por luz visible. En ella se subdivide en varios apartados que son: una introducción, estandarización de VLC, configuraciones de los enlaces VLC, modulaciones, características, ventajas y desventajas, comparativa con RF y aplicaciones.

**Capítulo 3.** En este apartado se aglomera un análisis realizado sobre la tecnología LoRa exhibiendo sus principales características por las que ha ayudado a crecer dentro del sector de las telecomunicaciones estos últimos años, las funciones para las que ha sido creada. Dentro de este apartado de encuentras varios subapartados que son: una introducción, LoRaWAN, características, ventajas y desventajas, comparativas con otra tecnología IoT y aplicaciones.

**Capítulo 4.** En este apartado se expone todo el hardware y software utilizado durante todo el proyecto.

**Capítulo 5.** En este apartado queda documentado la elaboración realizada a lo largo de la implementación del sistema, configuración, diseño y/o montaje de las diferentes partes. Dentro de este apartado se dividen en subapartados que son: una introducción, enlace VLC, enlace LoRa, aplicación web y, por último, implementación final.

**Capítulo 6.** En este apartado se exponen las conclusiones tomadas gracias al estudio de las tecnologías involucradas y a la completa implementación del sistema.

**Capítulo 7.** En este apartado se ofrece la bibliografía de toda la documentación utilizada para llevar a cabo el proyecto.

## Capítulo II

### Tecnología VLC. Comunicación por luz visible

La tecnología VLC es un método de comunicación donde se utiliza el rango de frecuencias que abarcan la luz visible en medios no guiados tales como la atmósfera. Dentro del espectro electromagnético se encuentran dentro de las frecuencias 400 a 800 THz. En la tecnología VLC usualmente se utilizan LEDs como transmisores gracias a su facilidad de adquirirlos por su bajo coste, porque necesitan poca corriente y porque son sencillos de modular con unos tiempos de respuesta mucho menores que las luces incandescentes y fluorescentes. Los LEDs también son más duraderos y eficientes pudiendo llegar a una vida útil entre unas 30000 y 50000 horas [2].

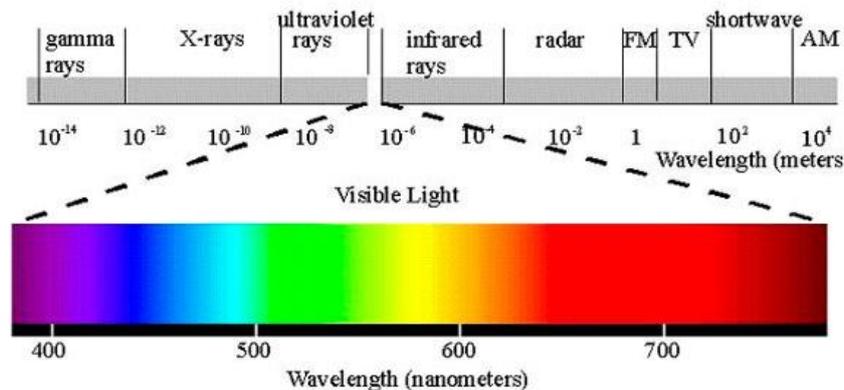


Figura 1. Localización del espectro visible.

En la actualidad, las frecuencias de radio se encuentran cada vez más saturadas debido primero, a que son tecnologías ya confiables y asentadas en el mercado, y segundo, por el aumento de la demanda por parte de la población. El rango de la comunicación por luz visible es un espectro sin restricciones legales a la hora de querer implementarlas en un sistema. Son comunicaciones que pueden llegar a velocidades vertiginosas de 1,72 terabits por segundo a lo largo de una distancia de 10,45 kilómetros utilizando láser para realizar la transmisión [3]. En cambio, con la utilización de LEDs se ha llegado a velocidades de hasta 2 Gbps [4].

Un enlace básico consta de tres bloques esenciales: un transmisor, un receptor y el canal por donde se transmitirán los datos.

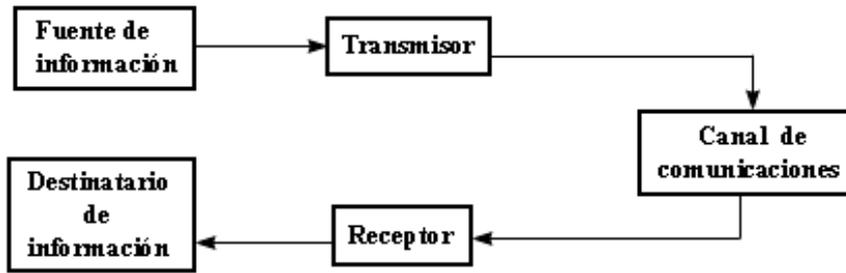


Figura 2. Enlace básico de telecomunicación

En un enlace VLC, el transmisor se suele usar un LEDs de luz blanca (como en el interior de un edificio) o LEDs rojos, verdes o azules pudiendo combinarlos para obtener un color deseado. Existen LEDs de color azul acoplados a una fuente de fósforo. El fósforo absorbe la luz de un color y emite luz de otro color al absorber parte de la energía. Parte de la luz del LED azul es absorbida por el fósforo haciendo así una emisión de un color amarillo y, con la combinación de la luz azul no absorbida, da la tonalidad de luz blanca. La simplicidad del funcionamiento de estos LEDs hace predominante en aplicaciones donde no requiere un cambio de color.

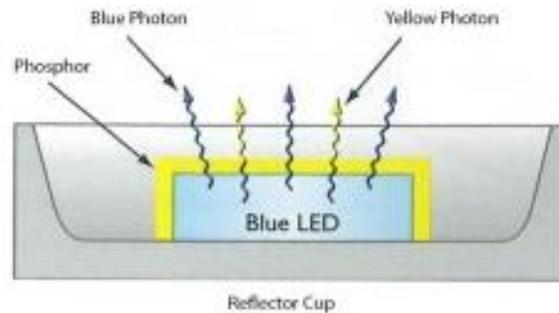


Figura 3. Representación del LED blanco basado en fósforo.

En un sistema VLC, los elementos básicos que lo constituyen son el transmisor óptico (LED, por ejemplo), el medio de transmisión (espacio libre) y el receptor óptico (fotodiodo o fototransistor).

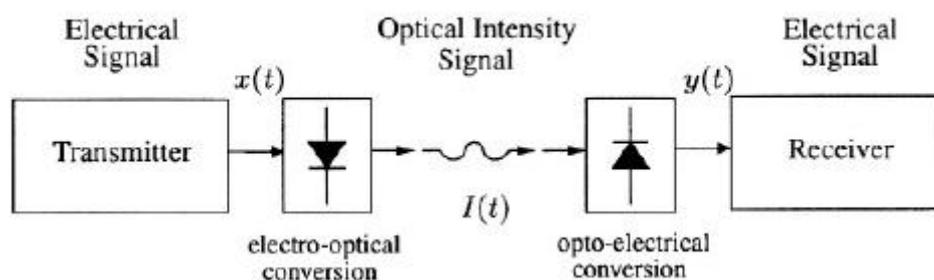


Figura 4. Elementos básicos de un enlace VLC

Para la implementación de sistemas VLC, los dispositivos más recomendados son con la utilización de LASER y LEDs. Estos últimos su implementación es bastante sencilla puesto que se encuentran en cualquier edificio o infraestructura.

## 1. Estandarización de VLC

La estandarización en la tecnología VLC ha sido realizada por el Instituto de Ingeniería y Electrónica o comúnmente conocida por sus siglas en inglés *IEEE (Institute of Electrical and Electronics Engineers)* proponiendo técnicas y esquemas. VLC se conglojera dentro del estándar IEEE 802.15.7. El estándar diseña la capa física (PHY) en tres partes: PHY I, PHY II y PHY III [5]. PHY I opera desde 11.67 kb/s a 266.6 kb / s, PHY II opera desde 1.25Mb/s a 96Mb/s, y PHY III opera desde 12Mb/s a 96 Mb/s [5]. PHY III fue dedicado a múltiples fuentes ópticas utilizando CSK. PHY I y PHY II usan esquemas como OOK (On-Off Keying) y VPPM (Variable Pulse Position Modulation).

Existen otras organizaciones de estandarización existen en la tecnología VLC como la que se encuentra en Japón. El consorcio de comunicación de luz visible (VLCC) de Japón proporciona una plataforma de colaboración para investigadores, universidades e industrias, para mejorar la tecnología VLC. Las actividades del consorcio VLCC son desarrollar y planificar estándares para la tecnología VLC. VLCC propone la utilización de la modulación VPPM para implementar sistemas de comunicación en tecnología VLC.

## 2. Configuraciones de los enlaces VLC

Los enlaces por luz visible pueden emplear múltiples diseños, y es conveniente clasificarlos de acuerdo a dos criterios. El primer criterio es el grado de direccionalidad del transmisor y el receptor. Los enlaces dirigidos emplean transmisores y receptores direccionales, que deben

apuntarse entre sí para establecer un enlace, mientras que los enlaces no direccionados emplean transmisores y receptores con gran ángulo de apertura, lo que alivia la necesidad de dicha señalización. El diseño del enlace dirigido maximiza la eficiencia energética, ya que minimiza la pérdida de trayectoria y la recepción del ruido de la luz ambiental. Por otro lado, los enlaces no dirigidos pueden ser más convenientes de usar, particularmente para terminales móviles ya que no requieren la orientación del transmisor o receptor. También es posible establecer enlaces híbridos que combinan transmisores y receptores con diferentes grados de direccionalidad.

El segundo criterio de clasificación consiste en la existencia de visión directa (LOS) o no (NLOS). Los enlaces LOS se basan en una trayectoria de este tipo, mientras que los enlaces no LOS generalmente dependen de la reflexión de la luz del techo o de alguna otra superficie de reflexión difusa. El diseño de enlace LOS maximiza la eficiencia de potencia y minimiza la distorsión multitrayecto (múltiples haces de luz que llegan al receptor). El diseño del enlace NLOS aumenta la robustez y la facilidad de uso del enlace permitiendo que el enlace funcione incluso cuando los obstáculos, como personas u objetos, se interponen entre el transmisor y el receptor. El diseño de enlaces no dirigidos aumenta la robustez y facilidad de uso de estos sistemas, que a menudo son denominados como enlaces difusos [6].

Dependiendo de las configuraciones empleadas se alcanzarán mayores o menores velocidades de transmisión. La utilización de enlaces LOS permite alcanzar velocidades superiores a otras configuraciones debido a la gran dependencia que existe entre la capacidad del canal y la línea de visión directa entre el transmisor y receptor.

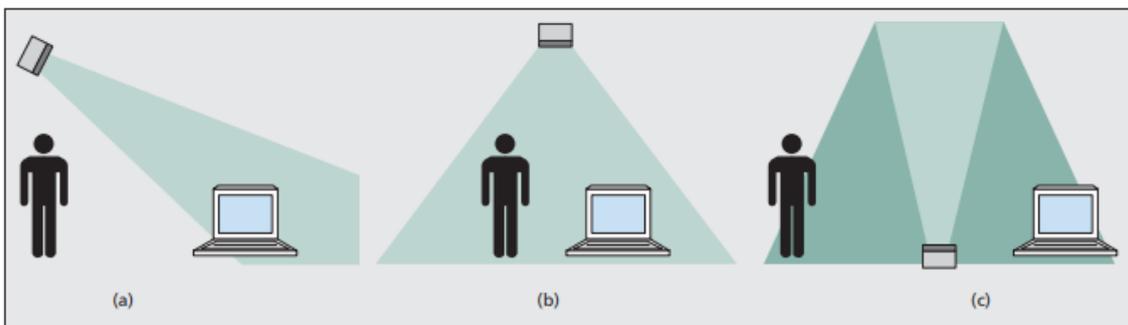


Figura 5. (a) configuración LOS; (b) NLOS; (c) enlace difuso.

### 3. Modulaciones

Las modulaciones más comunes para los sistemas VLC son basados en IM/DD (Intensity Modulation / Direct Detection) donde la luz emitida por el LED depende linealmente de la

corriente que lo excite y, para su recepción, la utilización de fotodetectores que generan una corriente proporcional a la potencia óptica recibida. A continuación, se presenta un esquema con las modulaciones que se exponen en este apartado.

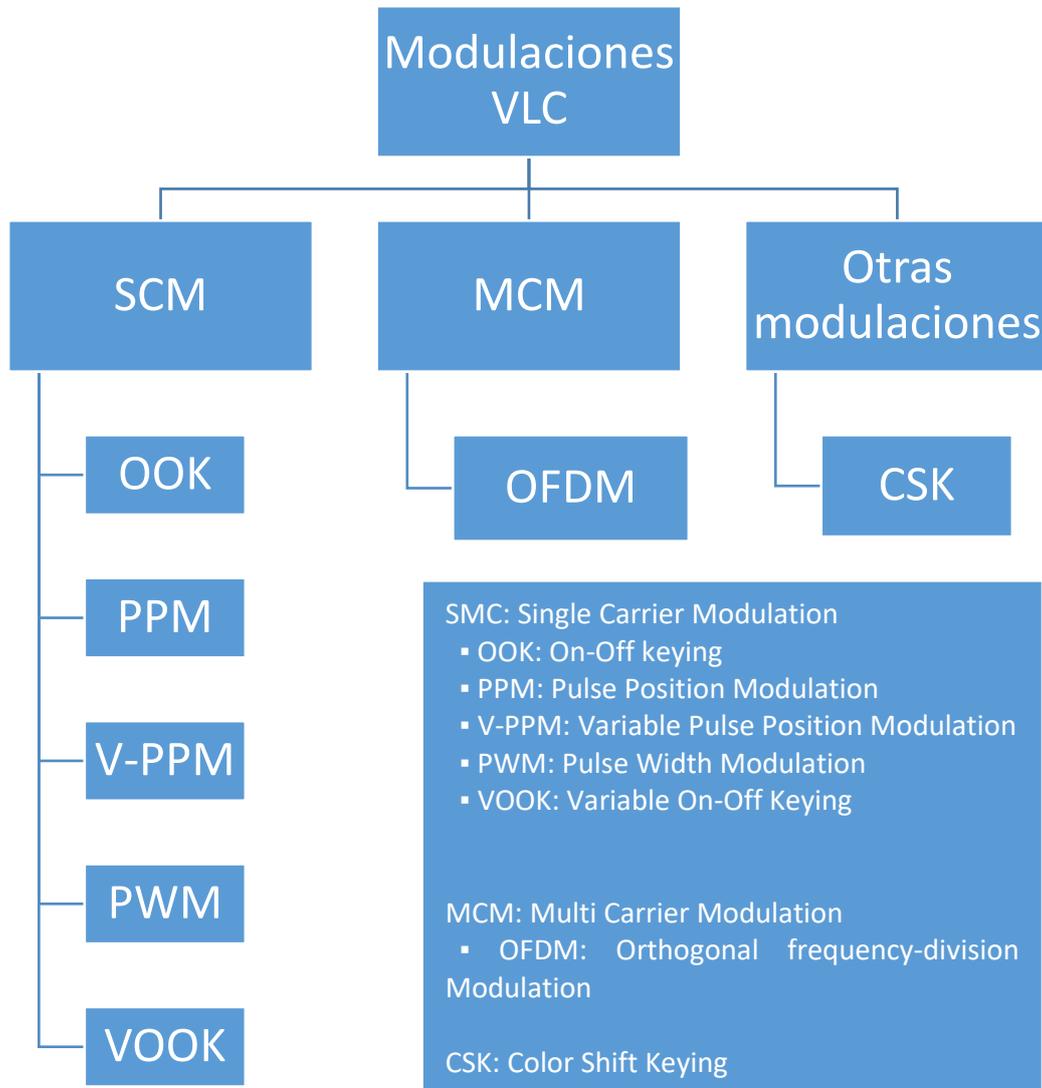


Figura 6. Esquema de modulaciones

### OOK (On-Off Keying)

OOK es la técnica que normalmente se utiliza para las comunicaciones ópticas por cable gracias a su simplicidad y facilidad a la hora de su implementación.

Los bits de datos “1” y “0” están representados por el pagado o encendido del LED respectivamente. En estado apagado, la intensidad de la luz se disminuye sin apagarlo completamente. En su forma más sencilla, el “1” representa un pulso de luz y el “0” su

ausencia. Con la utilización del código Manchester elimina el parpadeo provocado por largas series de ceros o unos gracias a que la potencia media es constante.

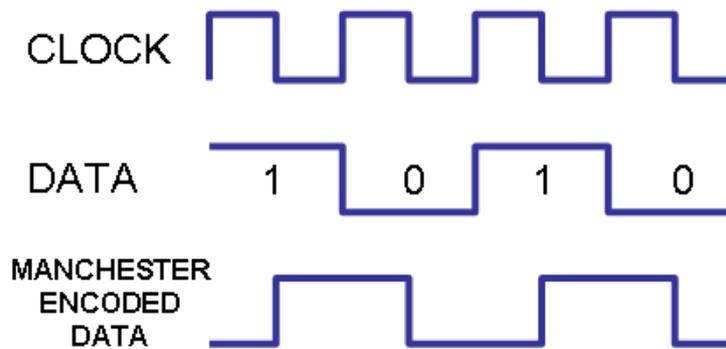


Figura 7. Modulación On-Off Keying

### PPM (Pulse Position Modulation).

La modulación por posición del pulso consiste en la codificación de M bits de palabra en un solo pulso donde la amplitud y su ancho son constantes, pero su posición variable donde  $2^M$  es el número de posiciones posibles.

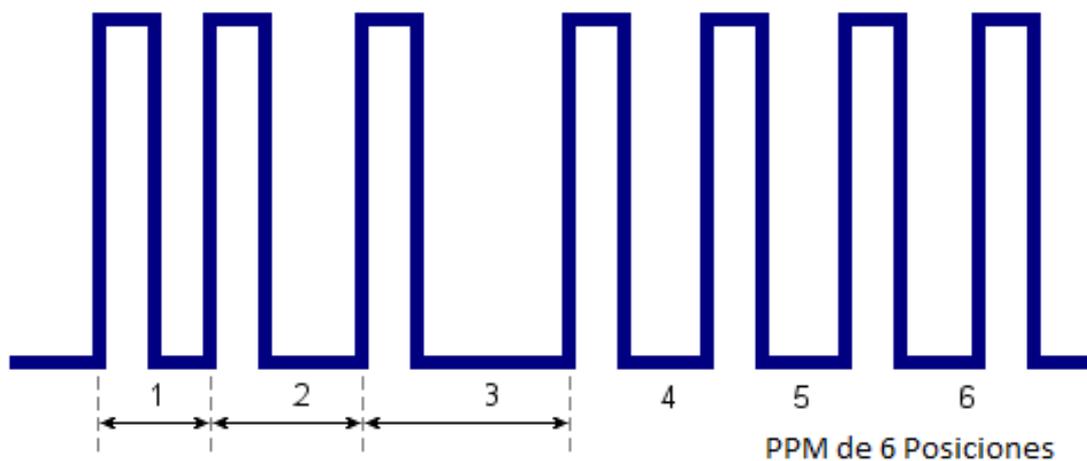


Figura 8. Ejemplo de PPM.

Esta técnica es comúnmente utilizada en enlaces ópticos inalámbricos, gracias a su alta eficiencia en potencia promedio. El principal inconveniente de esta técnica es la necesidad de complejas técnicas de sincronismo para garantizar su correcta detección.

Existe una modificación de ésta que es llamada VPPM. Sin embargo, la tasa de bits no es constante durante las transmisiones y esto afecta al rendimiento del sistema como dispositivo de iluminación.

### **PWM (Pulse Width Modulation).**

En la señal modulada por ancho de pulso (PWM), la amplitud y la posición de los pulsos son constantes, mientras que el ancho de los pulsos varía proporcionalmente con la amplitud de la señal útil analógica.

Con la utilización de esta modulación, se puede lograr que el LED parpadee a una velocidad indetectable para el ojo humano; los parpadeos se escalonan dependiendo de la cantidad de luz deseada. Su inconveniente es que es difícil su implementación pues que requiere de circuitos especiales que incluyen microprocesadores y osciladores. Además, la reducción del voltaje también puede atenuar los LEDs, pero el efecto de atenuación no es lineal ya que, si se reduce a un voltaje por debajo del mínimo para empezar a iluminar, este se apagará repentinamente. [8]

### **VOOK (Variable On-Off Keying).**

VOOK es una variación de OOK, añadiendo diferentes niveles de luminosidad para aumentar la cantidad de palabras. Esta modulación consiste en emitir un pulso rectangular de duración  $T$  y de intensidad  $2P$  para indicar un “1” y ningún pulso para un “0”. En condiciones de entrada equiprobable, OOK puede proporcionar un brillo completo sin necesidad de un control de ajuste. La señal OOK es:  $x(t) = 2Pd \Pi(\frac{t}{T})$ , donde  $d \in \{0, 1\}$  es un bit de información, y  $\Pi(x)$  el pulso que se define como “1” para  $0 < x \leq 1$  y “0” para lo contrario [10].

### **OFDM (Orthogonal Frequency-Division Modulation).**

La multiplexación por división de frecuencia ortogonal (OFDM) es utilizada para lograr altas velocidades de datos en sistemas VLC donde se utilizan un array de LEDs u otras configuraciones donde hay más de un LED. Su utilización es debido a su alta resistencia a interferencias entre símbolos y su alta eficiencia espectral. Los sistemas VLC modulan la potencia instantánea del LED, lo que significa que la señal transmitida debe ser positiva y real puesto que la luz no es posible polarizarla negativamente. Dado que las señales OFDM de RF en el dominio del tiempo convencionales suelen ser bipolares y complejas, las OFDM necesitan una adaptación para sistemas por luz visible. Para ello, se hace uso de la simetría

conjugada en la frecuencia OFDM para que la salida de la transformada rápida de Fourier (IFFT) sea real [11].

### CSK (Color Shift Keying).

La modulación CSK fue diseñada para la utilización de LEDs RGB, que pueden lograr cualquier color que se encuentre dentro de su gama cromática por la combinación del rojo, verde y azul a distintos niveles de intensidad. El objetivo de esta modulación consiste en la selección de coordenadas cromáticas dentro del gamut de los LEDs RGB utilizados para la transmisión de símbolos. La intensidad media final se mantendrá constante como el valor medio de la percepción cromática [9].

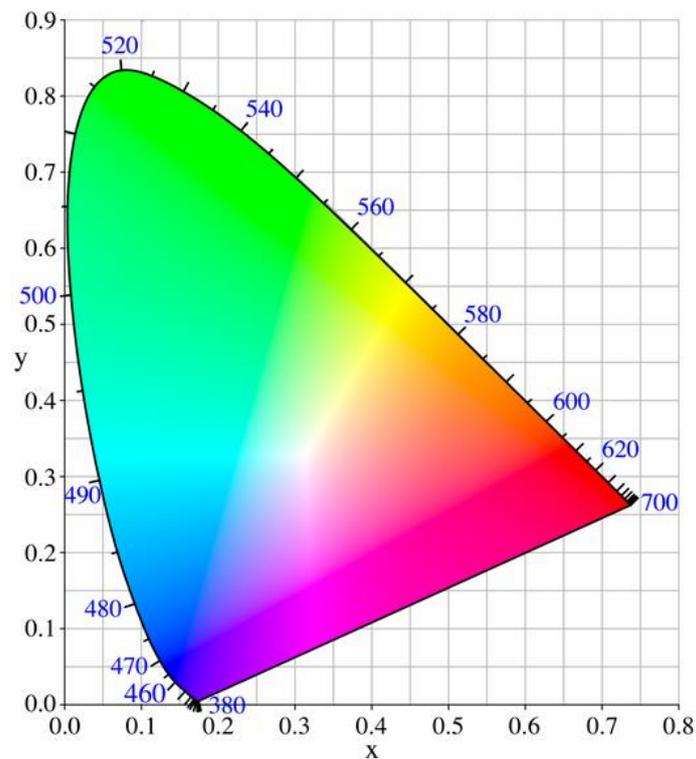


Figura 9. Gamut de un LED RGB

## 4. Características

A continuación, se explicarán las características principales para posteriormente continuar con las ventajas y desventajas de esta tecnología.

- **Ancho de banda:** Las comunicaciones por luz visible utilizan la banda de 400 a 800 THz que coincide con la banda de la región del espectro visible. El espectro radioeléctrico determinado por la UIT (Unión Internacional de telecomunicaciones) se comprende de 3 KHz a 300 GHz. Rangos más elevados de 300 GHz, la absorción

de la radiación electromagnética por la atmósfera es tan grande que resulta impracticable las comunicaciones por emisiones electromagnéticas, hasta que se vuelve nuevamente transparente cerca del infrarrojo y en los rangos de la luz visible. Por consiguiente, el ancho de banda utilizada en los sistemas VLC son de una magnitud 1000 veces más que las comunicaciones por radio.

- **Espectro libre:** El rango de frecuencias comprendida en todo el espectro visible que no requieren de licencias para su uso y emisión. Por lo que, en la implementación de sistemas VLC pueden realizarse de forma inmediata.
- **Visibilidad:** Dado que la emisión en este espectro se encuentra en el rango visible, existe una iluminación apreciable.
- **Seguridad:** Cabe señalar que la realización de transmisiones a través de luz puede encontrarse problemas de seguridad, sobre todo para el ojo humano. En ciertas modulaciones como la modulación PHY de VLC puede hacer que la luz parpadee. El parpadeo de la luz puede tener efectos nocivos para la salud en los animales o personas.
- **Computación ubicua:** Como todos los dispositivos que producen luz (tales como lámparas de interior/ exterior, señales de tráfico, faros de vehículos entre otros) son utilizados en todas partes pueden aprovecharse para implantación de sistemas VLC.
- **Económico:** Puesto que los fotodiodos LED son cada vez más baratos, la instauración de un sistema VLC es cada vez más barata. O en el caso de cualquier edificio que ya dispone de iluminación, solo requeriría la circuitería para su modulación.

## 5. Ventajas y desventajas.

Con la información del apartado anterior, se ha realizado una tabla resumiendo sus ventajas e inconvenientes.

VENTAJAS	INCONVENIENTES
Espectro sin licencia	Cobertura
Reutilización del espectro	Interferencias con otras luces
Ancho de Banda	Dispersión de la luz
Seguridad	Desvanecimientos
Privacidad	Pueden ser interceptadas
Ubicua	
Económica	
Consumo energético	

Sin riesgo para el ser humano
No interfiere con fuentes de radio
Velocidad de transmisión

*Tabla 1. Ventajas y desventajas.*

Empezando por las ventajas, VLC no requiere licencias y su uso es libre respetando las limitaciones en los niveles de potencia transmitida. Por esa razón, es una posible solución para paliar la congestión existente en las bandas ISM (Industrial Scientific and Medical) que se emplean en muchas de las redes inalámbricas existentes.

La reutilización del espectro en las comunicaciones VLC es una de las ventajas más fuertes de esta tecnología. Puesto que, poniendo como escenario un edificio, entre las diferentes habitaciones pueden reutilizarse los mismos canales dado que las fuentes no atraviesan las paredes y quedando confinadas. Su confinamiento también puede ser una desventaja si lo que se busca es la capacidad de penetración para albergar mayor cobertura. La luz es incapaz de penetrar casi ningún objeto o cuerpo exceptuando algunos que son transparentes. Este efecto aumenta en la seguridad del sistema puesto que no puede ser interceptada desde exteriores. En cambio, si se utilizan en exteriores, usuarios ajenos pueden recopilar la fuente de información por ello, en escenarios como este, debe codificarse la información bajando así el uso útil del canal.

El rango de frecuencias llega a ser unas 10000 veces más que las frecuencias ubicadas en el espectro radioeléctrico. Esto permite el uso de transmisiones de datos muy altas, pero depende del fotodiodo y las técnicas de modulación a utilizar.

Respecto a consumo energético, en la utilización de fotodiodos LED llegan a un 40% en ahorro de consumo en comparación con las fuentes de luz incandescente [7] pero sigue mejorando al paso del tiempo.

## **6. Comparativa con RF.**

Puesto que se está hablando de una tecnología emergente que utiliza un espectro con poco uso, es interesante realizar una comparativa con las comunicaciones más comunes que son las basadas en radiofrecuencia (RF).

Como ya se ha expuesto anteriormente, el espectro de radiofrecuencia es un bien muy escaso y valioso. La situación para las telecomunicaciones ha empezado a complicarse desde el suceso ocurrido en Londres donde, por primera vez, se afrontó la posibilidad real de que la red de comunicaciones londinense fallara por un exceso de demanda. Desde entonces, muchos gobiernos han empezado a buscar nuevas soluciones para aliviar la carga de sus redes de comunicaciones que está prevista que vaya en aumento y no de forma lineal. Una solución viable fue cuando se realizó el apagón analógico de las televisiones puesto que quedó libre la banda de 800 MHz [12]. Por ello, una de las ventajas más destacables del VLC es la disposición espectro electromagnético con capacidad para una inmensidad de canales dependiendo de las técnicas a utilizar y filtros ópticos.

VLC posee una cobertura espacial y rango de alcance muy inferior al de las radiofrecuencias. Por consiguiente, es lógico el uso extendido de frecuencias del espectro RF donde se busca grandes rangos y coberturas. Pero, para recintos en interiores, es eficaz el uso de comunicaciones por luz visible ya que las señales no logran penetrar las paredes quedando confinadas. Esto aumenta su seguridad y su reutilización del espectro.

En conclusión, VLC puede ser una buena sustitución para tecnologías como WIFI que son ampliamente usadas dentro de edificios y así, lograr una descongestión del espectro y aumentar la eficiencia de este.

## **7. Aplicaciones**

El interés por la tecnología VLC se debe gracias a la gran cantidad de aplicaciones posibles. A continuación, se indicarán las aplicaciones más interesantes en destacar.

### **LIFI**

VLC es una alternativa como sustitución de la tecnología WIFI dentro de los hogares permitiendo un alivio dentro de las bandas ISM que son usadas frecuentemente. LIFI es la similitud con WIFI, pero en VLC.

Como las señales ópticas no pueden atravesar paredes, para una cobertura total dentro del hogar, será necesario colocar bombillas LED en todas las habitaciones. Sin mencionar que LIFI requiere que la bombilla esté encendida en todo momento, lo que significa que las luces deberán estar encendidas durante el día.

Sin embargo, se anularán muchos problemas de seguridad asociados con las redes de banda ancha compartidas y, a menudo, sobrecargadas.

### **Localización en interiores**

La señal de GPS no llega a interiores de edificios. Como posible solución, es la utilización de VLC en estos entornos puesto que todos poseen iluminación. Puede llegar a ser muy útil en escenarios donde es importante la localización de sujetos como en hospitales.

### **Transporte**

La iluminación de los vehículos está siendo sustituida por LEDs por sus grandes características como la eficiencia energética y su luminancia. Es posible diseñar sistemas basados en VLC para la transmisión de datos críticos o importantes (la velocidad, aviso de obstáculo, por ejemplo) a vehículos de alrededor para la prevención de accidentes o mejora de circulación.

### **Comunicaciones submarinas**

Las comunicaciones submarinas siempre han sido una gran problemática debido a la gran absorción de longitudes de onda del espectro radioeléctrico que posee el agua. Por el contrario, el agua tiene una estrecha ventana de transparencia que incluye el espectro visible. Esto da la ocasión de la utilización de VLC como posible solución.

Esta tecnología abarca grandes posibilidades para crear aplicaciones nuevas dentro del ámbito comercial como gubernamental o doméstico. Los continuos avances dentro de esta tecnología lo confirman.

Grandes empresas como Movistar muestran un creciente interés por esta tecnología. Un artículo del 2013 notifica que han empezado a emprender nuevos proyectos para investigar su viabilidad y la posible sustitución o convivencia con la tecnología WIFI [13].

Existen propuestas para la utilización de VLC dentro del marco de Smart cities. Se estudian sus posibilidades dentro de entornos como el mobiliario urbano, luminarias, vehículos, electrodomésticos y otros aparatos que puedan enviar información para facilitar una mejor desenvolvura y movilidad de la ciudadanía e, incluso, una mejora de su seguridad. [14].

Para concluir, la Unión Europea también muestra interés en este tipo de proyectos. Dentro de la comisión europea CORDIS, que financia proyectos concretos de innovación y desarrollo, ha financiado varios proyectos relacionados con VLC [15] [16].

# Capítulo III

## Tecnología LoRa. Long-Range.

LoRa es una técnica de modulación de espectro ensanchado patentado por Semtech. Utiliza las bandas de frecuencia de licencia libre (bandas ISM); 169 MHz , 433 MHz, 868 MHz en Europa y 915 MHz en América del Norte. Llega a realizar transmisiones de más de 10 km en áreas rurales con un bajo consumo de energía. La tecnología se presenta en dos partes: LoRa, la capa física y LoRaWAN (Red de área amplia de gran alcance), las capas superiores.

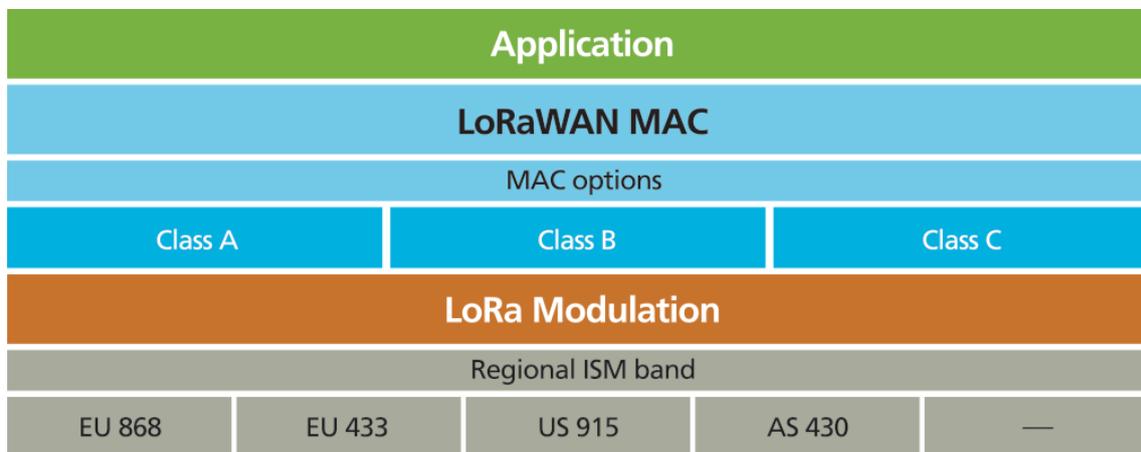


Figura 10. Esquema de LoRa.

Uno de los aspectos más importantes de cualquier tecnología de IoT es cómo se va a comportar una vez haya implementados millones de dispositivos. La solución LoRa puede designarse como CDMA (del inglés Code Division Multiple Access). Esto no solo aumenta la capacidad de la red, sino que también permite la adaptación dinámica de las velocidades de datos del dispositivo. Los dispositivos con un mejor enlace a una puerta de enlace (debido a la proximidad, el entorno de ruido más bajo, línea de visión directa) pueden utilizar velocidades de datos más altas (hasta 11 kbps) y ahorrar batería. Los dispositivos con una calidad de enlace deficiente pueden aumentar el presupuesto del enlace al utilizar tasas de datos más bajas, lo que amplía el alcance efectivo del enlace a más de 30 km en la línea de visión [17].

## 1. LoRaWAN

Como se expuso anteriormente, LoRaWAN es un protocolo de red de área amplia (LPWAN, en inglés Low Power Wide Area Network) de baja potencia, basado en la tecnología LoRa. Diseñado para conectar dispositivos inalámbricos con baterías a Internet en redes regionales, nacionales o globales. El protocolo LoRaWAN aprovecha el espectro de radio sin licencia de la banda ISM. Esta especificación define el dispositivo a la infraestructura de los parámetros de la capa física LoRa y el protocolo LoRaWAN proporcionando una interoperabilidad perfecta entre los dispositivos. Como la tecnología LoRa está patentada por Semtech y proporciona los chips de radio, se fundó una asociación sin fines de lucro llamada LoRa Alliance para impulsar la estandarización y la armonización global del protocolo LoRaWAN por todo el globo [18].

## 2. Características

Las características más destacables de la tecnología LoRa son las siguientes [18]:

- **Largo alcance.** Es capaz de conectar dispositivos a 16 kilómetros de distancia y penetra en entornos urbanos densos o interiores profundos.
- **Bajo consumo.** Su consumo reducido puede lograr una vida útil de la batería de hasta 10 años, minimizando los costos de reemplazo.
- **Seguridad.** Cuenta con encriptación AES128 de extremo a extremo, autenticación mutua, protección de integridad y confidencialidad.
- **Estandarización.** Ofrece interoperabilidad de dispositivos y disponibilidad global de redes LoRaWAN para una rápida implementación de aplicaciones IoT en cualquier escenario.
- **Geolocalización.** Permite aplicaciones de rastreo sin GPS, ofreciendo beneficios únicos de baja potencia que otras tecnologías no poseen.
- **Móvil.** Mantiene comunicaciones con dispositivos en movimiento sin agotar el consumo de energía.
- **Bajo costo.** Reduce la inversión en infraestructura, los gastos de sustitución de la batería y, en última instancia, los gastos operativos.

## 3. Ventajas y desventajas

La tecnología LoRa de Semtech ofrece excepcionales características tales como una alta sensibilidad de recepción gracias a la modulación de espectro ensanchado que permite aumentar drásticamente el rango a bajas tasas de bits, mientras siguen funcionando con una

potencia de salida de batería (20dBm / 100mW). Tiene un gran alcance pese a su bajo consumo ya que puede llegar a distancias entre 10 y 20 kilómetros.

Como en cualquier otra tecnología, posee algunas desventajas si el enfoque que se busca es una gran tasa binaria, puesto que sacrifica altas velocidades de transmisión para optar a un perfil bajo en consumo.

El hecho de que sea una tecnología patentada y propietaria es un gran punto en contra puesto que hace la documentación más difícil de consultar. También encarece su utilización ya que requiere de licencia y los chips son exclusivos de Semtech.

#### **4. Comparativa con SigFox**

SigFox es una compañía francesa proveedora de redes IoT que aspira a convertirse en el mayor proveedor global de redes IoT. Sus características más destacables son: el manejo de los mensajes haciendo peticiones HTTP a un backend preconfigurado, la gestión por parte de SigFox puesto que se encarga completamente de la comunicación entre los dispositivos IoT haciendo muy sencillo la integración del módulo de radio.

Por otra parte, SigFox impone una restricción de mensajes a través de su red, donde cada dispositivo solo puede enviar unos 140 mensajes al día con límite de 7 cada hora y la recepción de 4 mensajes por día [19]. Además, para el uso de la red es necesario darse de alta y la suscripción a dicha red aumentando los costes para su uso puesto y, si en un futuro se da de baja, el módulo deja de cumplir su función [20].

Comparando SigFox con LoRa, la restricción de mensajes en la red de SigFox es un punto en su contra ya que LoRa no posee restricciones. El hecho de, aparte de comprar el módulo SigFox, es necesario una suscripción cuyo coste ronda los 16 euros al año por dispositivo.

LoRa parece ser una solución más permisiva que permite al usuario final adaptar el sistema a sus necesidades y la transmisión de bajas tasas binarias con un largo alcance transmitiendo libremente y sin restricciones.

#### **5. Aplicaciones**

Las aplicaciones de la tecnología LoRa son muy diversas y numerosas gracias a su carácter de licencia libre. A continuación, se describen las más interesantes:

- **Monitorización.** La integración de LoRa en redes de sensores para la monitorización de cualquier proceso, dispositivo o máquina de forma remota con un mantenimiento muy bajo puesto que la vida útil de la batería con módulos LoRa puede llegar a los 10 años.
- **Localización.** Es perfecto para aplicaciones de localización de vehículos u otros objetos ya que puedes diseñar tu red de módulos LoRa de formas muy diversas.
- **Seguridad.** Instalaciones de sensores en recintos conectados dispositivos LoRa.
- **Sector automovilístico.** La integración de sensores en los vehículos interconectados con la red LoRaWAN. Realizar comunicaciones entre vehículos o estaciones para obtener nueva información sobre el estado actual de carreteras.
- **Agricultura y ganadería.** Una red de sensores en los campos de cultivo monitorizando los niveles de temperatura, humedad, luminancia, entre otras. En la ganadería puede realizarse un seguimiento de los animales.

La asociación LoRa Alliance cada vez aumenta en miembros rápidamente logrando ganar confianza dentro del sector de las telecomunicaciones y avanzando a su integración de esta tecnología dentro de la actividad comercial, industrial y científica. Recientemente, LoRa Alliance anunció la disponibilidad pública de tres nuevas especificaciones . Juntas, las nuevas especificaciones admiten y estandarizan las actualizaciones de firmware a través del aire (FUOTA, Firmware Updates Over The Air), una capacidad exclusiva de LoRaWAN entre las redes de área amplia de baja potencia [21].

La incorporación de esta tecnología está sucediendo rápidamente. Se puede considerar una buena sustitución a las antiguas redes 2G que cada vez se encuentran más en desuso y lograr así una red diseñada especialmente para el mundo del internet de las cosas.

# Capítulo IV

## Hardware y software utilizado

### 1. Hardware empleado

#### Ordenador portátil

El dispositivo utilizado para el desarrollo del código de las diferentes partes del proyecto es un Asus R510JX-DM301D mostrado en la siguiente figura.



*Figura 11. Ordenador portátil.*

#### Arduino Uno R3

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Tiene 14 entradas/salidas digitales, de las cuales 6 pueden utilizarse como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteo. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador.



Figura 12. Arduino Uno R3

#### Características técnicas

Microcontrolador	ATmega328
Voltaje	5 V
Voltaje entrada (recomendado)	7-12 V
Voltaje entrada (límites)	6-20 V
E/S digitales	14 (de los cuales 6 son salida PWM)
E/S analógicas	6
Corriente DC para pines E/S	40 mA
Corriente DC para pines 3.3V	50 mA
Memoria Flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz



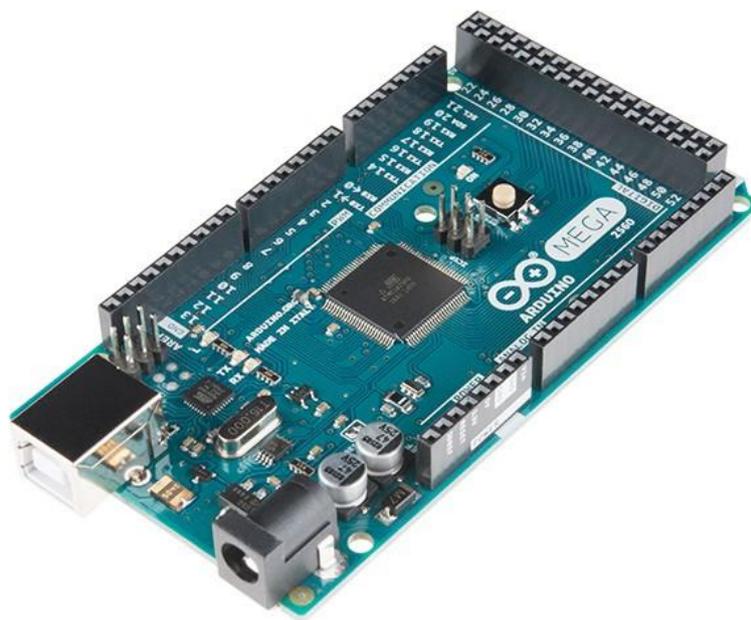


Figura 14. Arduino Mega 2560 R3

#### Características técnicas

Microcontrolador	ATmega2560
Voltaje	5 V
Voltaje entrada (recomendado)	7-12 V
Voltaje entrada (límites)	6-20 V
E/S digitales	54 (de los cuales 15 son salida PWM)
E/S analógicas	16
Corriente DC para pines E/S	40 mA
Corriente DC para pines 3.3V	50 mA
Memoria Flash	256 KB
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

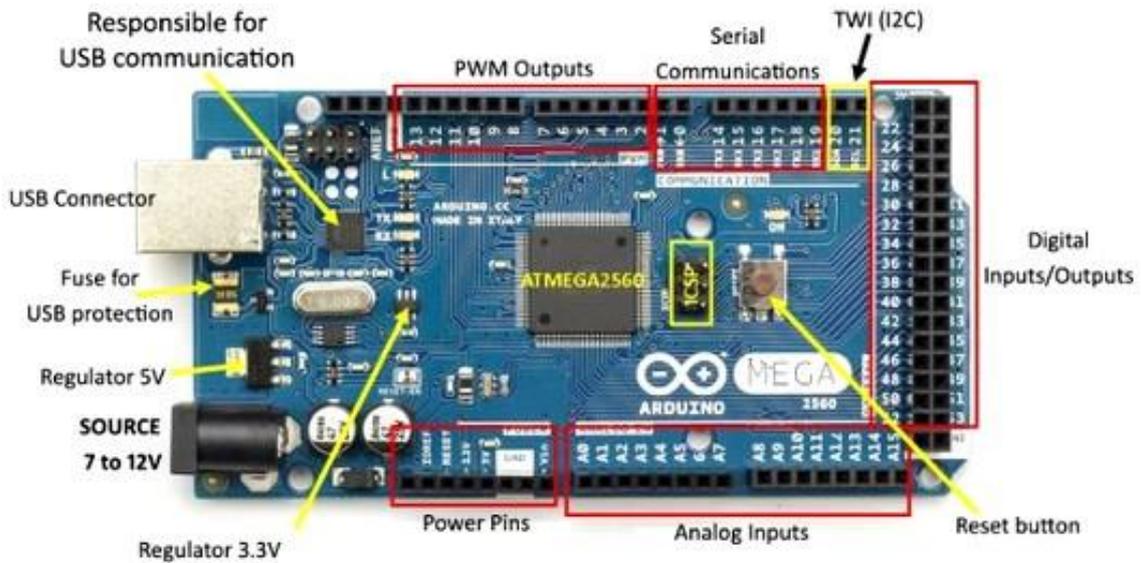


Figura 15. Esquema del Arduino Mega 2560 R3.

## Lámpara VLC

La lámpara VLC es el transmisor del enlace VLC implementado en el proyecto. Se conecta directamente a la electricidad doméstica. Utiliza modulación OOK. Posee una perilla rotativa para el cambio de código. Los códigos que emite son: R, G y B. No requiere configuración adicional puesto que el microcontrolador que posee ya lo está.



Figura 16. Lámpara VLC. Transmisor

## Receptor VLC

El receptor VLC consta de una agrupación de cuatro fototransistores. Está rodeada por una anilla de plástico para minimizar la recepción de otras fuentes cercanas de iluminación. Posee 4 pines: dos de conexión a tierra, transmisor y alimentación de 5V. Recibe a 9600 baudios.

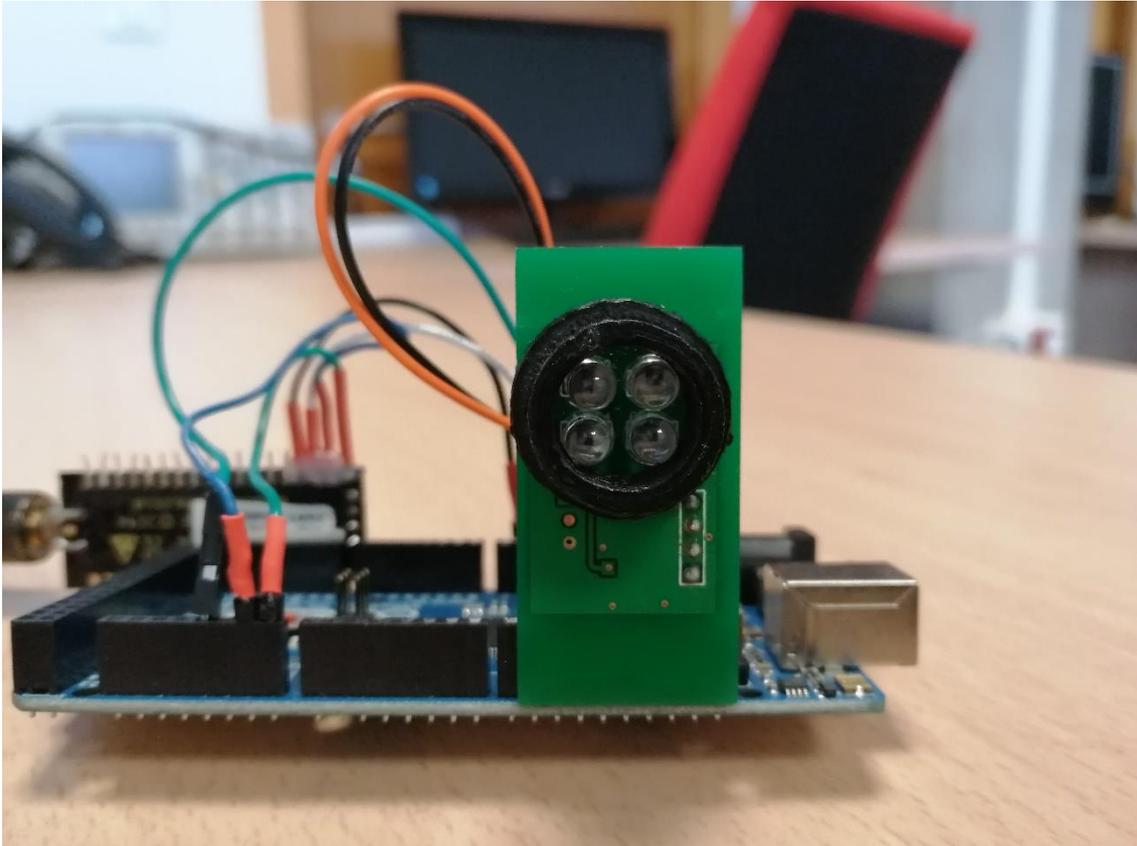


Figura 17. Receptor VLC.

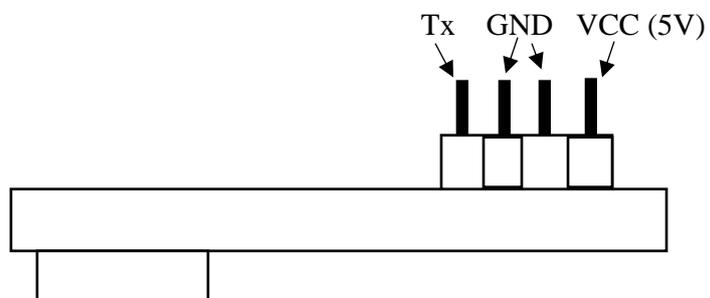
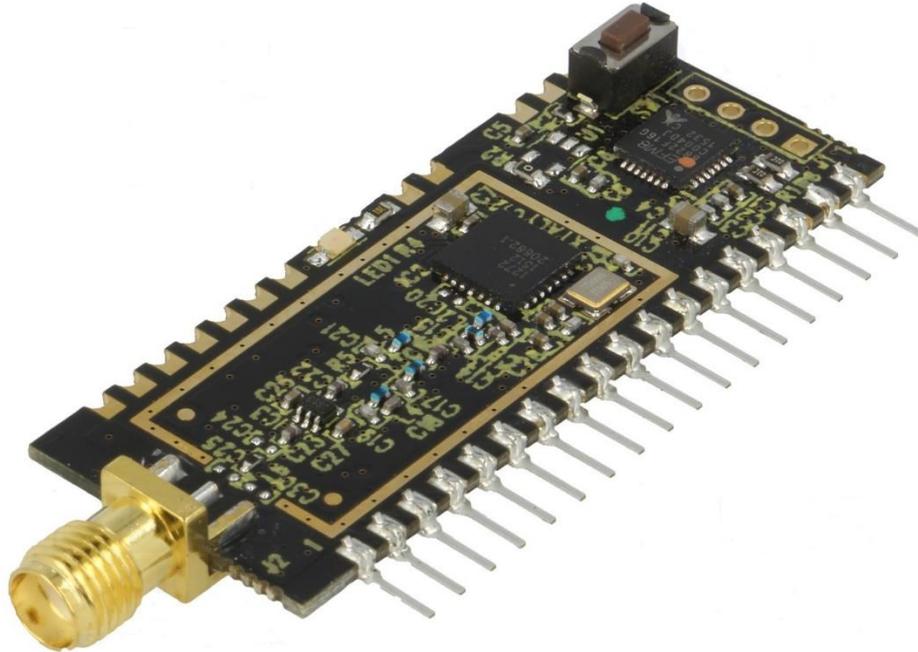


Figura 18. Esquema del Receptor VLC.

## **GAMMA-868**

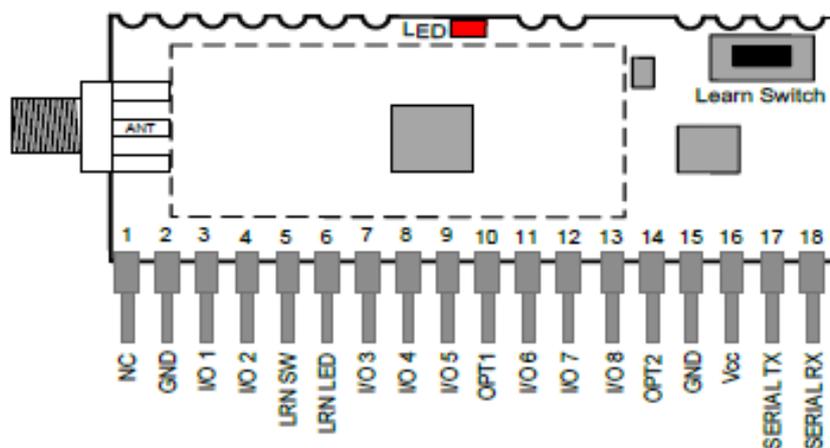
El módulo GAMMA-868 consta de 18 pines. Posee 2 modos de funcionamiento: Transceptor telemétrico (Tx, Rx o transceptor) y como Módem de datos serie. En el proyecto es utilizado como Módem de datos serie. Para ello, es necesario conectar el pin OPT2 con tierra.



*Figura 19. Dispositivo GAMMA-868.*

### Características técnicas

- Alcance de hasta 16km.
- Comunicaciones de datos serie a 38,4Kbps
- 8 entradas de conmutación.
- 8 salidas digitales.
- 4 canales de reconocimiento.
- Tecnología de espectro ensanchado.
- No requiere de componentes externos.
- Baja tensión 1.8-3.6V



## GAMMA-XXX (SIL version)

Figura 20. Esquema del GAMMA-868.

Los comandos de configuración son:

- **#C** para configurar el LoRa mode y la potencia de salida. Hay 6 tipos de LoRa mode en el que la longitud del paquete varía. Para la potencia de salida hay 15 niveles, de menor a mayor potencia. Los valores tanto de LoRa mode como de potencia de salida deben ponerse en valores decimales, pero en bytes.
- **#A** configura la dirección local y de destino. La longitud de las direcciones es de 3 registros y de longitud 3 bytes. Los valores deben ponerse como valores en bytes.
- **#H** habilita o deshabilita la opción de handshaking.
- **#B** configura la velocidad de transmisión.
- **#?** devuelve los valores de configuración actuales.
- **#Q** para reiniciar el dispositivo. Es necesario después de la configuración para que persista en memoria.

## 2. Software empleado

Respecto al software, su conjunto conformará la plataforma de gestión para la recepción y transmisión de los datos obtenidos de LoRa, su almacenamiento y consulta en una base de datos y, por último, la exposición de ellos. Existe variedad de software para las funciones requeridas. Por ello, se ha optado por un software libre que cumpla estas funciones.

## MongoDB

MongoDB es una base de datos de uso gratuito y código abierto. Es un sistema de base de datos NoSQL orientado a documentos. En lugar de salvar los datos en tablas como cualquier otra base de datos relacional, guarda las estructuras de datos en documentos similares a JSON llamado BSON, haciendo que la integración de los datos para ciertas aplicaciones sea más fácil y rápida.



*Figura 21. Logo de MongoDB.*

Las características más importantes de esta plataforma son [22]:

- **Indexación.** Todos los documentos de MongoDB pueden ser indexados, incluso realizar subíndices.
- **Consultas Ad hoc.** Pueden realizarse consultas en las que devuelvan un campo concreto del documento o una función JavaScript realizada por el usuario. MongoDB soporta búsquedas por rangos y campos.
- **Replicación.** MongoDB soporta replicación del tipo primario-secundario. Grupos de primario y sus secundarios se llaman replica set. Los primarios pueden ser ejecutados por comandos para lectura y escritura. En cambio, como los secundarios replican los datos del primario, solo puede usar la lectura o para copias de seguridad.
- **Balanceo de carga.** MongoDB utiliza el concepto de “shard” para su escalabilidad. Los datos son divididos en rangos basados en una clave sharding, que es elige el desarrollador, y se distribuye por múltiples “shard”. MongoDB puede ejecutarse en múltiples servidores balanceando la carga o replicando datos para que el sistema se mantenga operativo en caso de algún fallo de hardware.
- **Almacenamiento de archivos.** MongoDB utiliza la ventaja de la capacidad que posee en el balanceo de carga. Puede utilizarse también como sistema de archivos llamándose la función GridFS y consiste en una implementación en los drivers exponiendo funciones y métodos para la manipulación de estos.

- **Agregación.** MongoDB proporciona un framework de agregación para realizar operaciones similares a los comandos SQL.
- **Ejecución de JavaScript del lado del servidor.** MongoDB puede realizar consultas usando JavaScript, haciendo que sean enviadas directamente a la base de datos para ejecutarlas.

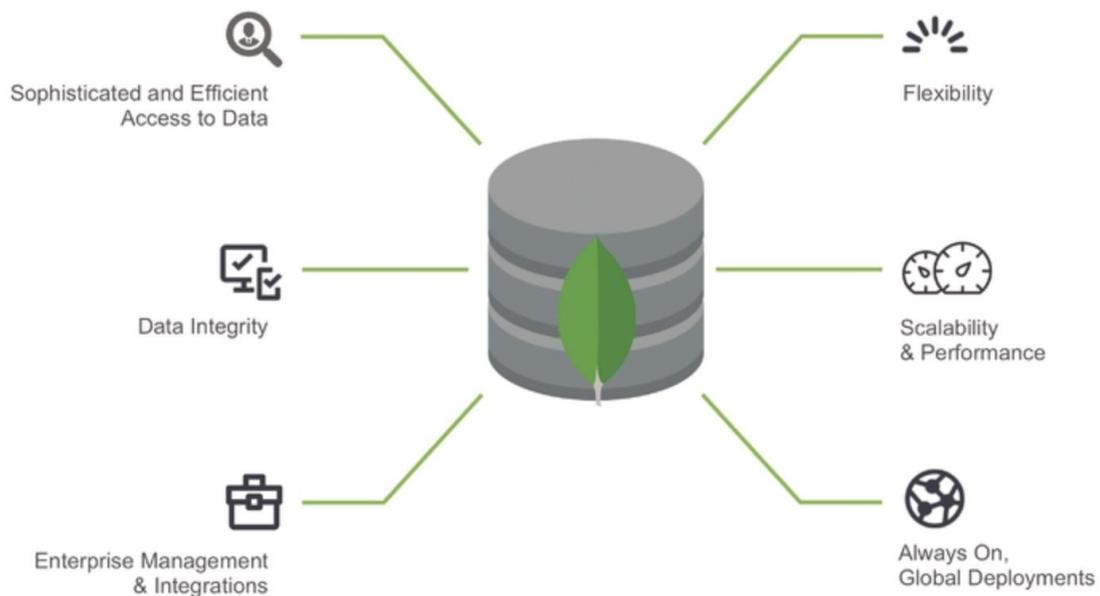


Figura 22. Algunas características de MongoDB.

La justificación de uso de este software ha sido por las siguientes motivaciones:

- Es una base de datos de código abierto y de uso gratuito.
- Almacena datos en documentos JSON flexibles. Cada documento puede modificarse en cualquier momento.
- Es fácil de aprender puesto que proporcionan una gran documentación bien detallada.

### Node.JS y Express.JS

Node.js es un entorno de ejecución (framework) para JavaScript construido con el motor de JavaScript V8 de Chrome. Está diseñado para la generación de aplicaciones web de una forma altamente optimizada. En otras palabras, es una forma de ejecutar JavaScript en el lado del servidor basado en eventos.



Figura 23. Logo de Node.js.

Sus características principales son [23]:

- **Compatibilidad Servidor-Cliente.** Puesto que todo se ejecuta en el servidor, no hay que temer a posibles incompatibilidades entre distintos clientes. Toda la ejecución del código del servidor se quedará aislada en el servidor.
- **Programación asíncrona.** Está pensado para la realización de programas sin bloquear la línea de ejecución de código respecto a entradas y salidas. Así, los ciclos de procesamiento quedarán disponibles durante la espera. Todas las APIs de node.js utilizan callbacks para definir las acciones a ejecutar después de cada operación entrada/salida, que se procesarán cuando se las entradas y/o salidas se hayan completado.
- **Programación orientada a eventos.** Como nos encontramos en el lado del servidor, no existen objetos. Todo es a base de eventos. En otras palabras, tanto la escritura como la lectura van asociados por los sucesos que ocurran en el sistema, definidos por el usuario o que ellos mismo provoquen.
- **Gestor de paquetes NPM.** NPM (Node Package Manager) es un gestor de paquetes que posee Node.js en el que tenemos disponible casi cualquier librería a tan solo una línea de comando. Con el simple comando “**npm install [NombreDeLaLibrería]**” tendremos ya en nuestro sistema una nueva librería en cuestión de segundos.

Express es un framework para Node.js enfocado para la creación de aplicaciones web de forma rápida ya que proporciona numerosas funcionalidades para la gestión de los elementos comunes en estos. Es instalado mediante el gestor de paquetes NPM. Sus características principales son su sencillez en la implementación de aplicaciones web, es robusto, flexible y ofrece funciones de enrutación sencillas (peticiones HTTP).



*Figura 24. Logo de Express.JS.*

La justificación de uso tanto de Node.JS como de Express.JS han sido por las siguientes motivaciones:

- Están basados en código abierto y son gratuitos puesto que son software libre.
- El marco de trabajo para el que fueron diseñados encaja perfectamente con las necesidades para la realización de la plataforma web.
- El gestor de paquetes NPM hace posible la incorporación de nuevas librerías de forma inmediata sin guías de instalación enrevesadas.

## **Angular**

Angular es un framework de desarrollo para JavaScript que ha sido creado por Google. Su finalidad es la de facilitar el desarrollo de aplicaciones web SPA (Single-Page Application) que consisten en la creación de una web en una sola página, en la cual la navegación entre las diferentes páginas y secciones se realizan de manera dinámica y asíncrona, haciendo llamadas al servidor. El efecto final es la de una página sin refrescos y no recargan el navegador [24].



*Figura 25. Logo de Angular.JS.*

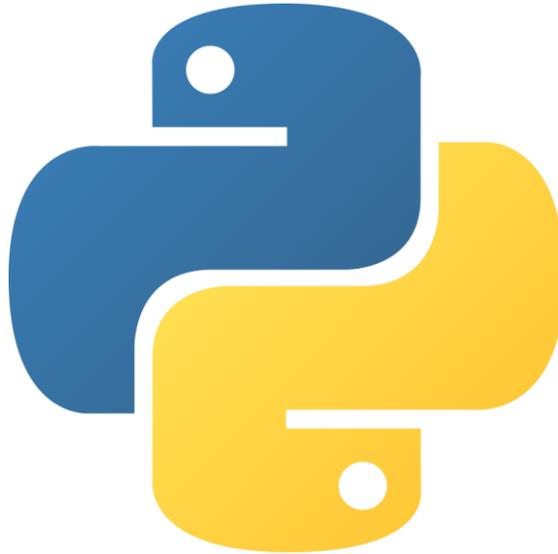
Angular CLI (Command Line Interface) es una herramienta por línea de comandos para utilizar el framework Angular y su objetivo es facilitar la creación de proyectos, componentes, interfaces y otros elementos en el desarrollo de una página web. Es instalada vía NPM, así que su incorporación es fácil y sencilla [25].

En este proyecto, Angular será utilizado para el desarrollo del Front-End de la aplicación web. Las características fundamentales por la que se ha utilizado este software han sido:

- **Es ampliamente utilizada.** Actualmente, dentro del panorama de la programación web, es uno de los frameworks más utilizados. Como posee una comunidad enorme, hay gran cantidad de documentación y de diferentes autores, por lo que su aprendizaje puede ser más llevadera.
- **Es modular.** Se basa en un core (núcleo) y el resto son módulos que permiten acceder a más características cuando sea necesario. Puede integrarse a otras tecnologías como Node.JS.
- **Compilación incremental.** Mientras se está desarrollando la página web, al realizar el guardado, no es necesario la compilación completa nuevamente minorizando así el tiempo de compilación.
- **Software libre.** Es de código abierto y licencia gratuita.

## Python

Python es un lenguaje de programación orientado a objetos, preparado para realizar cualquier tipo de programa cuya filosofía hace especial énfasis en una sintaxis que favorezca un código legible. Su característica más importante es la resolución dinámica de nombres. En otras palabras, es lo que enlaza un método y un nombre de variable durante la ejecución del programa. La justificación de uso ha sido principalmente a dos factores: es un software libre de licencia gratuita y su sencillez a la hora de utilizarlo.



*Figura 26. Logo de Python.*

### **Arduino IDE**

Arduino IDE es un entorno de desarrollo creada por Arduino para escribir y cargar programas en las placas Arduino. Admite lenguajes como Java, C y C++. La programación en Arduino tiene dos zonas diferenciadas: **setup()** y **loop()**.

- **setup()** se encarga de recoger la configuración. Es el lugar donde se debe declarar las variables puesto que solo se ejecuta una vez.
- **loop()** es donde debe contener el programa que se ejecutará cíclicamente. Esta función es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo.

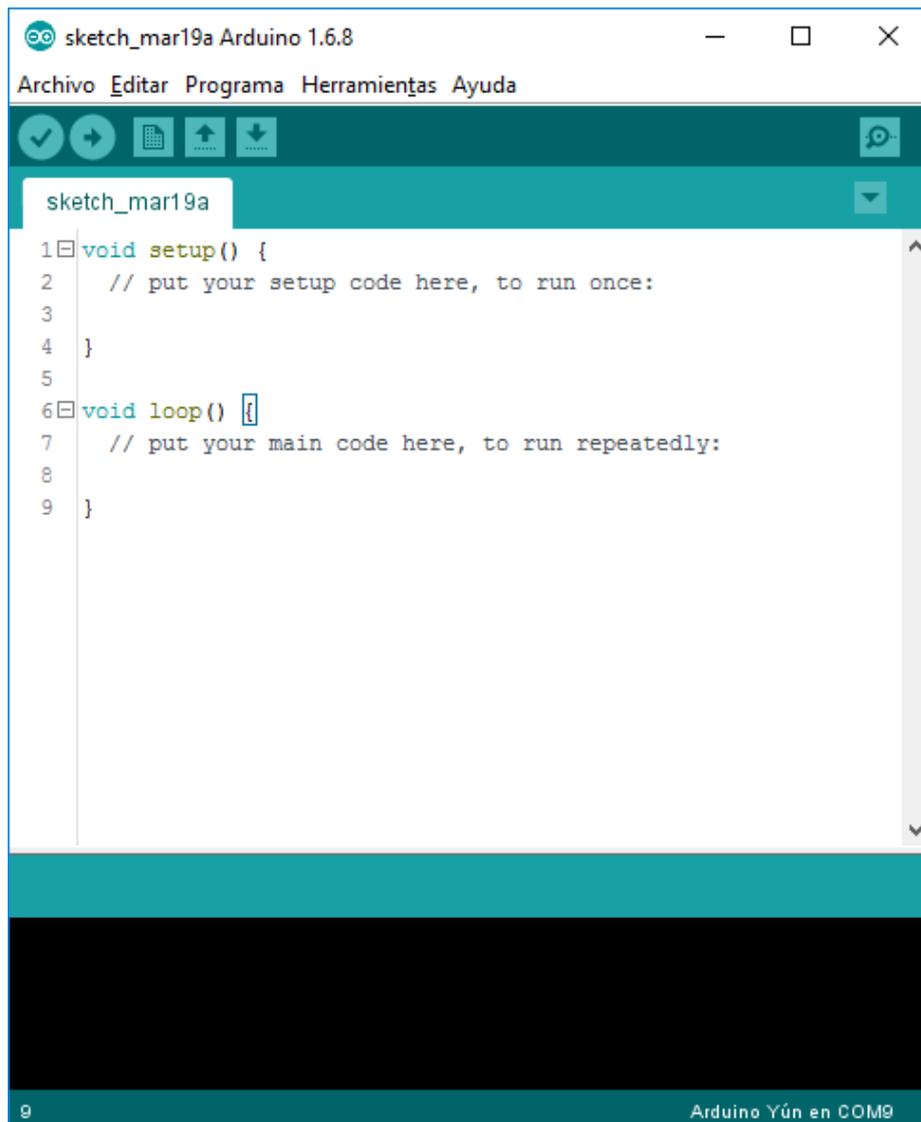


Figura 27. Ventana de Arduino IDE.

## Visual Studio Code

Visual Studio Code es un editor de código fuente realizado por Microsoft. Las características más importantes por las cuales se justifica su uso en el proyecto son porque su descarga y uso es gratuito y por su versatilidad a la hora de escribir diferentes tipos de código.

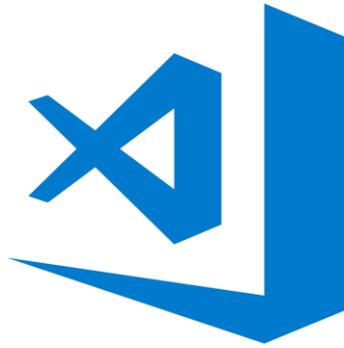


Figura 28. Logo de Visual Studio Code.

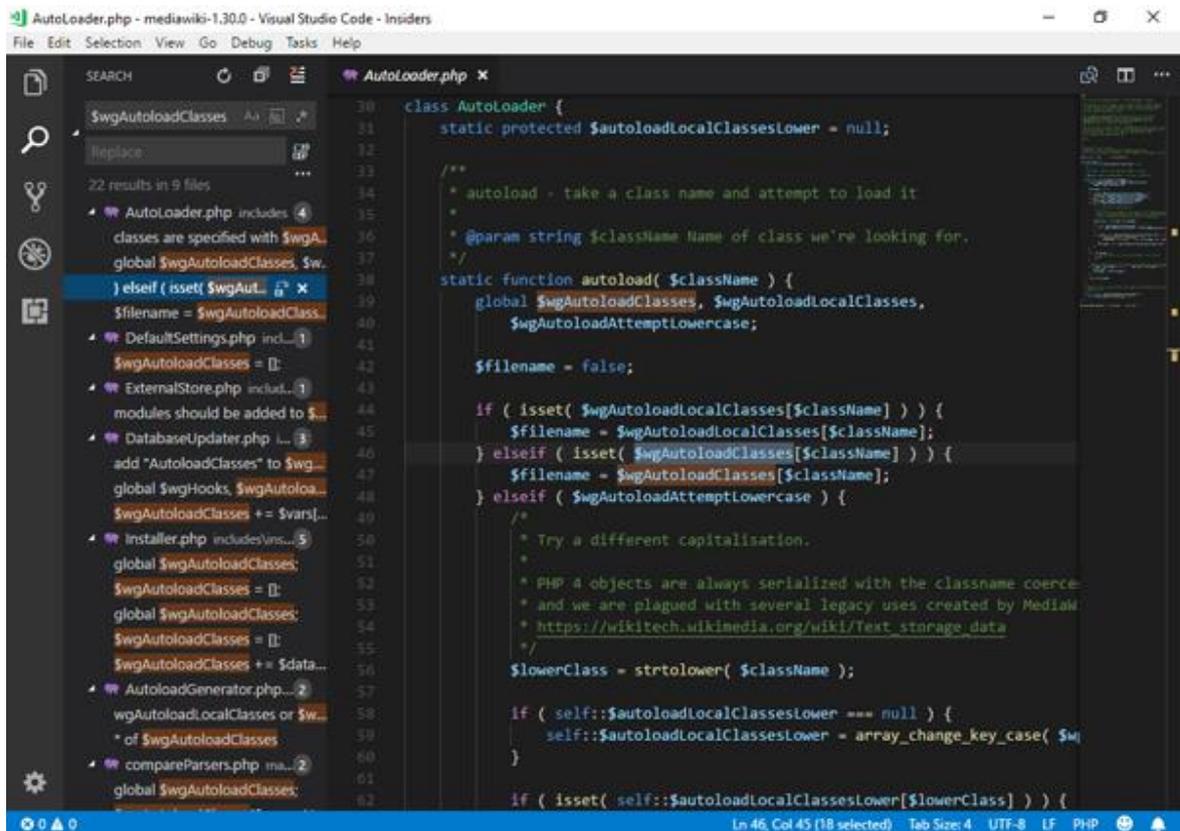


Figura 29. Ventana de Visual Studio Code.

## Postman

Postman es una herramienta diseñada para la prueba y verificación de APIs. Permite la creación de peticiones HTTP y analizar los valores devueltos. Su justificación de uso ha sido por dos factores, es de uso gratuito y su sencillez a la hora de utilizarlo.



Figura 30. Logo de Postman.

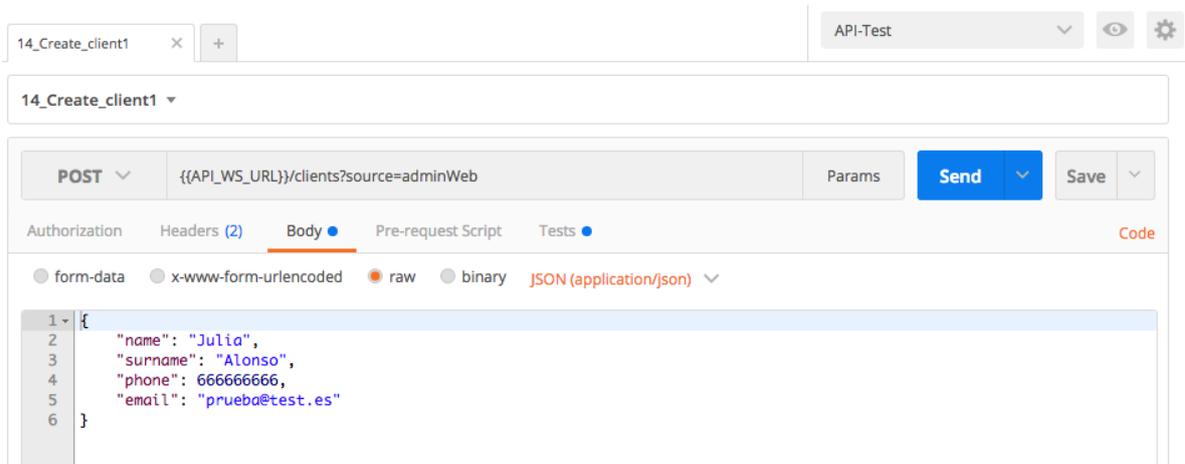


Figura 31. Ventana de Postman.

# Capítulo V

## Implementación del sistema

En este capítulo se describe la implementación del sistema VLC, LoRa y la plataforma de gestión basada en el stack M.E.A.N. Se detallará los procedimientos realizados durante todo el proyecto.

Para obtener una visión más global del sistema, se presenta el siguiente esquema:

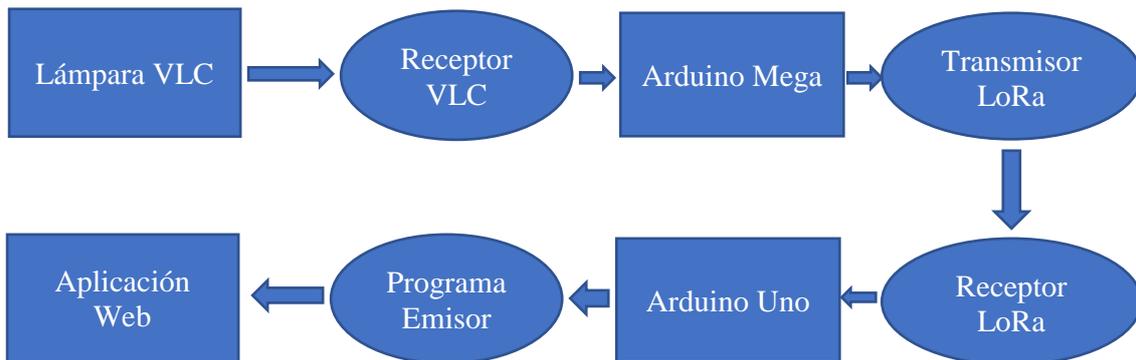


Figura 32. Esquema del sistema.

### 1. Enlace VLC

El sistema VLC consiste en un array de LEDs blancos que actuarán de transmisor emitiendo uno de los tres códigos diferentes. Como receptor se ha diseñado un circuito con cuatro fototransistores en paralelo que recopilarán la señal recibida mediante detección directa. A continuación, el enlace se ilustra en la siguiente imagen:



*Figura 33. Enlace VLC en funcionamiento.*

Tanto el transmisor como el receptor no requieren configuración. El receptor ha sido conectado al Arduino Mega. Se ha utilizado el pin 17 para la recepción de los datos del receptor VLC. En el Arduino Mega, se configura el puerto serie con el nombre **Serial12** a una velocidad de 9600 baudios utilizando la función **begin()**. Por último, se realiza un **flush()** para la eliminación de cualquier valor residual dentro del buffer.

```

String dato;
const char fin = '>';
char paquete, c2, anterior;
int c1, c3;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial) {
    ; // Espera a que termine de configurarse el puerto serie
  }
  delay(500);
  Serial2.begin(9600);
  while (!Serial2) {
    ; // Espera a que termine de configurarse el puerto serie
  }
  delay(500);
  Serial3.begin(9600);
  while (!Serial3) {
    ; // Espera a que termine de configurarse el puerto serie
  }
  delay(500);
  Serial.flush();
  Serial2.flush();
  Serial3.flush();
  Serial3.write("#C");
  Serial3.write(005);
  Serial3.write(010);
  delay(100);
  Serial3.flush();
  Serial3.write("#A");
  Serial3.write(126);
  Serial3.write(126);
  Serial3.write(126);
  Serial3.write(125);
  Serial3.write(125);
  Serial3.write(125);
  delay(100);
  Serial3.flush();
  Serial3.write("#H");
  Serial3.write(000);
  delay(100);
  Serial3.flush();
  Serial3.write("#B");
  Serial3.write(000);
  delay(100);
  Serial3.flush();
  Serial3.write("#Q");
  delay(100);
}

```

Configuraciones para la recepción de datos del receptor VLC

Figura 34. Configuración del Arduino Mega para receptor VLC. Variables y setup().

El siguiente paso es podreecer a la espera de recepción de datos con la función **available()**. En cuanto haya un dato en el puerto serie, la función devolverá un valor mayor que cero. En caso contrario, devolverá un **-1**. El dato se almacena en la variable **c2**.

```
void loop() {  
  // put your main code here, to run repeatedly:  
  delay(10);  
  if (Serial2.available() > 0) {  
    c2 = Serial2.read();  
  }  
  if ((c2 == 'r') || (c2 == 'g') || (c2 == 'b')) {  
    if (c2 != anterior) {  
      anterior = c2;  
      Serial3.write('<');  
      Serial3.write(c2);  
      Serial3.write('>');  
      //anterior = c2;  
    }  
  } else {  
  }  
  if (Serial.available() > 0) {  
    c1 = Serial.read();  
    Serial3.write(c1);  
  }  
  if (Serial3.available() > 0) {  
    c3 = Serial3.read();  
    Serial.println(c3);  
  }  
}
```

Captura de datos del receptor VLC

Figura 35. Configuración del Arduino Mega para receptor VLC. Loop().

El esquemático del transmisor y receptor VLC están expuestos en las siguientes figuras.

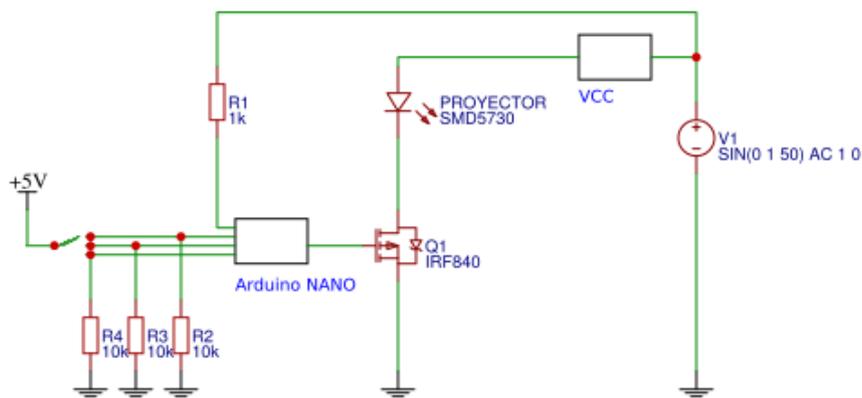


Figura 36. Esquemático de la lámpara VLC.

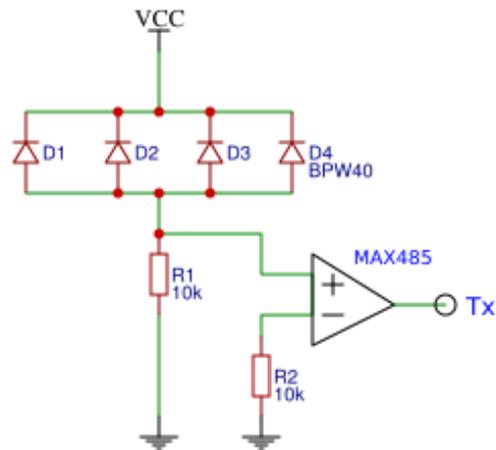


Figura 37. Esquemático del receptor VLC.

### Caracterización del enlace

Las características del enlace son las siguientes:

- Modulación: OOK (On-Off Keying).
- Configuración del enlace: LOS.
- Velocidad de transmisión: 9600 baudios.
- Alcance efectivo: 3 metros.

## 2. Enlace LoRa

El enlace LoRa consiste en dos módulos GAMMA-868 configurados como “modem serial” conectados en dos Arduinos diferentes. El transmisor se encuentra conectado al Arduino Mega y el receptor en el Arduino Uno. En la siguiente ilustración se encuentra la configuración empleada en el transmisor.

```

String dato;
const char fin = '>';
char paquete, c2, anterior;
int c1, c3;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial) {
    ; // Espera a que termine de configurarse el puerto serie
  }
  delay(500);
  Serial2.begin(9600);
  while (!Serial2) {
    ; // Espera a que termine de configurarse el puerto serie
  }
  delay(500);
  Serial3.begin(9600);
  while (!Serial3) {
    ; // Espera a que termine de configurarse el puerto serie
  }
  delay(500);
  Serial3.flush();
  Serial3.write("#C");
  Serial3.write(005);
  Serial3.write(010);
  delay(100);
  Serial3.flush();
  Serial3.write("#A");
  Serial3.write(126);
  Serial3.write(126);
  Serial3.write(126);
  Serial3.write(125);
  Serial3.write(125);
  Serial3.write(125);
  delay(100);
  Serial3.flush();
  Serial3.write("#H");
  Serial3.write(000);
  delay(100);
  Serial3.flush();
  Serial3.write("#B");
  Serial3.write(000);
  delay(100);
  Serial3.flush();
  Serial3.write("#Q");
  delay(100);
}

```

Inicio de la comunicación serie a 9600 baudios.

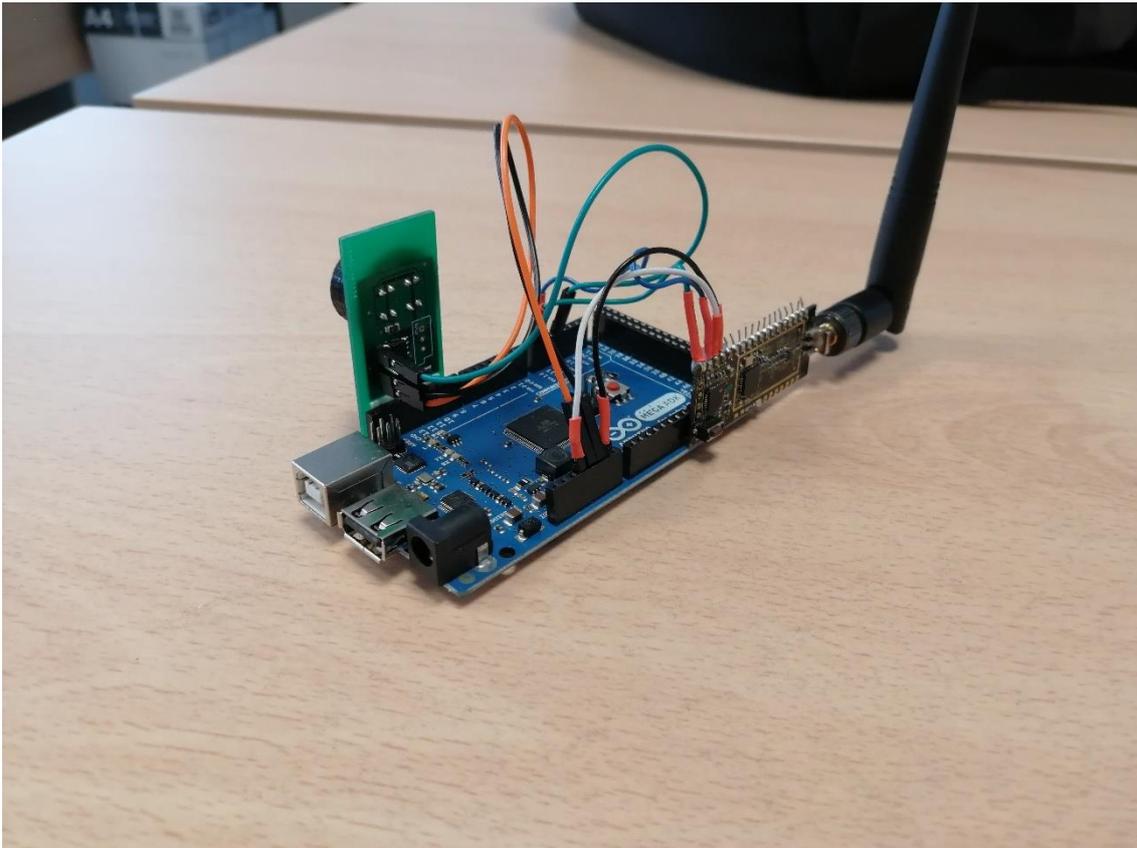
LoRa mode: 5  
 Potencia de salida: 10  
 Dirección local: 7E7E7E  
 Dirección del destinatario: 7D7D7D  
 Sin handshaking  
 Velocidad de transmisión: 9600 baudios  
 Reinicio del módulo

Figura 38. Configuración del módulo LoRa. Transmisor (Arduino Mega).

La configuración utilizada en el transmisor LoRa es:

- Para el campo **#C**, LoRa mode **5** y potencia de salida **10**.
- Para el campo **#A**, dirección local **7E7E7E** (valores en hexadecimal), dirección de destino **7D7D7D**.
- Para el campo **#H**, **000**, handshaking inhabilitado.
- **#B 000**, 9600 baudios de velocidad de transmisión.

**#Q** reinicia el módulo quedando así los datos guardados.



*Figura 39. Arduino Mega. Transmisor LoRa.*

En la siguiente figura se encuentra el código donde primero se captura el dato recibido por el receptor VLC observando si ha variado respecto al anterior. Cuando se produzca un cambio de código, se encapsulará dentro de < > y se enviará por el módulo LoRa.

```
void loop() {
  // put your main code here, to run repeatedly:
  delay(10);
  if (Serial2.available() > 0) {
    c2 = Serial2.read();
  }
  if ((c2 == 'r') || (c2 == 'g') || (c2 == 'b')) {
    if (c2 != anterior) {
      anterior = c2;
      Serial3.write('<');
      Serial3.write(c2);
      Serial3.write('>');
      //anterior = c2;
    }
  }
  else{
  }
  if (Serial.available() > 0) {
    c1 = Serial.read();
    Serial3.write(c1);
  }
  if (Serial3.available() > 0) {
    c3 = Serial3.read();
    Serial.println(c3);
  }
}
```

1°- Lectura y almacenaje del dato recibido por VLC

2°- Se vigila si se ha cambiado el código. En caso afirmativo, se envía al módulo LoRa para su transmisión.

Figura 40. Código de Arduino Mega. Envío de datos por LoRa.

En el receptor, se realizan los mismos pasos de configuración, pero con las direcciones intercambiadas. En la siguiente figura se encuentra el código donde primero se captura el dato recibido por el módulo LoRa.

```

#include <SoftwareSerial.h>
String dato;
const char fin = '>';
char paquete;
SoftwareSerial mySerial(10, 11); // RX, TX
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  while (!Serial) {
    ; // Espera a que termine de configurarse el puerto serie.
  }
  mySerial.begin(9600);
  delay(100);
  mySerial.flush();
  mySerial.print("#C");
  mySerial.write(005);
  mySerial.write(010);
  delay(100);
  mySerial.flush();
  mySerial.write("#A");
  mySerial.write(125);
  mySerial.write(125);
  mySerial.write(126);
  mySerial.write(126);
  mySerial.write(126);
  delay(100);
  mySerial.flush();
  mySerial.write("#H");
  mySerial.write((byte)0x00);
  delay(100);
  mySerial.flush();
  mySerial.write("#B");
  mySerial.write((byte)0x00);
  delay(100);
  mySerial.flush();
  mySerial.write("#Q");
  delay(100);
}

```

Figura 41. Configuración del módulo LoRa. Receptor (Arduino Uno).

La configuración utilizada en el receptor LoRa es:

- Para el campo **#C**, LoRa mode **5** y potencia de salida **10**.
- Para el campo **#A**, dirección local **7D7D7D** (valores en hexadecimal), dirección de destino **7E7E7E**.
- Para el campo **#H**, **000**, handshaking inhabilitado.
- **#B 000**, 9600 baudios de velocidad de transmisión.

#Q reinicia el módulo quedando así los datos guardados.



*Figura 42.Arduino Uno. Receptor LoRa.*

Al recibir el dato entregado por el transmisor LoRa, se desencapsula para luego enviarlo por el puerto serie donde será recibido por el programa que lo enviará a la API para su guardado en la base de datos.

```

void loop() {
  // put your main code here, to run repeatedly:
  if (mySerial.available() > 0) {
    paquete = mySerial.read();
    if (paquete == '<') { // start of packet
      dato = paquete;
      while (paquete != '>') { // until end of packet
        if (mySerial.available() > 0) {
          paquete = mySerial.read();
          dato += paquete;
        }
      }
      Serial.println(dato);
    }
  }
}

```

- 1°- Se recoge el dato recibido
- 2°- Guarda el dato que se encuentra encapsulado
- 3°- Se envía por el puerto serie

Figura 43. Código de Arduino Uno. Recepción de datos por LoRa.

### 3. Aplicación web

La aplicación web es la encargada de enviar los datos recibidos a la base de datos para posteriormente plasmarlos en una página web realizada en Angular. El esquema sería el siguiente:

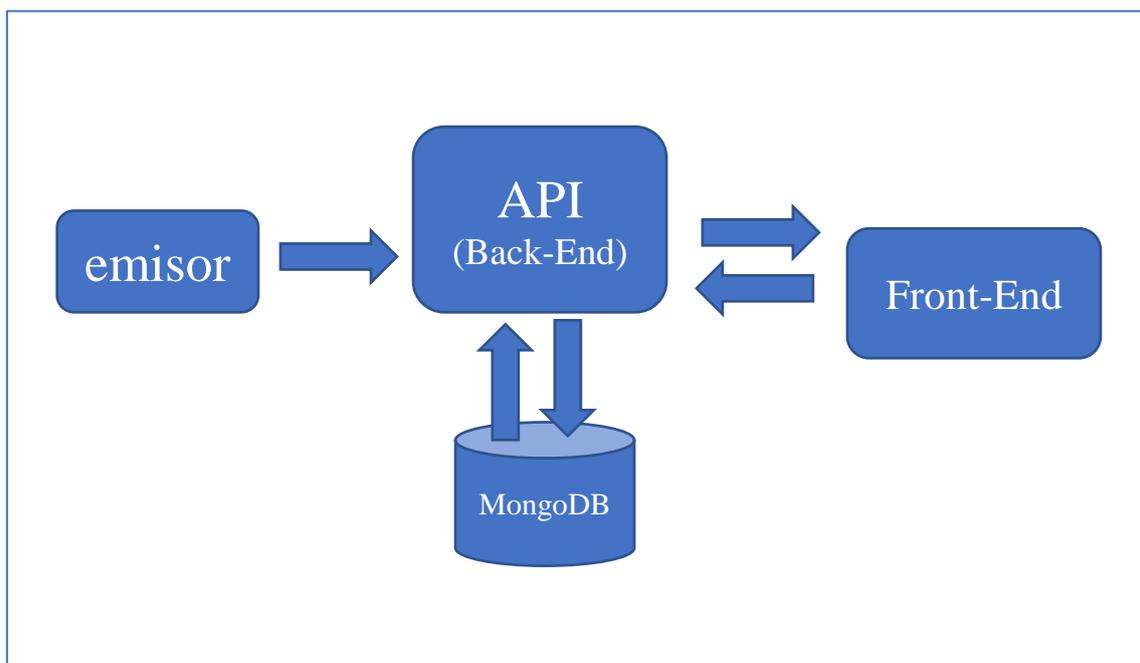
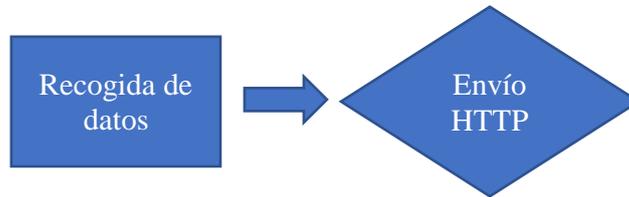


Figura 44. Esquema de la aplicación web.

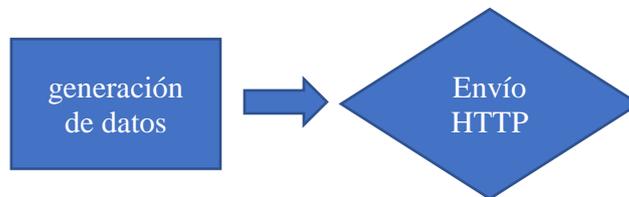
## Emisor

Para la obtención de los datos que envía el Arduino Uno, se ha desarrollado un programa llamado **emisor** en Python. Además, realiza el envío de los datos por petición HTTP para almacenarlos en la base de datos.



*Figura 45. Esquema de "emisor".*

Hay un segundo programa llamado **emisorRand**. Crea números aleatorios comprendidos entre 20 y 25 simulando un sensor de temperatura con un módulo LoRa.



*Figura 46. Esquema de "emisorRand".*

```
emisor.py ●
1  import serial
2  import requests
3  import serial.tools.list_ports    # import serial module
4
5  ser = serial.Serial()
6  ser.baudrate = 9600
7
8  ports = serial.tools.list_ports.comports()
9
10 puerto = input ("Seleccione el puerto: ")
11 print ("Ha seleccionado: "+puerto)
12 ser.port = puerto
13 print([port.name for port in ports])
14 uId = 'loraVLC'
15
16 url = 'http://localhost:3977/apiD/saveData'
17
18 ser.open()
19
20
21 while 1:
22
23     if ser.in_waiting > 0:
24         dato = ''
25         for x in range(5):
26             dato = dato + ser.read()
27
28         print('Dato enviado: ' +dato)
29         payload = {'data': dato, 'uId': uId}
30         r = requests.post(url, payload)
31
```

Figura 47. Código de emisor.py.

```
emisorRand.py x
1  import requests
2  import random
3  from random import randint
4  import time
5
6  uId = 'loraTEMP'
7  url = 'http://127.0.0.1:3977/apiD/saveData'
8  urltest = 'http://httpbin.org/post'
9  while 1:
10     entero = random.randint(20, 25)
11     decimal = round(random.random(), 2)
12     dato = entero + decimal
13     payload = {'data': dato, 'uId': uId}
14     r = requests.post(url, payload)
15     print ("Dato enviado: ",dato)
16     time.sleep(5)
17
```

Figura 48. Código de emisorRand.py.

### Back-End (API)

El Back-End es el encargado gestionar la base de datos y es el punto de acceso a la base de datos para el cliente. Está basada en Node.JS. Aquí se presenta un diagrama UML sobre los modelos que han sido desarrollados.

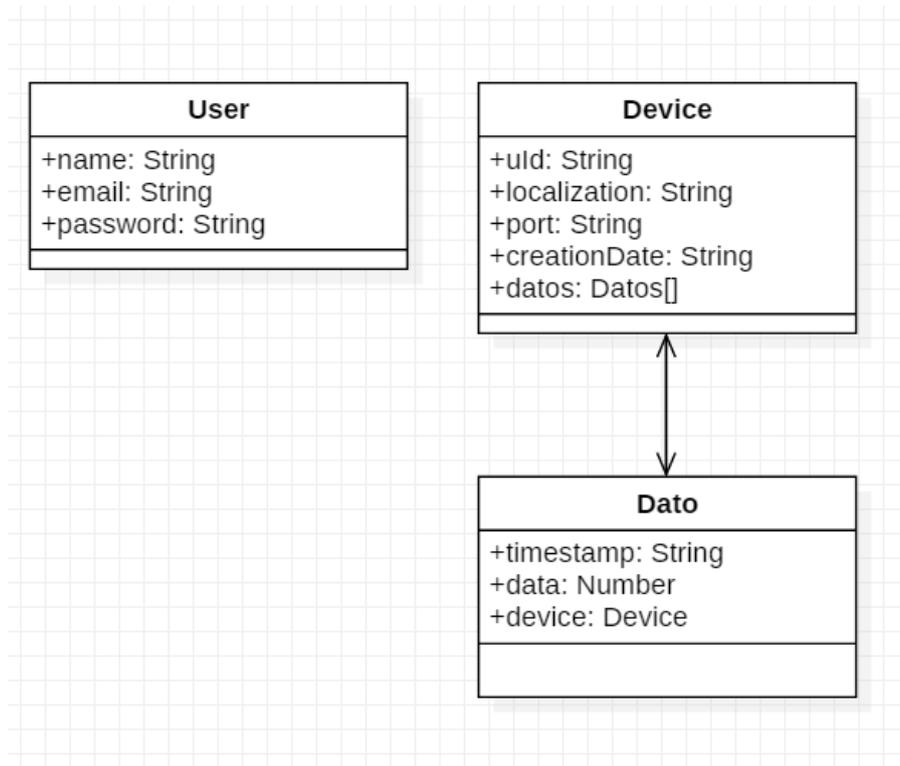


Figura 49. Diagrama UML

**User:** Será el usuario que se registrará en la página web. Contiene información como su nombre, email y contraseña de la cuenta.

**Device:** Es la identificación de los dispositivos dados de alta. Contiene un identificado **uId**, **localization** siendo el lugar donde fue colocado, **port** el puerto serie por el que se encuentra conectado, **creationDate** que contiene el momento en el que fue dado de alto al sistema y tiene asociados un array de objetos tipo **Dato**.

**Dato:** Será donde se almacenarán los datos. Poseen un **timestamp** que indicará el momento de la captura del dato, **data** que es el dato en sí y **device** el dispositivo al que pertenece ese dato.

La estructuración de la API se muestra en la siguiente figura:

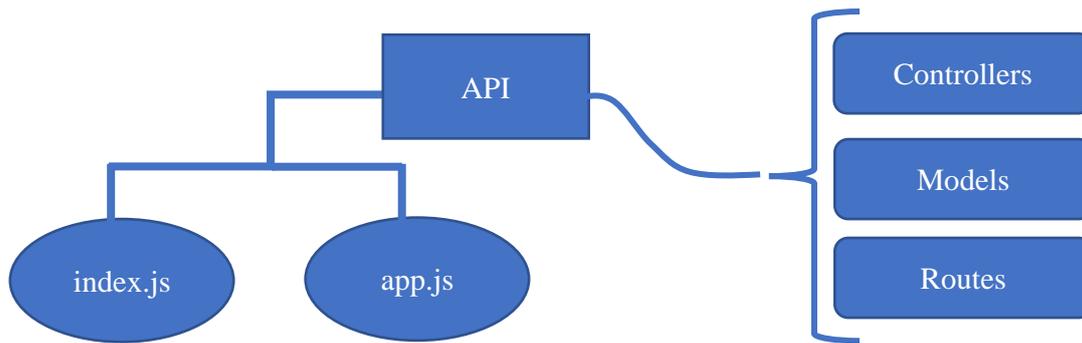


Figura 50. Esquema de la API.

- **index.js.** Es el archivo raíz. Se encarga de la conexión a MongoDB y el inicio del servidor. Utiliza el módulo **Mongoose**.
- **app.js.** Es donde se inicia la API, configuración de cabeceras y crea las rutas de los servicios que ofrece la aplicación. Realizada con el módulo **Express.js**.
- **Controllers.** Aquí se encuentran todas las funciones para la creación, modificación, envío, guardado y eliminación de los datos que se encuentran en MongoDB. Están organizados en dos partes: **user.js** donde están todas las funciones destinadas al usuario, y **device.js** destinadas para los dispositivos.
- **Models.** Es donde se encuentran los modelos de los esquemas para la base de datos. Hay tres tipos: **user.js** donde indica todos los atributos de las cuentas de usuario, **device.js** de los dispositivos dados de alta o por dar, y **dato.js** el modelo para los datos que se van a recoger de la interfaz.
- **Routes.** Es donde se programan todas las rutas para el acceso de los servicios ofrecidos hacia los clientes.

## Front-End

El Front-End se encarga de la interacción con el usuario. Esta realizada con Angular. Aquí es donde se representarán los datos obtenidos, el registro, modificación y eliminación de usuarios y dispositivos.

La estructuración del Front-End se muestra en la siguiente figura:

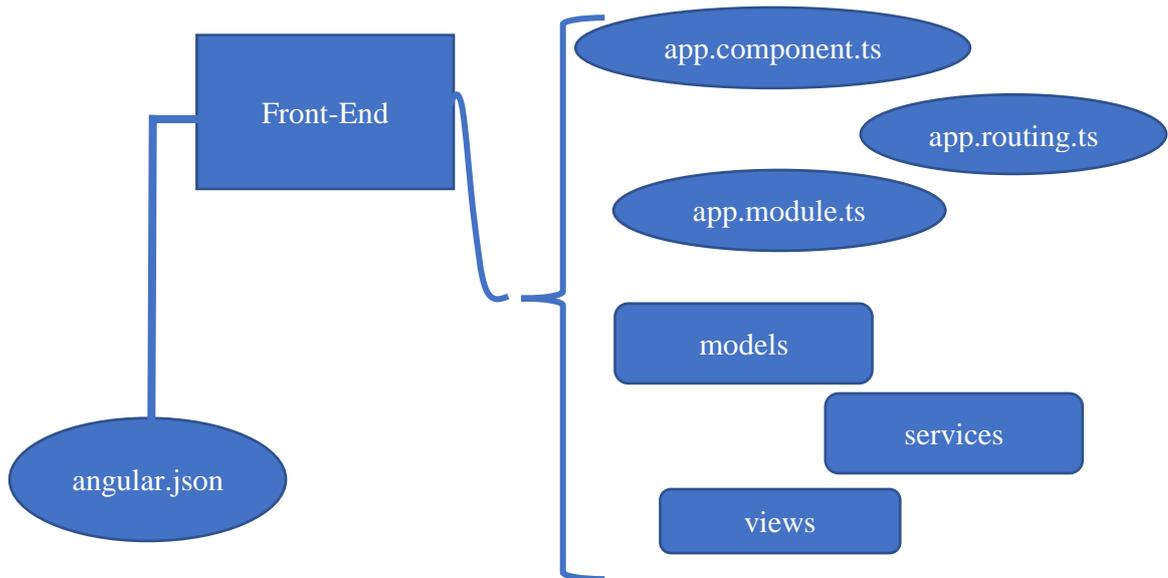


Figura 51. Esquema del Front-End.

- **angular.json.** Es donde queda recogida la configuración y carga de todas las partes implicadas para el correcto funcionamiento de la aplicación.
- **app.component.ts.** Es el componente raíz puesto que Angular se compone como un árbol de componentes.
- **app.module.ts.** Es donde se cargan todos los módulos que se quieren utilizar. Los módulos implicados en este proyecto son:
  - **BrowserModule.** Exporta la infraestructura requerida para todas las aplicaciones angulares.
  - **FormsModule.** Exporta los proveedores y las directivas requeridas para formularios controlados por plantillas.
  - **ReactiveFormsModule.** Exporta la infraestructura y las directivas necesarias para formularios reactivos.
  - **AppRoutingModule.** Importa al **RouterModule** de Angular toda la lógica necesaria para enrutar en el navegador y permite la definición de rutas en el array **Routes[]**.
  - **HttpClientModule.** Configura el inyector de dependencias para **HttpClient** con los servicios de soporte para XSRF (Cross-Site Request Forgery) previniendo ataques basados en la falsificación de solicitudes.
  - **ChartsModule.** Carga el módulo de terceros llamado Chart.js que sirve para la creación y representación de gráficos en la web.
- **app.routing.ts.** Es donde se configuran las rutas de los componentes que contiene la aplicación. Dichos componentes son los que se encuentran en **views**.

- **models.** Aquí se diseñan los modelos de los datos que más tarde se obtendrán de las llamadas a la API. Estos son los mismo que los de la API: **user.ts**, **device.ts** y **dato.ts**.
- **services.** Es donde se encuentra el servicio encargado de las peticiones al Back-End.
- **views.** En esta carpeta se encuentran todas las vistas de la aplicación web con las que interactuarán los usuarios. Las vistas realizadas son:
  - **Login.** Es donde el usuario se logeará mediante el email y su contraseña.
  - **Register.** Un nuevo usuario puede registrarse rellorando el formulario que aquí se encuentra.
  - **Navigator.** Es una barra de navegación por donde el usuario puede navegar.
  - **Line-chart.** Es donde se mostrarán los gráficos con los datos obtenidos.

## 4. Implementación final

Una vez terminado el desarrollo de todas las partes implicadas, se realizan las diferentes verificaciones para dar por efectuada los objetivos de este proyecto.

Para la verificación de la interfaz VLC-LoRa se basó en la recepción de los datos por el puerto serie correspondiente. En la figura 52 se puede observar los datos recibidos por el puerto serie del ordenador.



Figura 52. Recepción de los datos por el puerto serie.

Los datos deben ser enviados mediante el programa emisor.py y almacenados en la base de datos. Para su verificación, se realiza una petición HTTP a la API para que entregue los datos que posee el dispositivo que ha sido llamado como *loraVLC*. En la figura 53 puede observarse el objeto JSON que se recibe posee los datos que han sido recibidos por la interfaz VLC-LoRa.

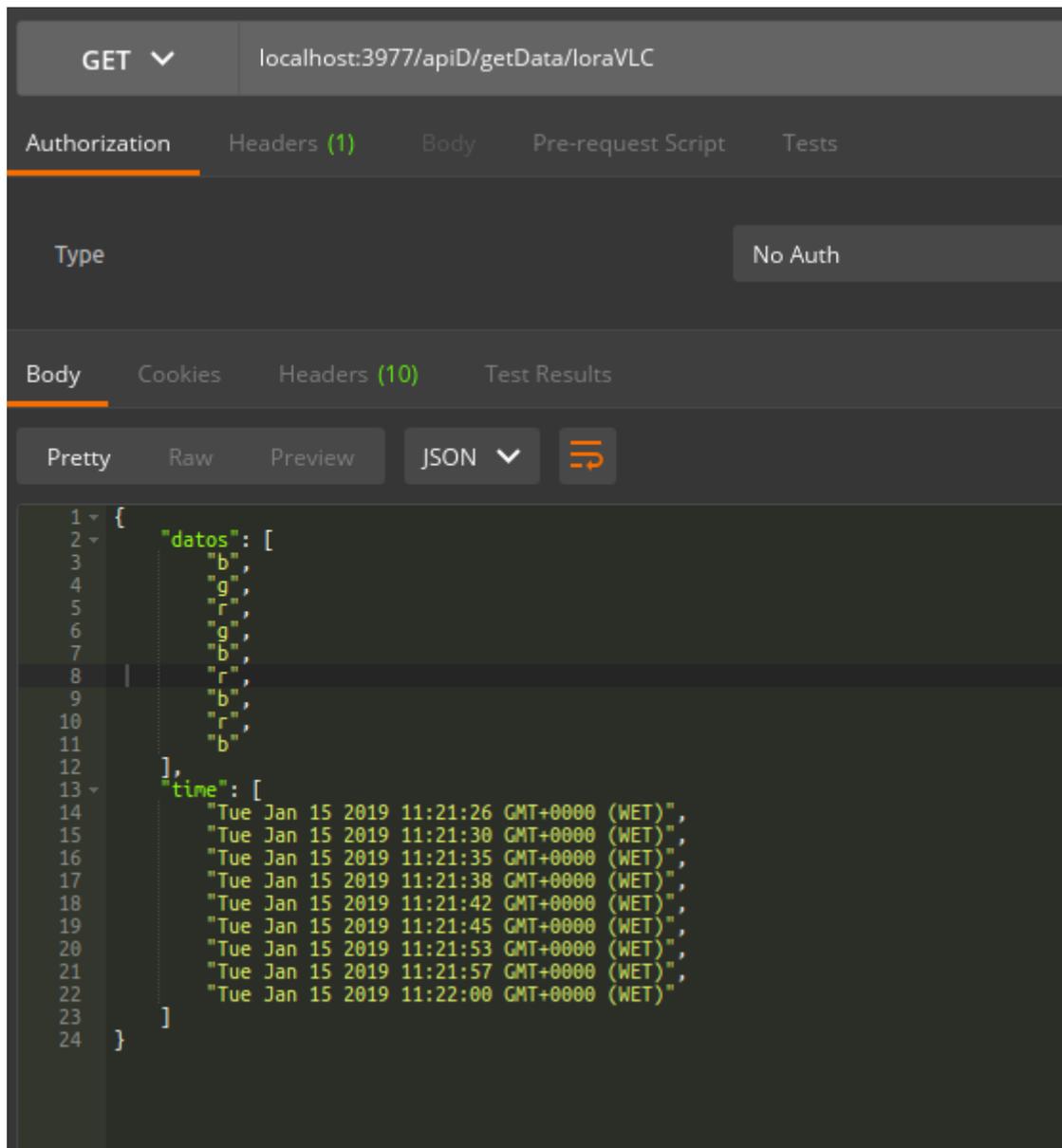
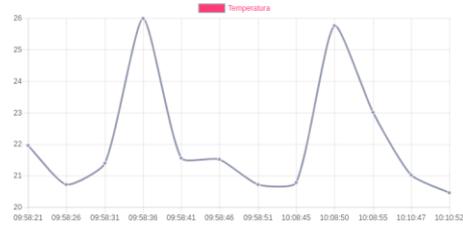


Figura 53. Datos de la base de datos.

A continuación, la validación de la aplicación web se basará en la recepción y exposición de los datos que se encuentran en la base de datos. Con este último paso, se da por validado todo el sistema realizado y el cumplimiento de las funciones de cada una de las partes. En la figura 55 se observa la muestra de los datos por parte de la página web.

# Interfaz VLC - LoRa con plataforma de gestión

Autor: Angel Caamaño Lourido

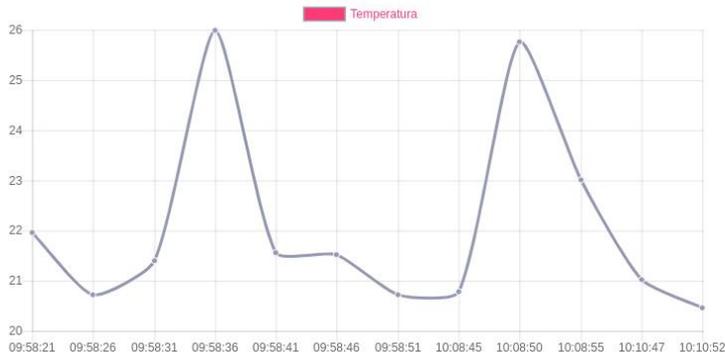


## Estado del enlace VLC

Se encuentra con el código: b

Los 3 anteriores: r, b, r

Figura 54. Página web.



Estado del enlace VLC

Se encuentra con el código: b

Los 3 anteriores: r, b ,r

Figura 55. Los datos mostrados.

## Capítulo VI

### Conclusiones

En vista de los resultados obtenidos en cada una de las diferentes pruebas realizadas se puede afirmar que con el presente Trabajo de Fin de Grado se ha logrado el objetivo de diseñar e implementar una interfaz VLC-LoRa con plataforma de gestión.

Se han conseguido así todos los objetivos propuestos, y el sistema cumple con los criterios y requisitos de diseño establecidos.

Estas son las conclusiones a las que se ha llegado:

- Se ha realizado un estudio del arte sobre la tecnología VLC para conocer y aprender y diseñar los sistemas VLC. Dentro de este estudio, se descubre que la tecnología posee mucho potencial ya que utilizado una banda de frecuencia poco explotada.
- Se ha realizado un último estudio del arte sobre la tecnología LoRa para conocer y aprender sus características. En dicho estudio, se ha conocido los diferentes campos que la componen y se observa la continua expansión en la que se encuentra presente.
- Se han conseguido las habilidades necesarias para la utilización y manipulación de todo el hardware y software empleado.
- Se ha diseñado e implementado un receptor VLC para adaptarse a la velocidad de transmisión de la lámpara VLC con un diseño que permita una conexión directa con el Arduino por el que se alimenta.

- Se ha implementado dos programas basados en Python para el envío de la información puerto serie de la interfaz VLC-LoRa y la simulación de un sensor de temperatura.
- Se ha implementado un servidor web con un back-End y un Front-End diferenciados. Back-End mediante Express.JS para la gestión de la base de datos utilizada. Front-End basada en Angular para la exposición de los datos.
- La tecnología VLC es perfecta para entornos interiores, pero, para la tecnología LoRa gracias a su gran alcance, podría emplearse para reforzar la debilidad de VLC, utilizarla para transmitir entre diferentes edificios.

# Capítulo VII

## Bibliografía

- [1] Semtech: What is LoRa®? <https://www.semtech.com/lora/what-is-lora> (2018). [Accessed 12/13 2018].
- [2] HGN: LED lifetime expectancy. LED Lifetime Expectancy, [http://www.eyervis.de/files/tn\\_led\\_cube\\_technical\\_note\\_led\\_lifetime.pdf](http://www.eyervis.de/files/tn_led_cube_technical_note_led_lifetime.pdf) (2012). [Accessed: 01-Jan-2019]
- [3] EUROPA PRESS: Nuevo récord de transmisión de datos por láser., <https://www.europapress.es/ciencia/laboratorio/noticia-nuevo-record-transmision-datos-laser-20161104110524.html> (04/11/2016). [Accessed: 01-Jan-2019]
- [4] Adarsh Verma: 2 gbps: New speed record set for data transmission using LED light., <http://fossbytes.com/data-transfer-using-led-light-record-2-gbps/> (August 15, 2016). [Accessed: 01-Jan-2019]
- [5] I. Standard and I. C. Society, “IEEE Standard for Local and metropolitan area networks — Part 15 . 7 : Short-Range Wireless Optical Communication Using Visible Light IEEE Computer Society Sponsored by the,” *Communication*, no. September, 2011, pp. 1-306. [Accessed: 01-Jan-2019]
- [6] L. Grobe, A. Paraskevopoulos, J. Hilt, D. Schulz, F. Lassak, F. Hartlieb, C. Kottke, V. Jungnickel, and K. D. Langer, “High-speed visible light communication systems,” *IEEE Commun. Mag.*, vol. 51, no. 12, pp. 60–66, 2013. [Accessed: 01-Jan-2019]
- [7] FAWOO TECHNOLOGY, “Tecnología LED,” *2017*, no. 2, p. 1, 2017. [Accessed: 01-Jan-2019]
- [8] Elemental LED, “ARE LEDS DIMMABLE?” [Online]. Available: <https://www.elementaled.com/are-leds-dimmable/>. [Accessed: 01-Jan-2019].
- [9] Y. Matsuda, Y. Kozawa, and Y. Umeda, “Cyclic mapping method for digital color shift keying with RGB-LED array,” *IEEE Radio Wirel. Symp. RWS*, no. 1, pp. 153–155, 2017. [Accessed: 02-Jan-2019]

- [10] I. Standard and I. C. Society, “IEEE Standard for Local and metropolitan area networks — Part 15 . 7 : Short-Range Wireless Optical Communication Using Visible Light IEEE Computer Society Sponsored by the,” *Communication*, no. September, 2011. [Accessed: 01-Jan-2019]
- [11] L. Ming-Li, Y. Hong-Yi, W. Guo-Feng, and W. Chao, “Multi-LED optical OFDM modulation for visible light communication,” *2017 9th IEEE Int. Conf. Commun. Softw. Networks, ICCSN 2017*, vol. 2017–January, pp. 635–639, 2017. [Accessed: 02-Jan-2019]
- [12] F. Swine, “Que es el ‘spectrum crunch,’” *26 octubre*, 2013. [Online]. Available: [https://www.bbc.com/mundo/noticias/2013/10/131025\\_tecnologia\\_banda\\_ancha\\_saturacion\\_espectro\\_amv\\_finde](https://www.bbc.com/mundo/noticias/2013/10/131025_tecnologia_banda_ancha_saturacion_espectro_amv_finde). [Accessed: 02-Jan-2019]
- [13] A. O. Cristina Calzadilla, “VLC o comunicaciones por luz visible: ¿antesala del Li-Fi?,” 7 de noviembre, 2013. [Online]. Available: <https://blogthinkbig.com/vlc-comunicaciones-por-luz-visible>. [Accessed: 03-Jan-2019]
- [14] ULL, “Luces que comunican,” miércoles 27 de junio, 2018. [Online]. Available: <https://www.ull.es/portal/noticias/2018/luces-que-comunican-vlc/>. [Accessed: 03-Jan-2019]
- [15] “European Training Network on Visible light based Interoperability and Networking,” 2017. [Online]. Available: <https://cordis.europa.eu/project/rcn/211543/factsheet/en>. [Accessed: 03-Jan-2019]
- [16] “Visible Light Communication (VLC) application for mobile devices as control access, validation, master key or secure payment solution.” [Online]. Available: <https://cordis.europa.eu/project/rcn/197160/factsheet/en>. [Accessed: 03-Jan-2019]
- [17] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M. D. Cano, and A. F. Skarmeta, “Performance evaluation of lora considering scenario conditions,” *Sensors (Switzerland)*, vol. 18, no. 3, 2018. [Accessed: 10-Jan-2019]
- [18] Sabas, “¿Qué es LoRa y LoRaWAN?,” 25 de septiembre, 2017. [Online]. Available: <https://medium.com/beelan/haciendo-iot-con-lora-capítulo-1-qué-es-lora-y-lorawan-8c08d44208e8>. [Accessed: 10-Jan-2019]
- [19] SigFox, “How many devices do you want to connect ?” [Online]. Available: <https://buy.sigfox.com/buy/offers/ES>. [Accessed: 10-Jan-2019]

- [20] SigFox, “Buy SigFox Connectivity.” [Online]. Available: <https://buy.sigfox.com/>. [Accessed: 10-Jan-2019]
- [21] L. Alliance, “LoRa Alliance™ Enhances LoRaWAN™ Protocol with New Specifications to Support Firmware Updates Over the Air,” October 25, 2018. [Online]. Available: <https://lora-alliance.org/in-the-news/lora-alliancetm-enhances-lorawantm-protocol-new-specifications-support-firmware-updates>. [Accessed: 10-Jan-2019]
- [22] “¿Qué es MongoDB?,” 2019. [Online]. Available: <https://www.mongodb.com/es/what-is-mongodb>. [Accessed: 10-Jan-2019]
- [23] M. Á. Álvarez and A. Morales Gámez, “Características destacables de NodeJS,” 18 de enero, 2013. [Online]. Available: <https://desarrolloweb.com/articulos/caracteristicas-nodejs.html>. [Accessed: 10-Jan-2019]
- [24] V. Robles, “¿Qué es Angular y para qué sirve?,” 28 de Abril, 2018. [Online]. Available: <https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/>. [Accessed: 10-Jan-2019]
- [25] NGuerrero, “¿Que es Angular CLI?,” 18 de junio, 2018. [Online]. Available: <http://programaenlinea.net/angular-cli-sirve/>. [Accessed: 10-Jan-2019]

## **Parte II**

# **Pliego de Condiciones**



# Pliego de condiciones

A continuación, se detallan las características de los recursos hardware y software empleados durante todo el proyecto.

## 1. Equipo hardware

Los recursos hardware utilizados se muestran en la tabla.

Tabla 2. Equipo hardware.

Equipo	Características
Portátil	12 GB RAM, Intel Core i7-4750HQ
Arduino UNO R3	Microcontrolador ATmega328P, memoria Flash 32 KB, SRAM 2KB
Arduino Mega 2560 R3	Microcontrolador ATmega2560, memoria Flash 256 KB, SRAM 8 KB
GAMMA-868 (x2)	alimentación 1.8-3.6V, datos serie 38.4 KB, frecuencia de trabajo 869.5 MHz, sensibilidad +20dBm TX/ -137 dBm RX
Lámpara VLC	Lámpara LED Phillips. Ancho de banda 1 Mbps. Velocidad de transmisión 9600 baudios.
Receptor VLC	4 fototransistores BPW40, ángulo de visión 40°, rango espectral 520 – 950 nm. Velocidad de transmisión 9600 baudios.

## 2. Equipo software

Los recursos software utilizados se muestran en la siguiente tabla.

Tabla 3. Equipo software.

Software	Versión	Fabricante	Descripción
Windows	1809	Microsoft	Sistema Operativo
Node.JS	11.6.0	Node.js Foundation	Entorno de ejecución para JavaScript
MongoDB	4.02	MongoDB Inc	Base de datos NoSQL
Angular CLI	7.0	Google	Framework para JavaScript para aplicaciones web
Python	3.7.2	Python Soft. Foundation	Lenguaje de programación interpretad

Arduino IDE	1.8.8	Arduino Soft.	Entorno de desarrollo para placas Arduino
VS Code	1.30	Microsoft	Entorno de desarrollo integrado
Postman	6.7.1	Postdot Technologies Pvt. Ltd.	Herramienta de desarrollo web para APIs

# **Parte III**

## **Presupuesto**



# Presupuesto

## Introducción

En este capítulo se estimarán los gastos generados por el trabajo fin de grado presentado en esta memoria. El presupuesto presentado se divide en las siguientes partes:

- Tarifa de honorarios por tiempo empleado.
- Amortización de los equipos empleados.
  - Amortización del material hardware.
  - Amortización del material software.
- Redacción de la documentación.
- Derechos de visado del COITT (Colegio Oficial de Ingenieros Técnicos de Telecomunicación).
- Gastos de tramitación y envío.

## Tarifa de honorarios por tiempo empleado

Este concepto contabiliza los gastos correspondientes a la mano de obra. Para realizar este cálculo, se propone la siguiente fórmula:

$$H = C_t \cdot 74,88 \cdot H_n + C_t \cdot 96,72 \cdot H_e$$

Siendo:

- **H**: Honorarios totales por el tiempo dedicado.
- **H<sub>n</sub>**: Número de horas normales trabajadas dentro de la jornada laboral.
- **C<sub>t</sub>**: Factor de corrección que depende del número de horas trabajadas.
- **H<sub>e</sub>**: Número de horas especiales trabajadas.

Se han invertido un total de 300 horas en la realización del presente Trabajo Fin de Grado. Todas ellas se han realizado dentro del horario normal, por lo que el número de horas especiales es cero. De acuerdo con lo establecido por el COITT, el factor de corrección  $C_t$  que se aplica para 300 horas trabajadas es de 0,60, como se aprecia en la siguiente tabla.

Tabla 4. Coeficientes reductores para trabajo tarifado según el COITT.

Horas	Factor de corrección
Hasta 36	1

<b>Exceso de 36 hasta 72</b>	0,90
<b>Exceso de 72 hasta 108</b>	0,80
<b>Exceso de 108 hasta 144</b>	0,70
<b>Exceso de 144 hasta 180</b>	0,65
<b>Exceso de 180 hasta 360</b>	0,60

Entonces, haciendo uso de la anterior ecuación:

$$H = 0,60 \cdot 74,88 \cdot 300 + 0,60 \cdot 96,72 \cdot 0 = 13.478,40 \text{ €}$$

Por lo tanto, el trabajo tarifado por tiempo empleado asciende a la cantidad de **TRECE MIL CUATROCIENTOS SETENTA Y OCHO EUROS CON CUARENTA CÉNTIMOS**.

## **Amortización de los equipos empleados**

Dentro de este concepto se considera tanto la amortización del hardware como del software empleado en la realización del trabajo presentado. De este modo, se estipula el coste de amortización para un período de 3 años, utilizando un sistema de amortización lineal o constante. En este sistema, se supone que el inmovilizado material se deprecia de forma constante a lo largo de su vida útil. La cuota de amortización anual se calcula haciendo uso de la siguiente formula:

$$C = \frac{V_{ad} - V}{N}$$

Donde:

- **C**: cuota de amortización anual
- **V<sub>ad</sub>**: valor de la adquisición
- **V**: valor residual.
- **N**: número de años de vida útil de la adquisición.

Siendo el valor residual el valor teórico que se supone tendrá el elemento en cuestión después de su vida útil, teniendo en cuenta los índices de depreciación actual. En el caso del hardware y del software son 3 años (al 33 % de depreciación máximo por año).

### **Amortización del material hardware**

Debido a que el trabajo se ha elaborado en un periodo inferior a 3 años, que es el periodo en que se calcula la amortización del material hardware, se realizará una amortización equiparable al período de duración de este. Según esto, se obtienen los gastos expuestos en la tabla .

Tabla 5. Precios y costes de amortización hardware.

Descripción	Unidades	Valor de adquisición	Tiempo de uso	Coste anual	Total
ASUS	1	845 €	6 meses	169 €	84,5 €
Arduino UNO R3	1	30 €	6 meses	30 €	30 €
Arduino Mega 2560 R3	1	36 €	6 meses	36 €	36 €
GAMMA-868	2	72,56 €	6 meses	72,56 €	72,56 €
Lámpara VLC	1	47 €	6 meses	47 €	47 €
Receptor VLC	1	5 €	6 meses	5 €	5 €
<b>Total</b>					<b>275,06 €</b>

Por lo tanto, el coste total de hardware asciende a la cantidad de **DOSCIENTOS SETENTA Y CINCO EUROS CON SEIS CÉNTIMOS**.

### Amortización del material software

El coste total del software utilizado es la suma de las herramientas software empleadas para la realización del trabajo que se encuentra en la tabla 6. De esta manera se describe cada uno de las herramientas utilizadas y el coste supuesto para su utilización usando la fórmula de amortización anteriormente citada.

Tabla 6. Precios y costes de amortización del software.

Descripción	Valor de adquisición	Tiempo de uso	Coste anual	Total
Windows 10	0	6 meses	0	0
Visual Studio Code	0	6 meses	0	0

Node.JS	0	6 meses		0
MongoDB	0	6 meses	0	0
Angular	0	6 meses	0	0
Python	0	6 meses	0	0
Arduino IDE	0	6 meses	0	0
Postman	0	6 meses	0	0
<b>Total</b>				0 €

El software Windows 10 es de licencia privada y para su uso es necesario comprarlo. En este proyecto se ha utilizado un Windows 10 con una licencia contratada por la Universidad Las Palmas de Gran Canaria puesto que la ofrece para sus alumnos. En cuanto al resto del software, es de licencia libre y su uso es gratuito.

Por tanto, el coste total de software asciende a **CERO EUROS**.

## Redacción de la documentación

Para calcular el valor monetario de la redacción del trabajo se aplicará la fórmula siguiente:

Siendo:

- R: coste de la redacción.
- P: presupuesto.

El valor de P se obtiene sumando los costes de las secciones anteriores tal como se muestra en la tabla.

*Tabla 7. Precios y costes de la ejecución del trabajo más la amortización.*

Concepto	Coste
Tarifa de honorarios por tiempo empleado	13.478,40 €
Amortización del material hardware	275,06 €
Amortización del material software	0 €
<b>Total</b>	<b>13.753,46 €</b>

De este modo, se obtiene:

$$R = 0,05 \cdot 13.753,46 = 687,67 \text{ €}$$

Al coste de redacción obtenido hasta el momento se le deben añadir ciertos gastos adicionales, quedando el importe final de redacción del trabajo como se describe en la tabla 7. Por consiguiente, la redacción del trabajo asciende al total de **SETECIENTOS TREINTA Y CINCO EUROS CON SESENTA Y DOS CÉNTIMOS**.

Tabla 8. Coste final de redacción del trabajo.

Concepto	Coste
Redacción del trabajo	687,67 €
Papel de impresión	5,00 €
Impresión	25,00 €
Encuadernación	6,00 €
CD-ROM de 700 MB	11,95 €
<b>Total</b>	<b>735,62 €</b>

## Derechos de visado

El COITT establece que, para proyectos técnicos de carácter general, los derechos de visado para 2018 se calculan en base a:

$$V = 0,006 \cdot P \cdot C$$

Siendo:

- V: coste del visado.
- P: presupuesto.
- C: coeficiente reductor en función del presupuesto.

Tabla 9. Coeficientes para el cálculo del visado.

Coste del presupuesto (€)	Factor de Correlación (C)
Hasta 30.050	1
Exceso de 30.050 hasta 60.101	0,9
Exceso de 60.101 hasta 90.151	0,8
Exceso de 90.151 hasta 120.202	0,7
...	

Tabla 10. Cálculo total P para el cálculo del visado (Presupuesto base).

Concepto	Coste
----------	-------

Coste del tiempo empleado en la ejecución y amortización	13.753,46 €
Redacción del trabajo	735,62 €
<b>Total</b>	<b>14.489,08 €</b>

El valor del coeficiente se obtiene de la tabla 9, mientras que el valor de P es el valor total del presupuesto, que se pasa a calcular en la tabla 10. Por lo tanto, el valor del visado asciende a:

$$V = 0,006 \cdot 14.489,08 \cdot 1 = 86,93€$$

Los costes de derechos de visado del trabajo ascienden a un total de **OCHENTA Y SEIS EUROS CON NOVENTA Y TRES CÉNTIMOS**.

## Gastos de tramitación y envío

Los gastos de tramitación y envío según la tarifa asciende a **SEIS EUROS** por cada documento visado de forma telemática.

## Presupuesto antes de impuestos

Sumando todos los conceptos calculados hasta el momento, se obtiene el presupuesto, sin incluir los impuestos, que se muestra en la tabla 11.

El presupuesto calculado, antes de incluir los impuestos, asciende a **CATORCE MIL QUINIENTOS OCHENTA Y DOS EUROS CON UN CÉNTIMO**.

*Tabla 11. Presupuesto total sin impuestos.*

Concepto	Coste
Presupuesto base	14.489,08 €
Derechos de visado	86,93 €
Gastos de tramitación y envío	6,00 €
<b>Total</b>	<b>14.582,01 €</b>

## Presupuesto incluyendo impuestos

Al presupuesto calculado anteriormente hay que incluirle un 7 % de IGIC obteniendo el coste del presupuesto final (tabla 12).

*Tabla 12. Presupuesto total.*

Concepto	Coste
Total (sin IGIC)	14582,01 €
IGIC (7%)	1020,74 €
<b>Total</b>	<b>15.602,75 €</b>

Por tanto, el presupuesto total, incluyendo impuestos, asciende a la cantidad de ***QUINCE MIL SEISCIENTOS DOS EUROS CON SETENTA Y CINCO CÉNTIMOS.***

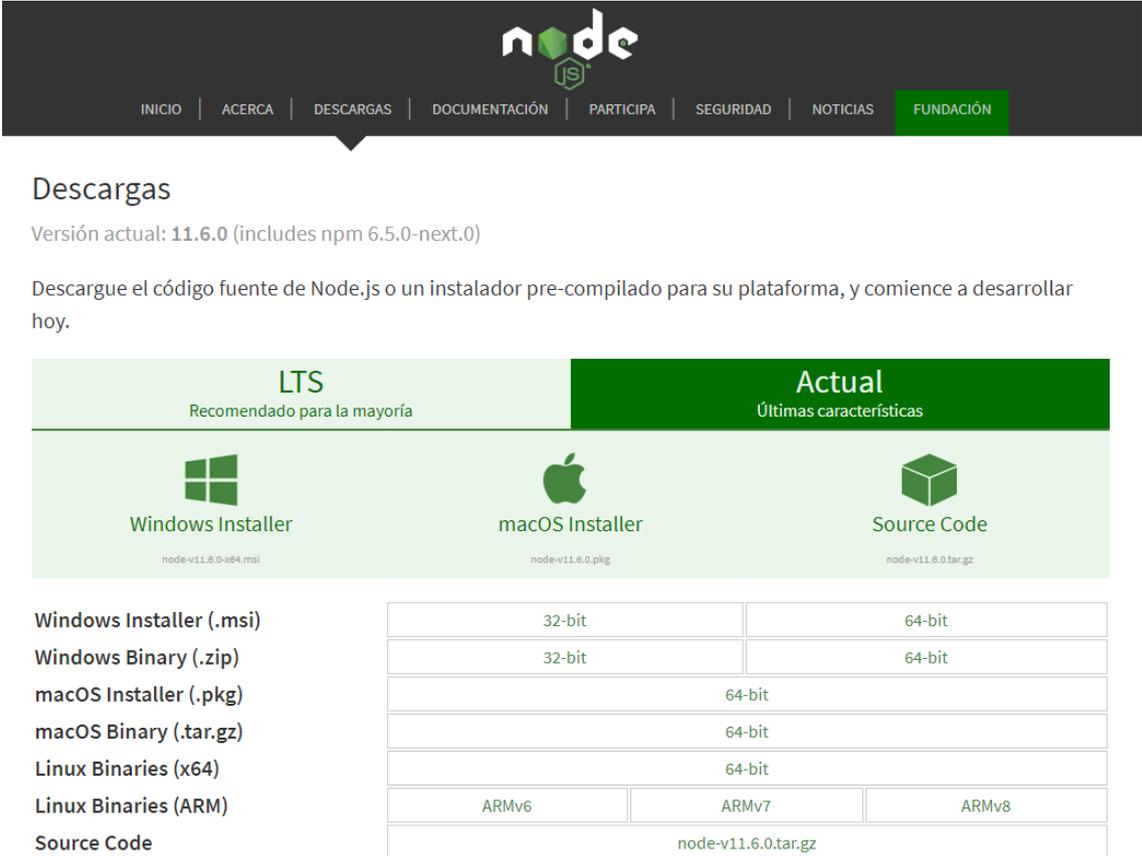
En Las Palmas de Gran Canaria, a 10 de enero de 2019.

Fdo. Ángel Caamaño Lourido

# Anexos

# 1. Instalación de Node.Js y otros módulos

Para la instalación de Node.JS es necesario visitar la URL <https://nodejs.org/es/download/current/>, donde se selecciona el sistema operativo y la versión. Para el proyecto se ha utilizado la última versión disponible.



The screenshot shows the Node.js download page. At the top, there is a navigation bar with links: INICIO, ACERCA, DESCARGAS, DOCUMENTACIÓN, PARTICIPA, SEGURIDAD, NOTICIAS, and FUNDACIÓN. Below the navigation bar, the page is titled 'Descargas' and shows the current version: 'Versión actual: 11.6.0 (includes npm 6.5.0-next.0)'. The page instructs users to download the source code or a pre-compiled installer. There are two main sections: 'LTS' (Recommended for the majority) and 'Actual' (Latest features). Under the 'Actual' section, there are three download options: 'Windows Installer' (node-v11.6.0-x64.msi), 'macOS Installer' (node-v11.6.0.pkg), and 'Source Code' (node-v11.6.0.tar.gz). Below these options, there is a table listing download links for various architectures.

	32-bit	64-bit	
Windows Installer (.msi)			
Windows Binary (.zip)	32-bit	64-bit	
macOS Installer (.pkg)		64-bit	
macOS Binary (.tar.gz)		64-bit	
Linux Binaries (x64)		64-bit	
Linux Binaries (ARM)	ARMv6	ARMv7	ARMv8
Source Code	node-v11.6.0.tar.gz		

Figura 56. Descarga de Node.JS.

Una vez finalizado su descarga, se ejecuta. Al abrirse, el instalador simplemente habrá que clicar “next” hasta finalizar. Hay algunos pasos en los que hay que aceptar los términos y condiciones, la dirección de instalación y la instalación de otros softwares adicionales para Node.JS.

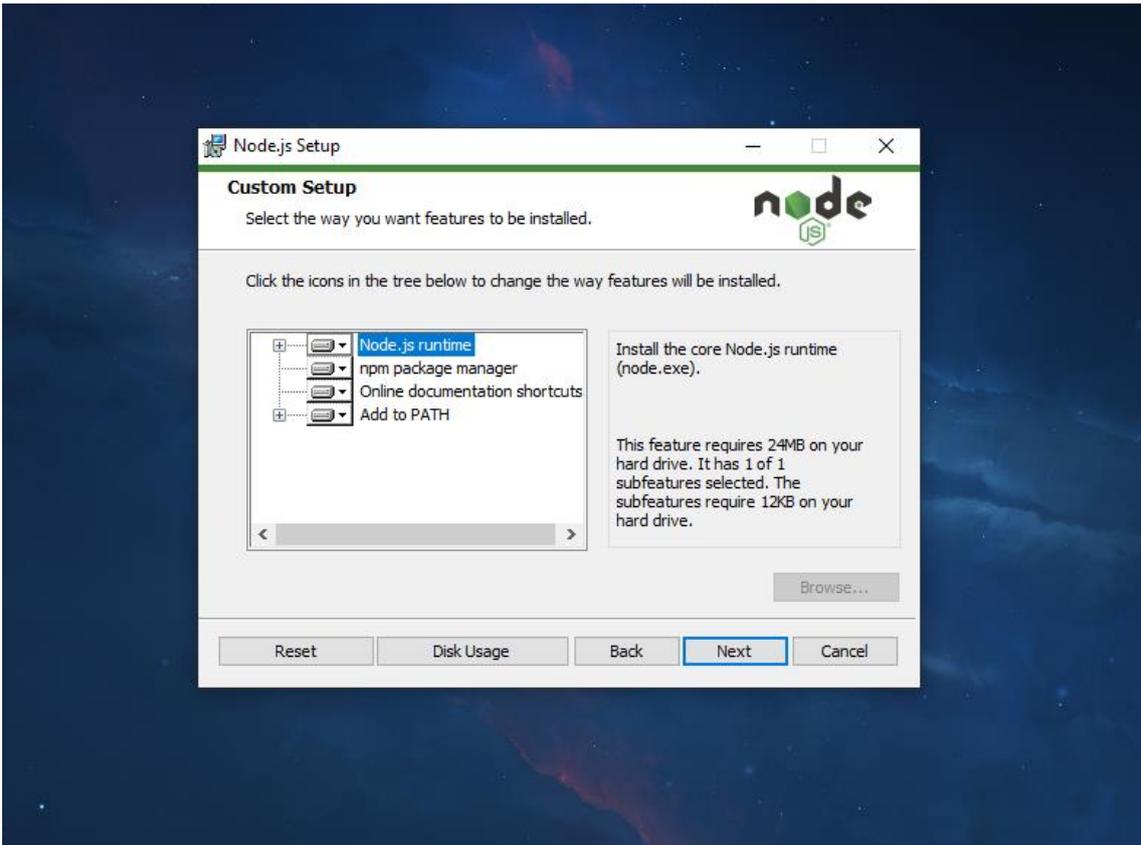


Figura 57. Opciones en la instalación (I)

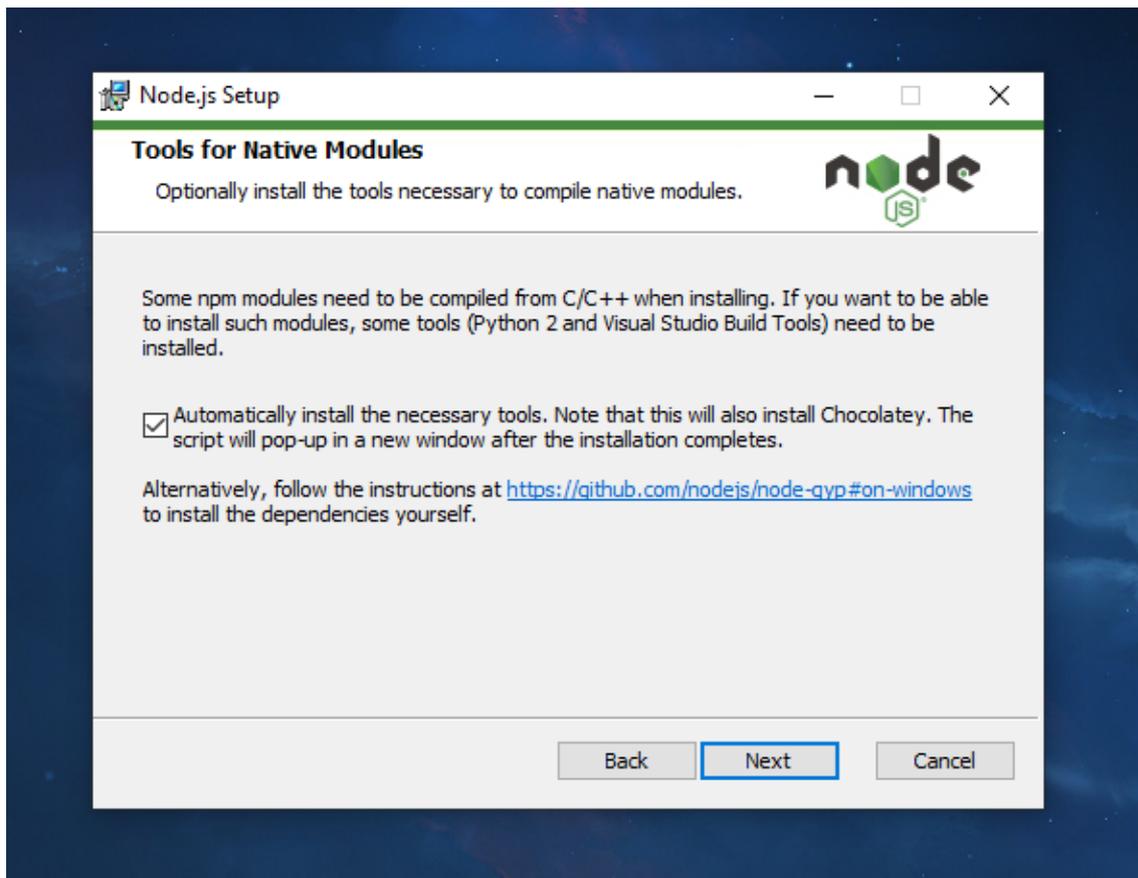


Figura 58. Opciones en la instalación (II).

Una vez finalizada, se abre el **Símbolo del sistema**, nos dirigimos al directorio donde se pretenda guardar el proyecto. Luego, se escribe **npm init** para la creación de un proyecto. Para la instalación de los módulos, el comando **npm install NombreDelMódulo** dentro de la carpeta donde se quiere guardar todo el proyecto.

Las dependencias instaladas (express, por ejemplo) se encuentran dentro del archivo **package.json**. Para la ejecución del proyecto se escribe el comando **npm start** (gracias a la dependencia **nodemon**).

## 2. Instalación de MongoDB

Para la instalación de MongoDB es necesario visitar la URL <https://www.mongodb.com/download-center/community>, donde se descarga el sistema operativo y la versión. Para el proyecto se ha descargado la última versión disponible de **MongoDB Community Server**.

The screenshot shows the MongoDB Download Center interface. At the top, there is a navigation bar with links for 'DOCS', 'LEARN', 'WHAT'S MONGODB?', 'BLOG', and 'LOGIN'. A search icon and a 'Get MongoDB' button are also present. Below the navigation bar, the main heading reads 'MongoDB Download Center'. The page is divided into three tabs: 'Cloud', 'Server', and 'Tools', with 'Server' being the active tab. A prompt asks the user to 'Select the server you would like to run:'. Two options are available: 'MongoDB Community Server' (labeled 'FEATURE RICH, DEVELOPER READY') and 'MongoDB Enterprise Server' (labeled 'ADVANCED FEATURES, PERFORMANCE GRADE'). The 'MongoDB Community Server' option is selected. Below this, there are three dropdown menus: 'Version' (set to '4.0.5 (current release)'), 'OS' (set to 'Windows 64-bit x64'), and 'Package' (set to 'MSI'). A prominent green 'Download' button is positioned to the right of these dropdowns. Below the 'Download' button, a URL is displayed: [https://fastdl.mongodb.org/win32/mongodb-win32-x86\\_64-2008plus-ssl-4.0.5-signed.msi](https://fastdl.mongodb.org/win32/mongodb-win32-x86_64-2008plus-ssl-4.0.5-signed.msi). To the right of the download options, a list of links is provided: 'Release notes', 'Changelog', 'All version binaries', 'Installation instructions', 'Download source (tgz)', and 'Download source (zip)'.

Figura 59. Descarga de MongoDB.

Durante la instalación, se acepta los términos y condiciones y se pulsa siguiente hasta acabar.



*Figura 60. Instalación de MongoDB.*

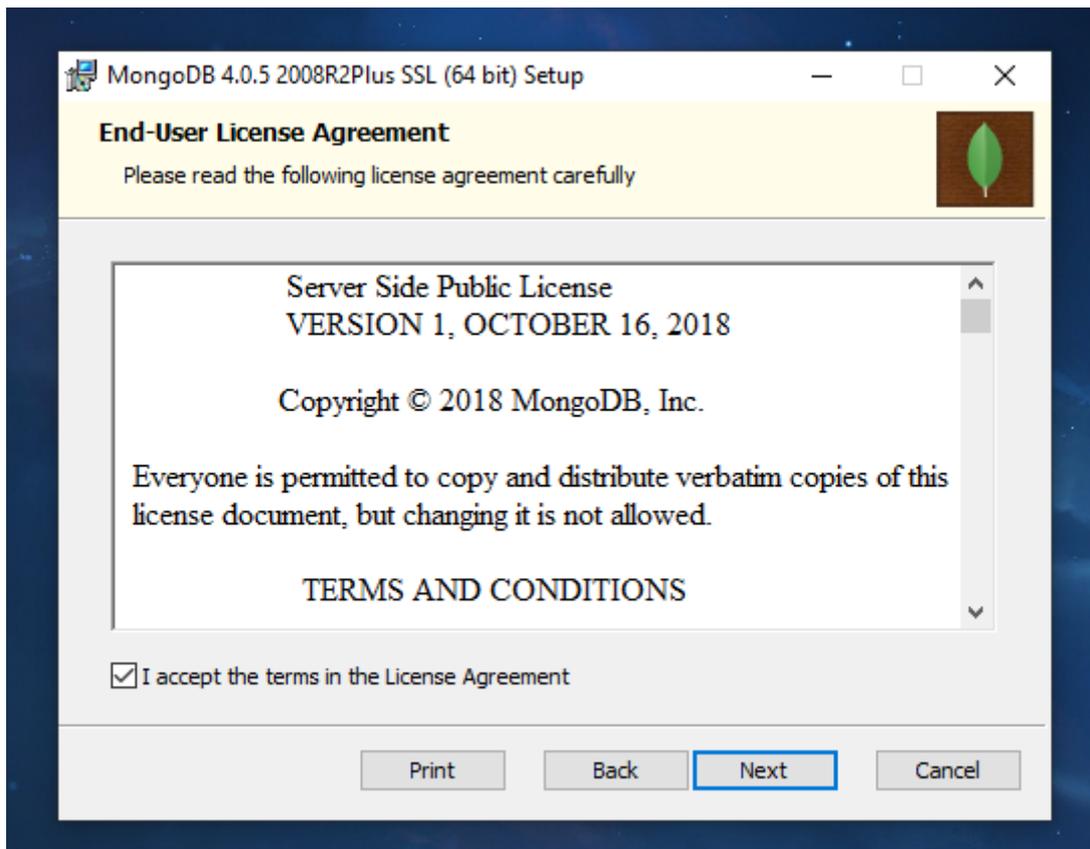


Figura 61. Términos y condiciones de MongoDB.

### 3. Instalación de Angular CLI

Para la instalación de Angular CLI basta con la introducción del comando `npm install -g angular-cli`. Para la creación de un proyecto nuevo sería el comando `ng new NombreProyecto`. Se creará una carpeta con el proyecto. Para su ejecución se escribe `ng serve`.

### 4. Instalación de Python

Para la instalación de Python es necesario visitar la URL <https://www.python.org/downloads/>, donde se descarga el sistema operativo y la versión. Para el proyecto se ha descargado la última versión disponible de **Python3**.

**Download the latest version for Windows**

[Download Python 3.7.2](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#)

Looking for Python 2.7? See below for specific releases

Looking for a specific release?  
Python releases by version number:

Release version	Release date		Click for more
<a href="#">Python 3.7.2</a>	2018-12-24	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.6.8</a>	2018-12-24	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.7.1</a>	2018-10-20	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.6.7</a>	2018-10-20	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.5.6</a>	2018-08-02	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.4.9</a>	2018-08-02	<a href="#">Download</a>	<a href="#">Release Notes</a>

Figura 62. Descarga de Python.

Para su instalación simplemente hay aceptar los términos y condiciones y seguir la instalación hasta finalizar.

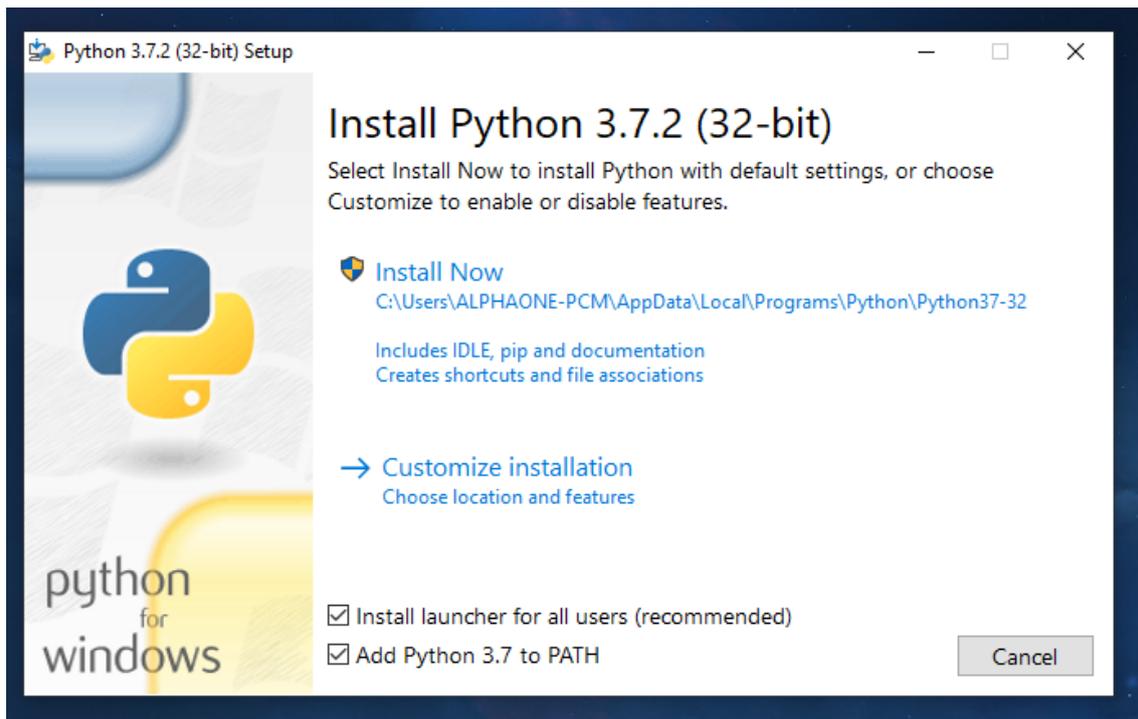


Figura 63. Instalación de Python.

Es necesario instalar un módulo que permitirá instalar nuevas dependencias para proyectos basados en Python. Se llama **pip** y para su instalación es necesario abrir la URL <https://bootstrap.pypa.io/get-pip.py> y crear un archivo Python (get-pip.py, por ejemplo) añadiendo todo el contenido de la URL. Luego, se ejecuta el comando **python get-pip.py to install pip** dentro del directorio donde se encuentra el archivo creado anteriormente.

## 5. Instalación de Arduino IDE

Para la instalación de Arduino IDE es necesario visitar la URL <https://www.arduino.cc/en/Main/Donate>, donde automáticamente detecta el sistema operativo.

### Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **29,139,870** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3 \$5 \$10 \$25 \$50 OTHER

JUST DOWNLOAD **CONTRIBUTE & DOWNLOAD**

Figura 64. Descarga de Arduino IDE.

Para su instalación se es necesario aceptar los términos y condiciones y seguir avanzando hasta finalizar.

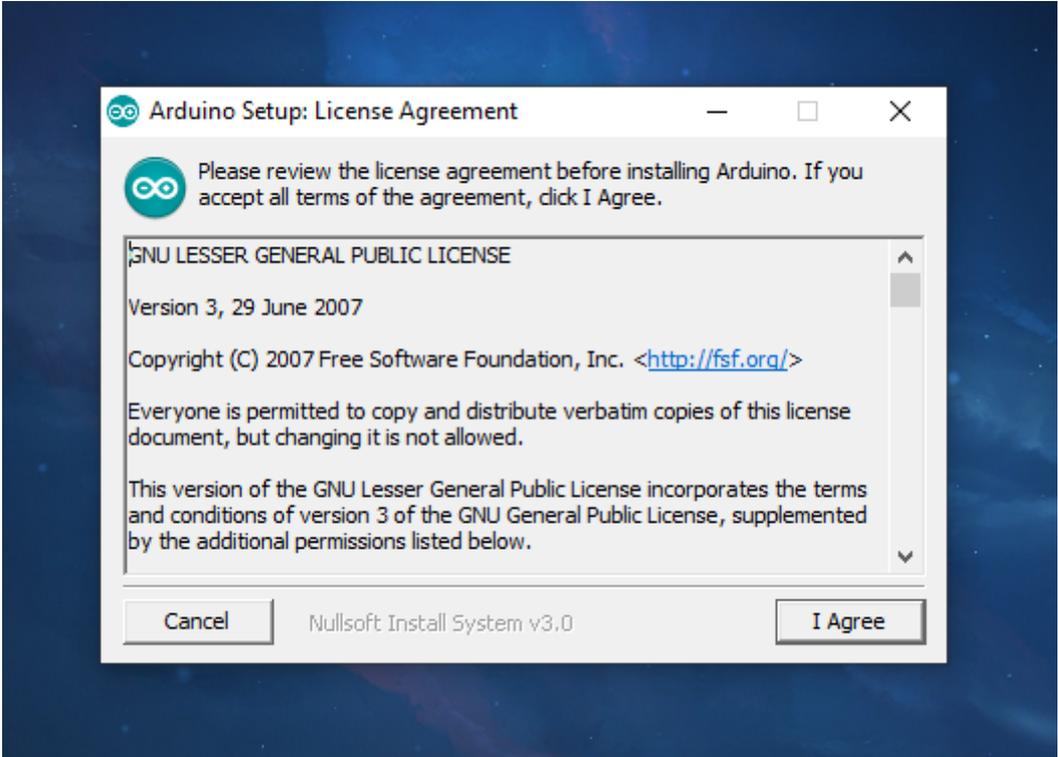
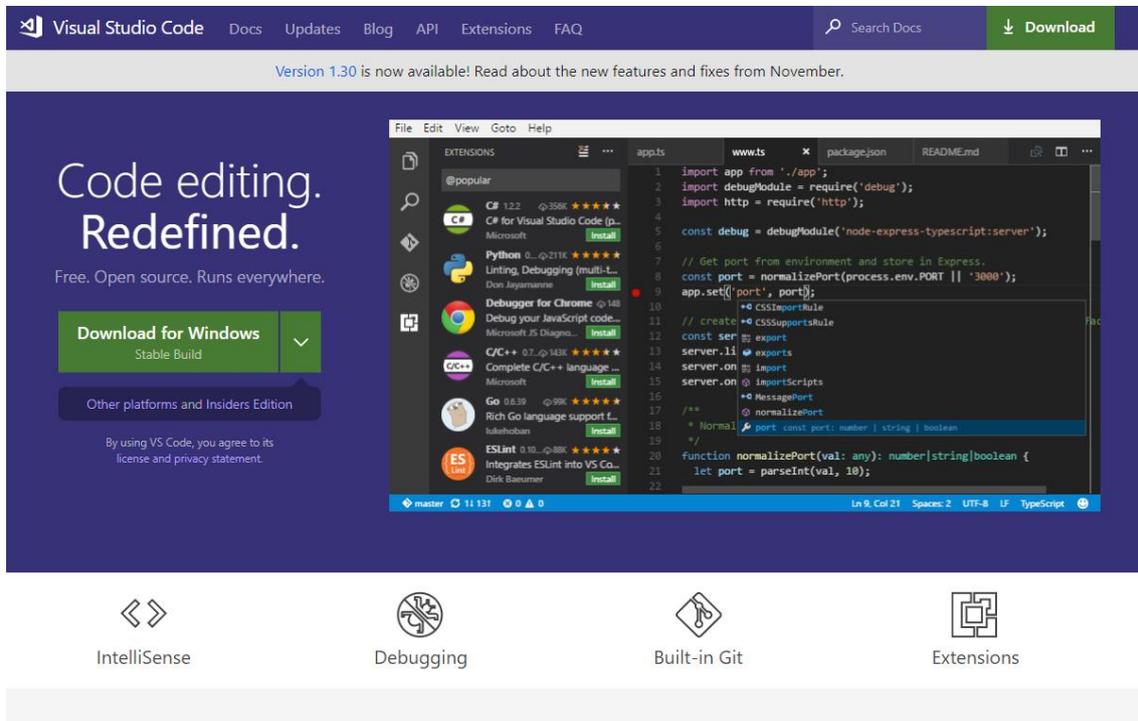


Figura 65. Términos y condiciones de Arduino IDE.

## 6. Instalación de Visual Studio Code

Para la descarga de Visual Studio Code es necesario visitar la URL <https://code.visualstudio.com/>, donde automáticamente detecta el sistema operativo.



Para su instalación, simplemente aceptamos los términos y condiciones y seguir con la instalación hasta finalizar.

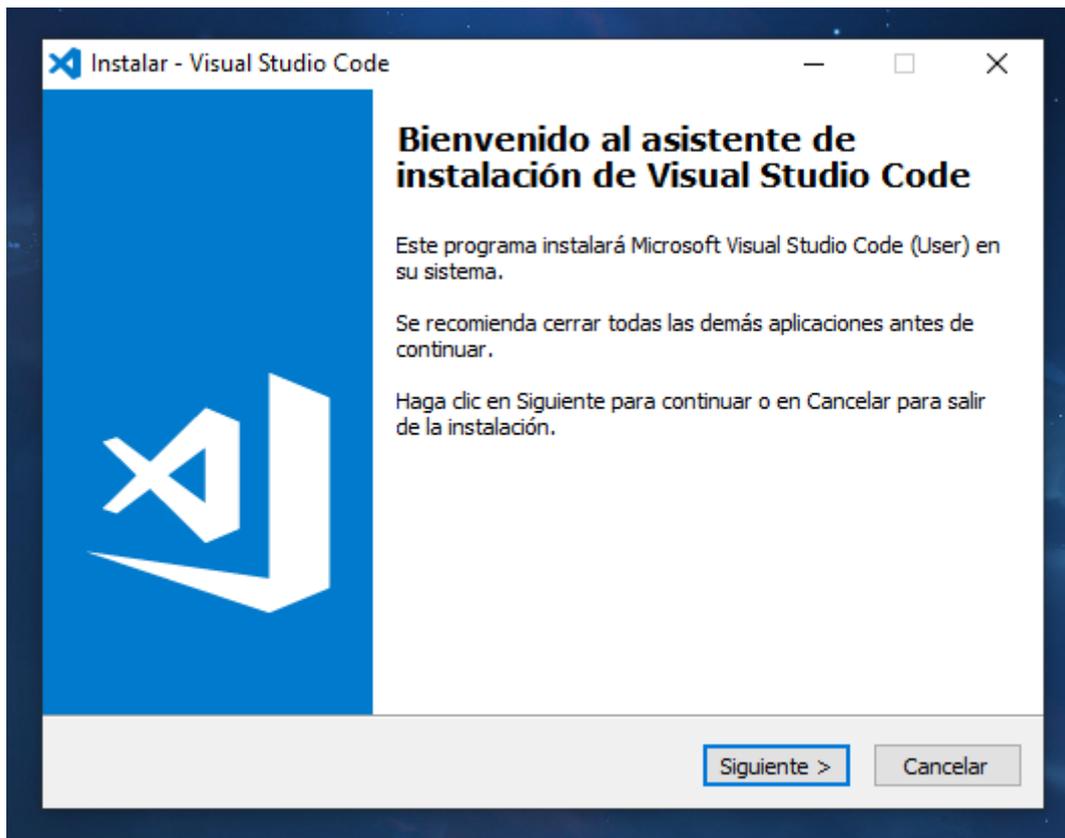


Figura 66. Instalación de Visual Studio Code.

## 7. Instalación de Postman

Para la descarga de Postman es necesario visitar la URL <https://www.getpostman.com/downloads/>, donde automáticamente detecta el sistema operativo.

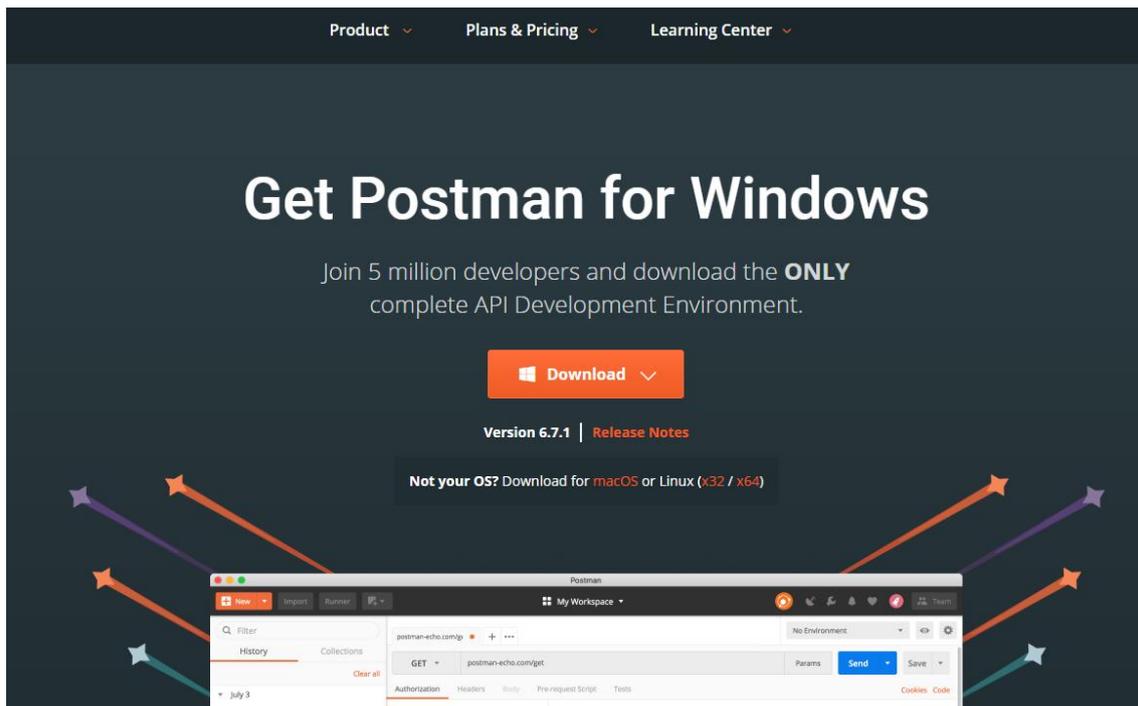


Figura 67. Descarga de Postman.

Para su instalación, simplemente se clikea sobre el fichero y automáticamente se instala.

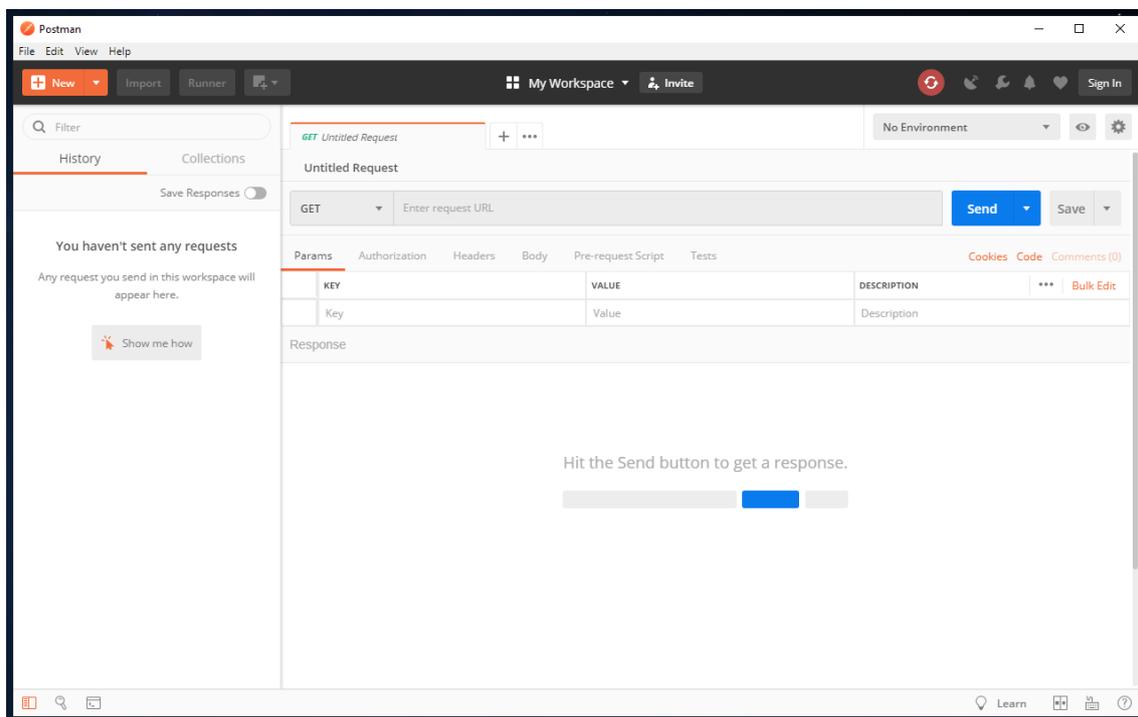


Figura 68. Ventana inicial de Postman.

## Contenido del CD-ROM

Adjunto a la memoria del presente Trabajo Fin de Grado se encuentra disponible un Compact Disc Read-Only Memory (CD-ROM). En este anexo se especifica el contenido incluido:

- En el directorio Memoria Trabajo Fin de Grado se encuentra la memoria del TFG “Interfaz VLC-LoRa con plataforma de gestión” en lengua española y en formato PDF.
- En el directorio **APP** se encuentra todo el código involucrado en el TFG donde:
  - **api** es donde se encuentra el back-End.
  - **client** donde se encuentra el Front-End.
  - **emisor** donde se encuentran los dos programas basados en Python (emisor.py y emisorRand.py).
  - **arduino** donde se encuentran las librerías y los programas de las diferentes placas: unoRx para el Arduino UNO y adkTx para el Arduino Mega.