



UNIVERSITAT  
POLITÀCNICA  
DE VALÈNCIA

CAMPUS DE GANDIA

---

***Proyecto GTI\_2A (GTI\_2A\_20\_TEAM\_12)***

***Murbin - Farolas inteligentes***

***Documento Técnico de Diseño***

***Volumen 2***

## ÍNDICE

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>1</b>
1.1	PROPÓSITO .....	1
1.2	ALCANCE.....	1
1.3	EQUIPO DE DESARROLLO E INTERESADOS EN EL PROYECTO .....	4
1.4	DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS .....	6
1.5	ESTRUCTURA DEL DOCUMENTO .....	6
1.6	REPOSITORIO GIT Y LISTADO DOCUMENTAL.....	6
<b>2</b>	<b>SISTEMA.....</b>	<b>7</b>
2.1	CARACTERÍSTICAS DEL SISTEMA.....	7
2.2	ARQUITECTURA DEL SISTEMA .....	9
<b>3</b>	<b>RED DE NODOS .....</b>	<b>10</b>
<b>4</b>	<b>CONCLUSIONES Y LÍNEAS DE MEJORA .....</b>	<b>11</b>
<b>5</b>	<b>REFERENCIAS.....</b>	<b>12</b>

# 1 INTRODUCCIÓN

Murbin (Mobiliario urbano inteligente) plantea un prototipo de alumbrado público inteligente. Más concretamente, en nuestro caso, ser un sistema inteligente implica varias cosas: la conexión a Internet, la capacidad de realizar lecturas de diversas magnitudes físicas, así como la transmisión de éstas a posteriori y la activación automática de comportamientos preestablecidos (actuadores) en función del valor de medidas nominales. Por tanto, se trata de un proyecto que recae sobre el marco del Internet de las cosas (IoT) y requiere de numerosas tecnologías y disciplinas de la ingeniería.

Se recopila información sensible de campo, a fin de monitorizar y optimizar la actividad ambiental (intensidad de la luz, contaminación, nivel de ruido, presencia humana, geolocalización del dispositivo, etc.). Se propone implementar también un mecanismo de autoabastecimiento (célula fotovoltaica, batería y convertidor buck-boost). Para minimizar el consumo energético, se prevee la realización de las mediciones cada cierto tiempo. Ésto es configurable. Inmediatamente entraría en modo de hibernación o stand-by, hasta la siguiente captura de datos o el acaecimiento de una interrupción producida por alguna subrutina.

Para implementar las susodichas funcionalidades, se ha optado por utilizar el microcontrolador **ESP32** (como esclavo), y el micro-ordenador **Raspberry Pi 3** (como maestro). También son necesarios múltiples dispositivos y componentes electrónicos (resistencias, amplificadores operacionales, diodos, termistores, sensores específicos, ...). El ESP32 puede ser programado utilizando la plataforma de desarrollo Arduino.

## 1.1 PROPÓSITO

En este documento se especifica el planteamiento inicial de las características técnicas del producto, así como su funcionamiento. Además, proporciona información sobre los dispositivos y tecnologías utilizados. Éste es un documento técnico dirigido a un público con conocimiento tecnológico mínimo, relativo al campo de la electrónica y la programación. Su redacción pretende exponer de manera rigurosa y objetiva las capacidades, propiedades y limitaciones del sistema. Con ello, se espera concretar las amplias posibilidades del producto.

## 1.2 ALCANCE

El término de alumbrado público inteligente hace referencia a alumbrado capaz de detectar diferentes niveles de magnitud y adaptar la intensidad de la luz de manera proporcional. Es decir, para diferentes condiciones la actuación de la farola será diferente. La intensidad baja cuando no se detecta movimiento y, por el contrario, sube cuando se detecta. Esto supone un cambio drástico de paradigma; contrasta con el tradicional alumbrado estático de intensidad variable por tiempos fijos (programado con reloj). Las posibilidades se extienden más allá de la luminosidad, permitiéndonos interactuar con los usuarios del medio público de manera acorde a la situación. Otro ejemplo sería el de cambiar la temperatura de los haces de luz en función del nivel de ruido del entorno. Así pues, es un sistema directamente ligado al desarrollo y la evolución de las Smart Cities [1] [2] [3].

Aclarado esto, los objetivos del producto son la optimización y reducción del consumo energético, el control de la contaminación en sus múltiples expresiones (sonora, lumínica y CO2), mejorar la experiencia de usuario del viandante y la creación de una interfaz

homogénea y céntrica (aplicación Android) para consultas varias: búsqueda de parking, trazado de caminos, alertas personalizadas en función del nivel de ruido u otras magnitudes de interés, mapas de calor y un largo etcétera.

También estará equipado con un sistema de posicionamiento global, bien sea un receptor GPS físico (hardware) o un par de coordenadas geográficas fijadas durante la instalación (software). Además el dispositivo ha de ser hermético; se puede estropear si alguno de sus componentes internos entra en contacto con el agua, provocando un cortocircuito.

Cabe recalcar que solo se han mencionado algunas de las funcionalidades pensadas desde un punto de vista empático con el perfil de usuario del viandante. No obstante, se han definido otros perfiles como el del técnico, el administrador y (opcionalmente) el investigador o científico. Las funcionalidades a implementar deberán ser específicas (por lo general) al tipo de usuario en cuestión. Véanse debajo los perfiles de usuario estudiados y los porcentajes aproximados que se sopesan.

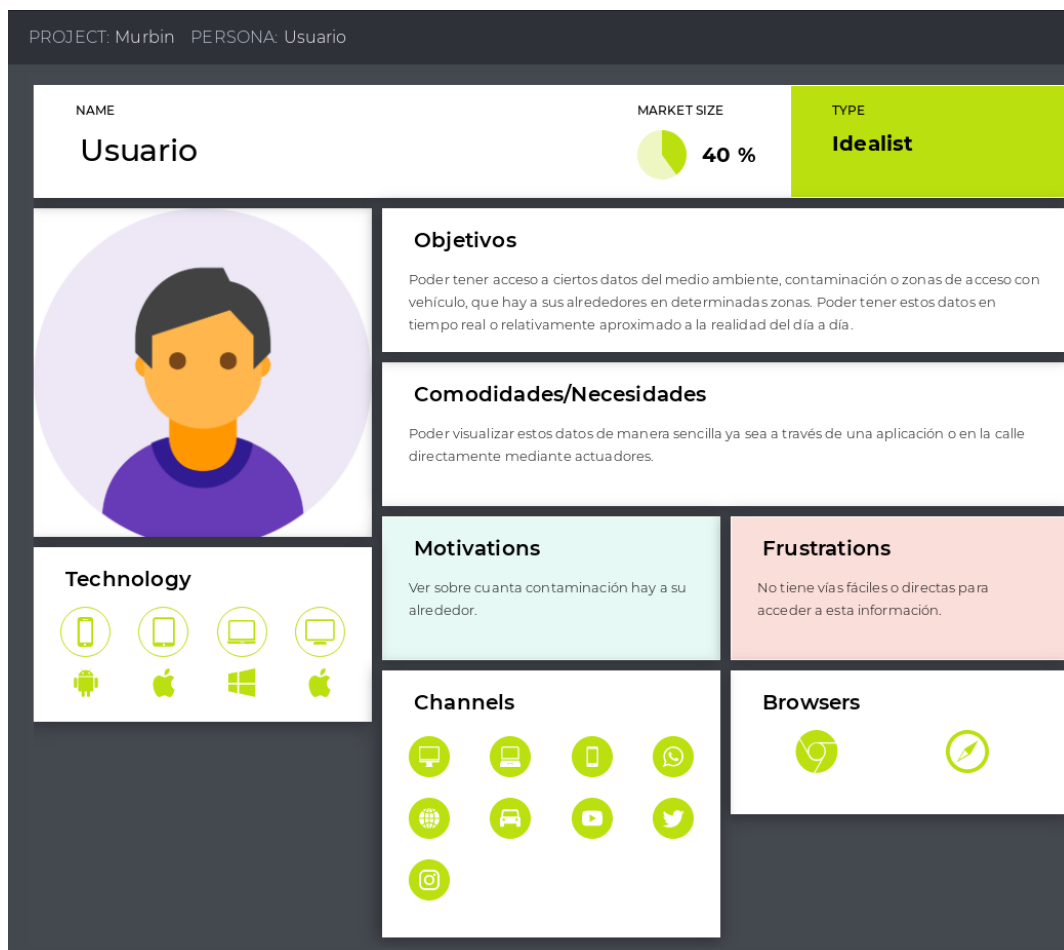


Ilustración 1. Perfil de usuario (usuario).

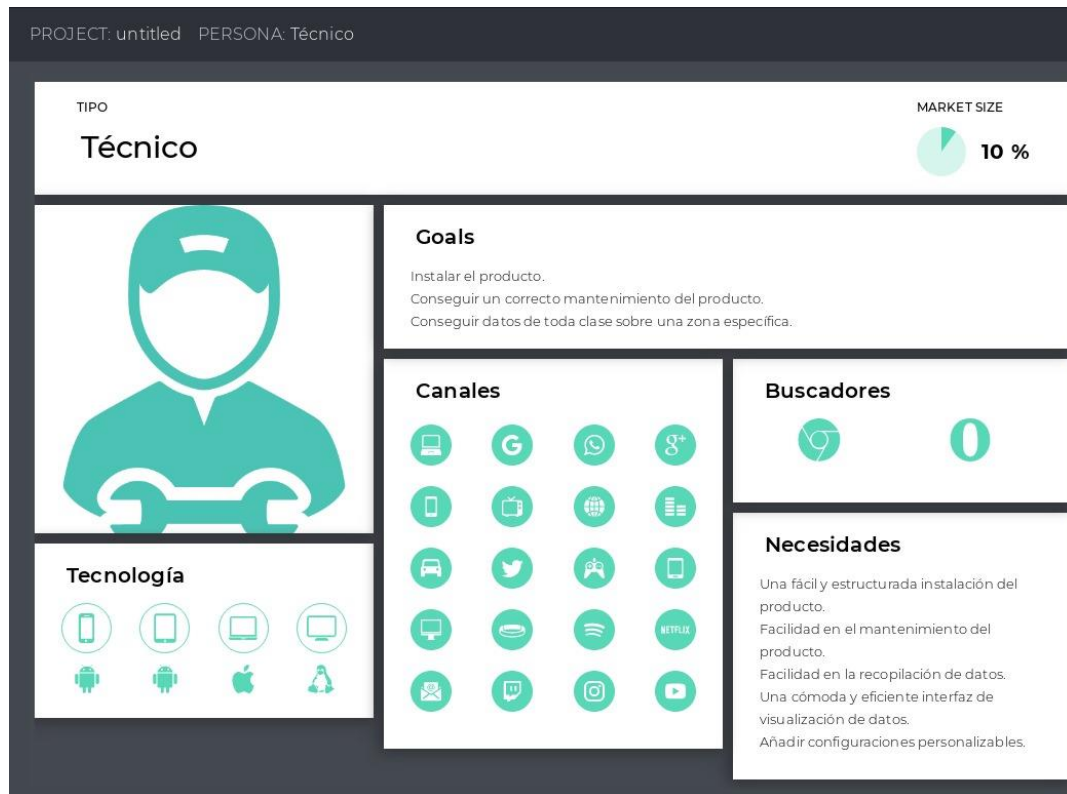


Ilustración 2. Perfil de usuario (técnico).

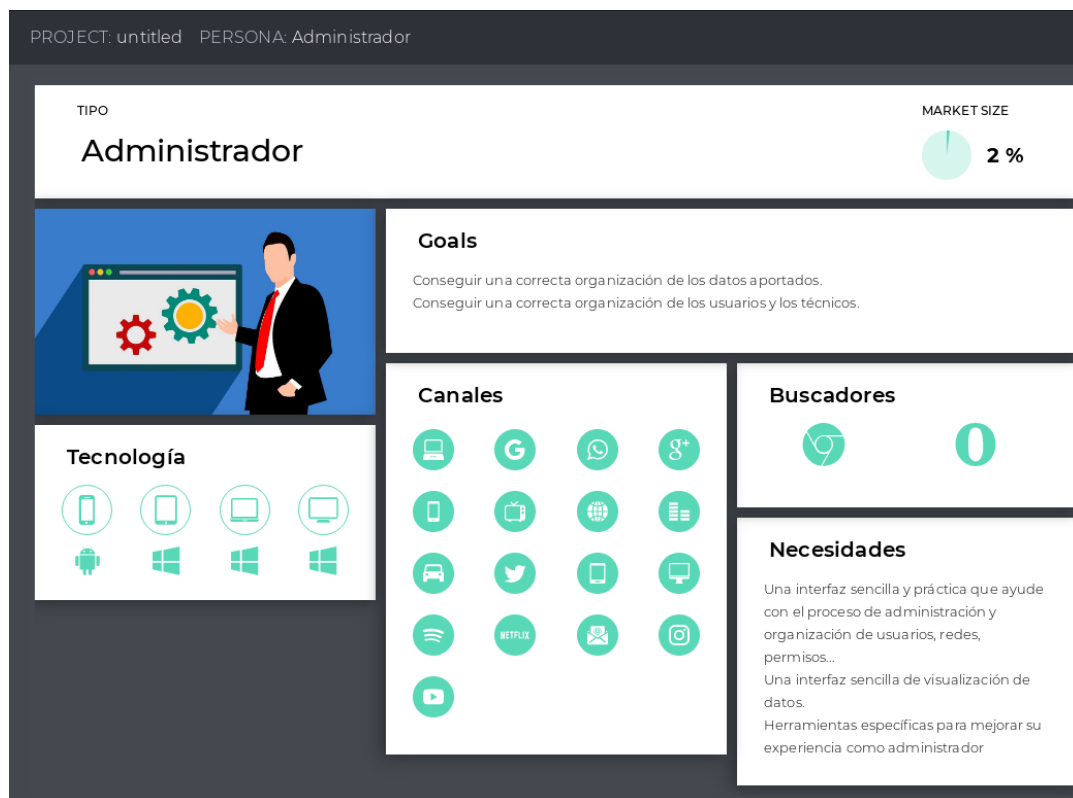


Ilustración 3. Perfil de usuario (administrador).

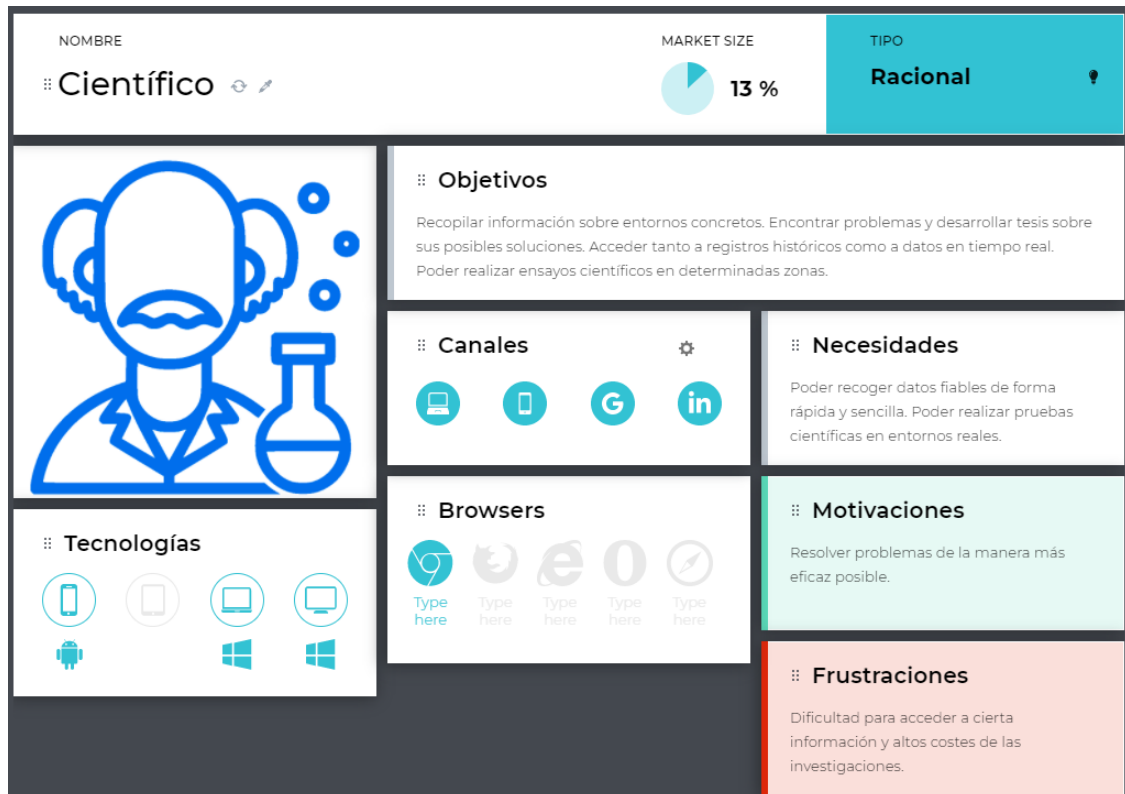


Ilustración 4. Perfil de usuario (científico).

### 1.3 EQUIPO DE DESARROLLO E INTERESADOS EN EL PROYECTO

A continuación se listan todos los integrantes del equipo de desarrollo del proyecto, así como algunas personalidades de interés por su carácter asesorativo y/o de formación.

#### *Miembros del equipo*

Abidán Brito Clavijo

- Estudiante de 2º del grado en Tecnologías Interactivas.
- Manager del equipo.
- Desarrollo electrónico.
- Desarrollo software embebido.

Alejandro Losa García

- Estudiante de 2º del grado en Tecnologías Interactivas.
- Diseño visual y de interacción.
- Maquetación interfaz gráfica.

Francisco Javier Paños Madrona

- Estudiante de 2º del grado en Tecnologías Interactivas.
- Desarrollo de aplicación Android.

Yeray Candel Sampedro

- Estudiante de 2º del grado en Tecnologías Interactivas.
- Desarrollo electrónico.
- Desarrollo software embebido.

Raúl De La Fe Robles

- Estudiante de 2º del grado en Tecnologías Interactivas.
- Generalista.

***Otros interesados en el proyecto***

José Marín-Roig Ramón

- Profesor Titular en Universitat Politècnica de València.
- Product Owner.
- Asesor.

Jesús Tomás Gironés

- Profesor Titular en Universitat Politècnica de València.
- Formación teórica Android.
- Asesor.

Stella María Heras Barberá

- Profesora Asociada en Universitat Politècnica de València.
- Formación práctica Android.
- Asesora.

Enrique Colomar Pous

- Profesor Titular en Universitat Politècnica de València.
- Formación práctica de laboratorio (desarrollo electrónico).

Tomás Carlos Sogorb Devesa

- Profesor Titular en Universitat Politècnica de València.
- Formación teórica microprocesadores y acondicionadores de señal.

#### **1.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS**

- IDE: entorno de desarrollo integrado (Integrated Development Environment).
- TCP: Transmission Control Protocol.
- IP: Internet Protocol address.
- UDP: User Datagram Protocol.
- HTTP: Hypertext Transfer Protocol.
- UART: Universal Asynchronous Receiver-Transmitter.
- UART: Universal Asynchronous Receiver-Transmitter.
- I2C: Inter-Integrated Circuit.
- SPI: Serial Peripheral Interface.
- GPS: Global Positioning System.

#### **1.5 ESTRUCTURA DEL DOCUMENTO**

El documento se estructura de la siguiente manera:

- /1 La Sección 1 es la introducción e incluye una descripción detallada del proyecto y los documentos referenciados.
- /2 La Sección 2 proporciona una visión general del sistema.
- /3 La Sección 3 incluye el mapa de la red de nodos especificando las diferentes capas
- /4 La Sección 4 presenta las conclusiones del trabajo desarrollado y las posibles líneas de mejora.
- /5 Por último se presentan las referencias del documento

#### **1.6 REPOSITORIO GIT Y LISTADO DOCUMENTAL**

Para la base de código y documentos pertinentes a este proyecto, se ha utilizado un repositorio Git como sistema de control de versiones, alojado en la plataforma GitHub. Para acceder al repositorio, haga click sobre el hipervínculo aquí expuesto:

<https://github.com/abidanBrito/murbin>



## 2 SISTEMA

### 2.1 CARACTERÍSTICAS DEL SISTEMA

El sistema aquí expuesto parte del microcontrolador ESP32 V1, o un M5Stack acoplado al micro-ordenador Raspberry Pi 3. De esta manera podemos tener una red de sensores centralizada. Además, no es necesario el uso de un convertidor Analógico-Digital externo.

#### DOIT ESP32 DEVKIT V1 PINOUT

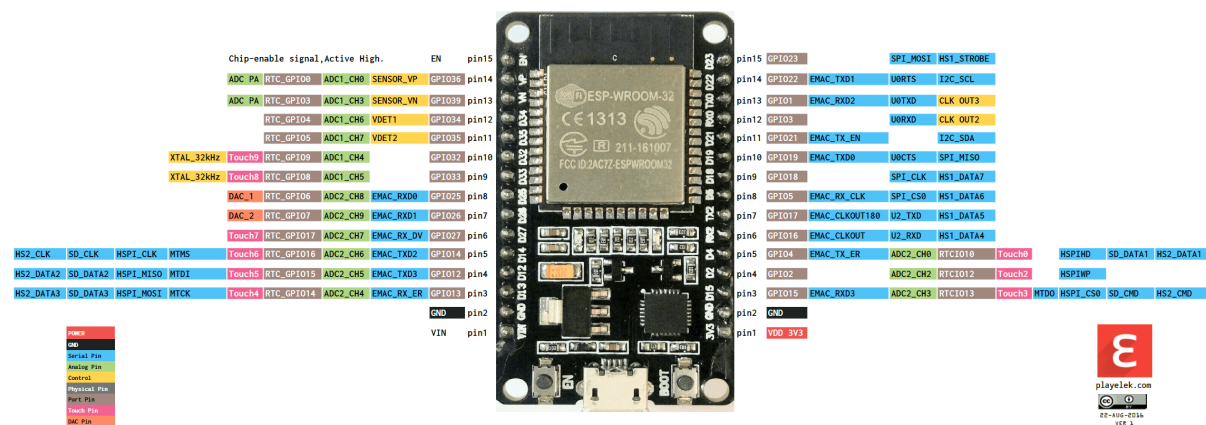


Ilustración 5. Esquema de pines del ESP32.

Lo primero que se debe hacer es establecer una línea de tierra. Los pines de **GND** han de coincidir, y se ha de tener especial precaución con el sentido en que se acoplan los pines. Si se hiciese en una orientación errónea, se podría cortocircuitar la placa.

El circuito del **Transmisor-Receptor Asíncrono Universal (UART)** controla los puertos y dispositivos serie. Se utilizará para interconectar el microprocesador y el microcontrolador, mediante un protocolo de comunicación en serie. De esta manera permitimos el análisis en tiempo real con el monitor en serie del Arduino IDE o el equivalente Logcat del Android Studio. La velocidad de transferencia ha de ser la misma para maestro y esclavo, para que la transmisión de datos coincida. Nótese que la placa ESP32 solo tiene un UART. Es decir, no pueden haber varios periféricos conectados al mismo tiempo por la misma UART. La transmisión/recepción de datos se lleva a cabo por los pines **TXD** y **RXD**. No obstante, al conectar la placa al PC por el puerto **microUSB**, se conecta directamente al UART.

Nuestro sistema utiliza el **protocolo I2C**. Se trata de un protocolo de comunicación en serie, bidireccional (half-duplex) y síncrono. El **bus I2C** se conecta a todos los sensores. Es el mismo para todos. La conexión al microcontrolador se lleva a cabo a través de dos líneas o buses:

- **SDA** ( data): primero se manda el bit de inicio (start) y luego la secuencia; el tren de datos en sí. Finalmente, un bit de acuerdo y otro de parada.
- **SCL / CLK** (clock): bus para el reloj. Marca el ritmo o compás, por así decirlo.

Ésto quiere decir que el maestro y el esclavo envían datos por el mismo bus, el cual es controlado por el maestro, que crea la señal de reloj. Nótese que I2C no utiliza selección de esclavo, sino direccionamiento.

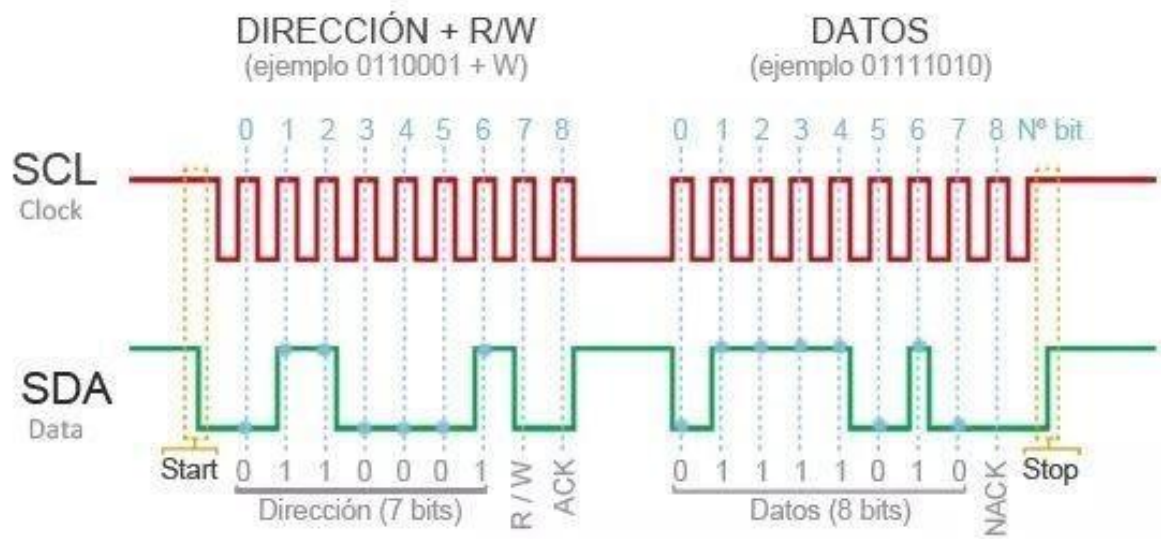


Ilustración 6. Transmisión de bits por I2C.

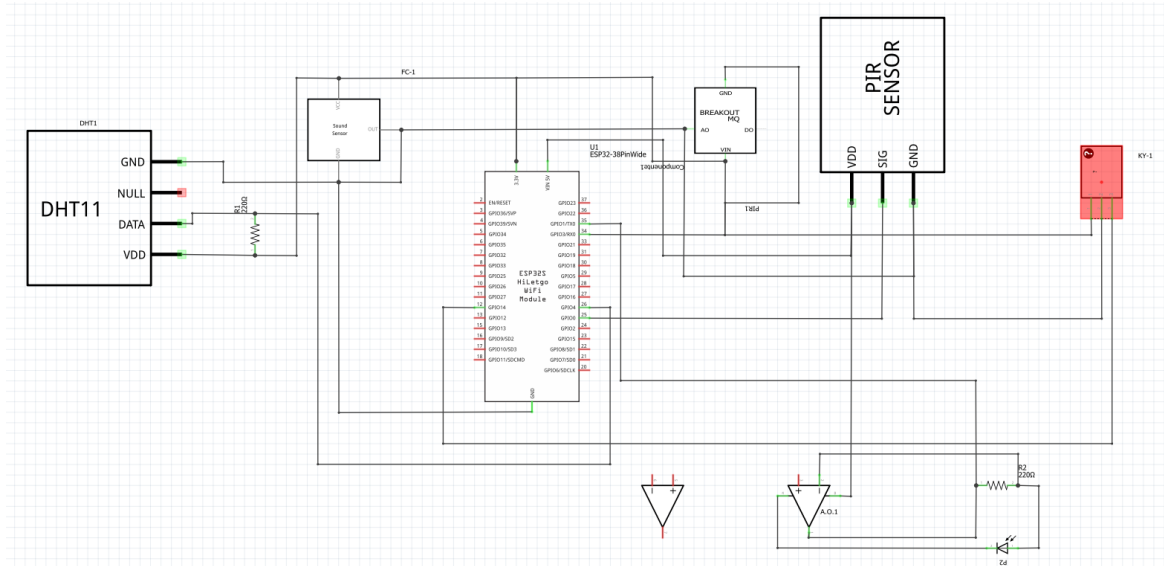


Ilustración 7. Esquemático de los sensores.

## 2.2 ARQUITECTURA DEL SISTEMA

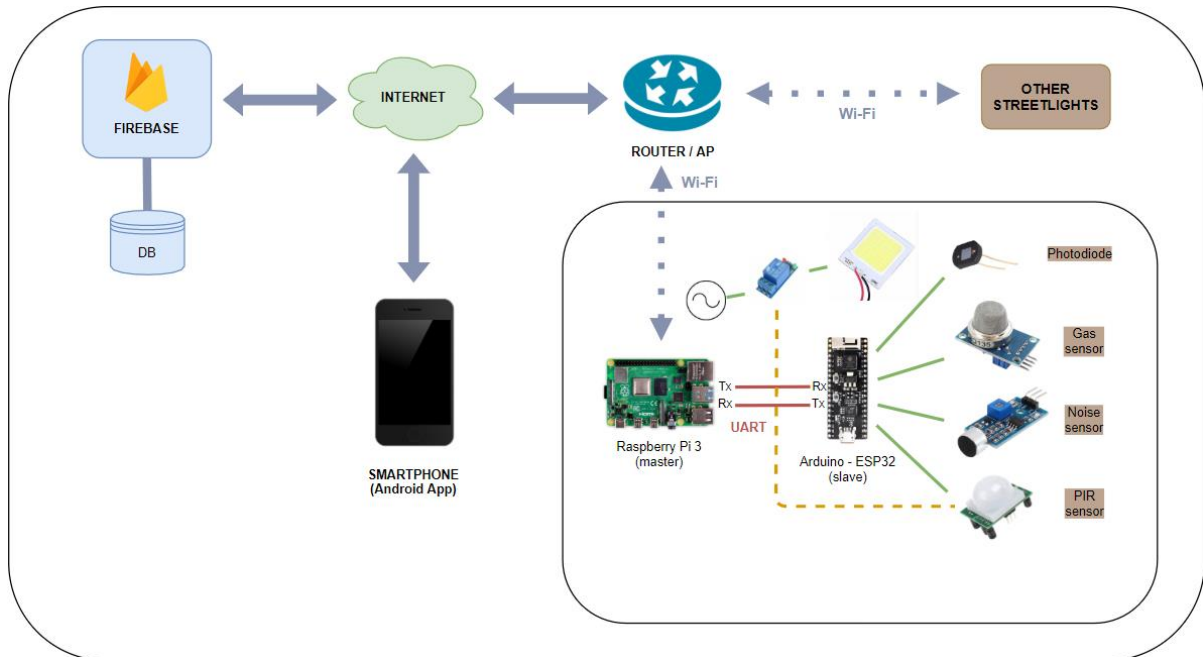


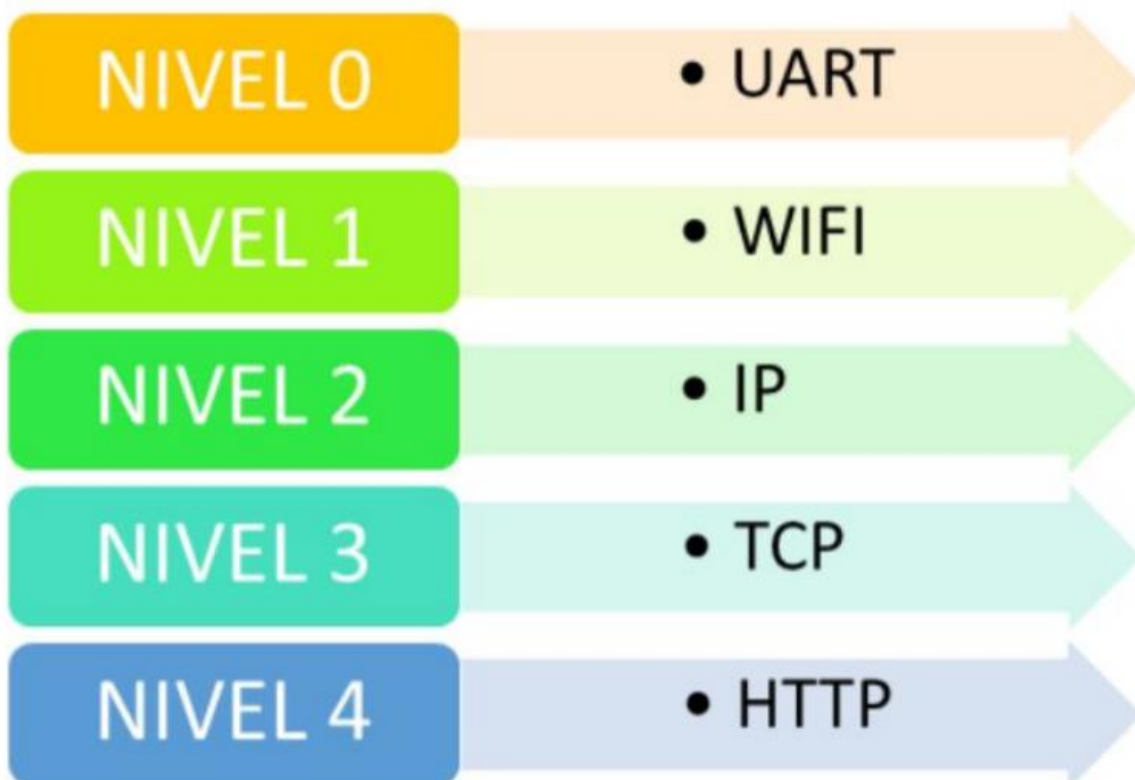
Ilustración 8. Diagrama del sistema.

### 3 RED DE NODOS

Se puede desglosar el camino de los datos de manera esquemática en los siguientes 5 niveles:

1. Comunicación hardware (nivel 0).
2. Acceso a la red (nivel 1).
3. Internet (nivel 2).
4. Transporte (nivel 3).
5. Aplicación (nivel 4).

En primer lugar nuestro microcontrolador se comunica con los sensores mediante el protocolo **I2C** o **SPI**, y con la Raspberry Pi mediante el protocolo UART (con la finalidad de enviar los datos capturados). Éstos, son enviados mediante conexión inalámbrica (Wi-Fi) a un servidor en la nube y almacenados en una **base de datos NoSQL** (Firebase), utilizando el protocolo IP. Ya que la conexión a Internet es la función que más consume, y no es necesario enviar datos en grandes cantidades, se podría utilizar el protocolo **TCP** o el **UDP**. Finalmente estos datos acaban en el servidor web mediante el protocolo **HTTP**. Nótese que, en la práctica no sería necesario tener el prototipo conectado a ningún otro dispositivo, pues es autosuficiente y los datos serían enviados directamente al servidor. A continuación, adjuntamos la siguiente ilustración a modo de ayuda visual. No obstante, en el nivel 0 deberían estar representados los ya mencionados protocolos I2C y SPI.



*Ilustración 9. Red de nodos.*

## 4 CONCLUSIONES Y LÍNEAS DE MEJORA

Tras la implementación de la segunda versión del producto (prototipo del Sprint 2) se extraen las siguientes conclusiones:

- Se ha ampliado la red de sensores interconectada de manera inalámbrica a Internet (tecnología Wi-Fi). A medida que avance el desarrollo la red de sensores seguirá creciendo en magnitud.
- Se ha desarrollado el wireframe de la aplicación, teniendo en cuenta el diseño visual, el diseño de interacción y la experiencia de usuario de la aplicación móvil.
- Se ha creado una base de datos NoSQL en tiempo real.
- Se ha maquetado y configurado una primera versión de la aplicación móvil (Android) con varios roles de usuario autenticados, así como el tipo de usuario invitado.
- Se han implementado nuevos actuadores y se hace uso de relés.
- Se deben reemplazar los actuadores simulados con diodos LED por dispositivos o sistemas propios del contexto y funcionalidad en cuestión. Esto implica concebir las funcionalidades de los distintos perfiles de usuario ya definidos.
- Se deben calibrar y testear los diferentes sensores para el siguiente Sprint.
- Se puede optimizar la fase de desarrollo electrónico adquiriendo el material necesario con la suficiente antelación, así como una mejor gestión de éste entre los miembros involucrados.
- No se ha tenido en cuenta el consumo energético, por ser la primera iteración. No obstante, es algo a tener presente para iteraciones futuras.

## 5 REFERENCIAS

- [1] Zak Derler (2020). [Smart street lighting will help future-proof our cities.](#)
- [2] Esther Fuldauer (2019). [Lighting the road to smart cities and sustainability.](#)
- [3] Chris Teale (2020). [Cities ‘finally waking up’ to the benefits of smart streetlights: survey.](#)
- Wikipedia (2014). [Intelligent street lighting.](#)
- Wikipedia (2020). [Universal asynchronous receiver-transmitter.](#)
- Campbell, Scott (2016). [Basics of UART Communication.](#)
- Loflin, Lewis (2018). [Photodiode Circuit Operation and Uses.](#)
- Universitat Politècnica de València (2017). [Master en Desarrollo de Aplicaciones Android.](#)
- Gironés, T. (2013). *El gran libro de Android*. Barcelona: Marcombo.

## CONTROL DEL DOCUMENTO

**Título:** *Documento Técnico de Diseño*  
**Volumen:** *Vol. 2*  
**Fecha:** *19 Noviembre 2020*  
**Autor:** *Abidán Brito Clavijo*

**Referencia:** *GTI\_20\_2A\_TEAM\_12*  
**Nombre de fichero:** *GTI\_20\_2A\_TEAM\_12-01*

## FIRMAS DEL DOCUMENTO

Naturaleza del firmante	Nombre	Firma	Fecha	Rol
<i>Autor</i>	<i>Abidán Brito Clavijo.</i>		<i>30/10/2020</i>	<i>Miembro y manager del equipo.</i>
<i>Revisor</i>				

## REGISTRO DE CAMBIOS DEL DOCUMENTO

Fecha	Versión	Autor	Detalles de los cambios
<i>20 Octubre 2020</i>	<i>Vol 1. Borrador 1</i>	<i>Abidán Brito Clavijo.</i>	<i>Primer borrador del sistema (Sprint 1).</i>
<i>19 Noviembre 2020</i>	<i>Vol 2. Borrador 1.</i>	<i>Abidán Brito Clavijo</i>	<i>Segundo borrador del sistema (Sprint 2).</i>