Desarrollo de un sistema para la gestión y la supervisión de personas con capacidades reducidas mediante sensores

Trabajo de Fin de Grado

INGENIERÍA INFORMÁTICA

Memoria

Julio de 2022

Autor

Miguel Carrasco Bueno

Tutor

Iván Álvarez Navia

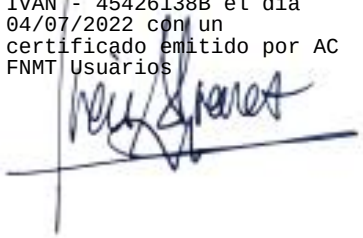
D. Iván Álvarez Navia profesor del Departamento de Informática y Automática de la Universidad de Salamanca.

CERTIFICA:

Que el trabajo titulado “Desarrollo de un sistema para la gestión y la supervisión de personas con capacidades reducidas mediante sensores” ha sido realizado por D. Miguel Carrasco Bueno, con DNI 70964170Q, y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado en Ingeniería Informática de esta Universidad.

Y para que así conste a todos los efectos oportunos.

Firmado por ALVAREZ NAVIA IVAN - 45426138B el día 04/07/2022 con un certificado emitido por AC FNMT Usuarios



En Salamanca, a 4 de Julio de 2022

D. Iván Álvarez Navia

Dpto. Informática y Automática

Universidad de Salamanca

Agradecimientos

Transmitir mi más sincero agradecimiento a todos aquellos que me han ayudado a lo largo de esta etapa y han aportado su granito de arena en el proyecto.

En primer lugar, a mi tutor, Iván Álvarez Navia, por su indispensable ayuda tanto en la planificación, desarrollo y por las detalladas correcciones de este Trabajo de Fin de Grado.

En segundo lugar, a mi familia, en especial a mis padres Marisa y Chuchi, y a mi hermano Carlos, por su incansable soporte, constante ayuda y por acompañarme durante el desarrollo del proyecto. También agradecer a mi tío Marcelino Zazo por iniciarme en el mundo de la informática, estando siempre pendiente y dispuesto a resolverme cualquier duda.

A mis amigos, en concreto a Rosa, Roberto y Aitor por ser un apoyo durante estos años y estar dispuestos a escucharme siempre, guiándome cuando tenía dudas.

A David y José Muñoz por acompañarme todos estos años.

También, expresar mi más sentido agradecimiento a FabLab, de la USAL, en concreto al técnico Juan Carlos Andaluz Delgado, por su ayuda tanto en el planteamiento como en la elaboración de la maqueta.

Agradecer también a ESaLAB, de la USAL, y en concreto al Dr. Gabriel Villarrubia González por permitirme emplear sus instalaciones para desarrollar elementos de la maqueta y a Sergio García González, becario de investigación, por asistirme en este proceso.

A todos ellos, mil gracias.

Resumen

El trabajo de fin de grado que se presenta a continuación tiene como punto central la creación de un sistema que permita gestionar y supervisar a personas con capacidades reducidas usando una serie de sensores.

El proyecto está diseñado para ser instalado tanto para un único usuario como para ser utilizado en grandes residencias. En ningún momento se deja de respetar la privacidad del usuario, únicamente se informa de los movimientos que puedan resultar potencialmente peligrosos, como son caídas o salidas de la estancia fuera de un horario limitado.

Además de estos controles se implanta sistemas para controlar las luces de la habitación. Pese a que se han elegido estas características, el proyecto permite en todo momento expandir sus límites, permitiendo controlar otro tipo de información como temperatura, humedad o ruido.

Uno de los objetivos principales del trabajo es demostrar que se puede implementar un sistema fácilmente escalable, de coste reducido y con componentes que todas las personas puedan conseguir sin dificultad. Además de intentar mejorar la calidad de vida tanto de los usuarios directos como de las personas responsables. Buscando la no sustitución de personal y sirviendo como una herramienta de apoyo únicamente.

Para poder llevar este objetivo a cabo se ha empleado la plataforma de Google Cloud IoT Core. Este servicio permite controlar cientos de sensores y dispositivos por un coste reducido o prácticamente nulo en muchos casos.

El último objetivo se completa usando sensores y tarjetas Arduino que presentan un coste asumible y que pueden ser adquiridos fácilmente. Es decir, no se presenta hardware propietario. Como se comentó anteriormente y dado que este proyecto es una prueba de concepto, puede ser fácilmente ampliado y mejorado.

Complementando este conjunto de sensores, que nos permitirán controlar una o varias estancias, se ha diseñado e implementado un sensor inalámbrico que permite la detección de caídas en cualquier lugar, además de funcionar como botón del pánico. El único requerimiento que tiene este dispositivo, y por ende el proyecto, es la necesidad de una conexión a internet mediante WIFI. Existiendo esta conexión, tanto los sensores como el módulo inalámbrico funcionarán en cualquier lugar.

Por último, para dotar de funcionalidad a todos estos datos que recogen los sensores, se ha diseñado una plataforma web utilizando HTML, CSS y JavaScript. Al igual que los elementos hardware del proyecto, esta plataforma podrá ser accedida desde cualquier parte del mundo. Esto se debe a que los datos que recogen los sensores son almacenados en la base de datos de Google, FireStore.

Abstract

The dissertation presented below focuses on the creation of a system to manage and monitor people with reduced capacities using a series of sensors.

The project is designed to be implemented both for a single user and for large residences. At no time is the user's privacy not respected, only potentially dangerous movements are reported, such as falls or leaving the room outside a limited timetable.

In addition to these controls, systems are implemented to control the lights in the room. Although these features have been chosen, the project allows at any time to expand its limits. Allowing the control of other types of information such as temperature, humidity, or noise.

One of the main objectives of the work is to demonstrate that a system can be easily scalable, low cost and with components that everyone can get without difficulty. In addition to trying to improve the quality of life of both direct users and responsible persons. Seeking the non-substitution of personnel and serving as a support tool only.

In order to achieve this goal, the Google Cloud IoT Core platform has been used. This service allows to control hundreds of sensors and devices for a reduced or practically zero cost in many cases.

The last objective is completed using sensors and Arduino boards that present a reduced cost and can be easily acquired. In other words, no proprietary hardware is presented. As commented above and since this project is a proof of concept, it can be easily extended and improved.

Complementing this set of sensors, which will allow us to control one or more rooms, we have designed and implemented a wireless sensor that allows the detection of falls anywhere, in addition to functioning as a panic button.

The only requirement of this device and therefore the project is the need for an internet connection via WIFI. With this connection, both the sensors and the wireless module will work anywhere.

Finally, to provide functionality to all the data collected by the sensors, a web platform has been designed using HTML, CSS and JavaScript. Like the hardware elements of the project, this platform can be accessed from anywhere in the world. This is because the data collected by the sensors is stored in Google's FireStore database.

Tabla de contenido

Tabla de figuras

Índice de tablas

1. Introducción

Según el INE [1] (Instituto Nacional de Estadística), la población española está derivando a una pirámide de población regresiva que veremos más adelante (Figura 1). Hablamos del tipo de pirámide que caracteriza a los países ricos, los cuales presentan, y como sabemos muy bien, una tasa de natalidad muy baja en contraste con la que presentan los países en desarrollo y los subdesarrollados. De la misma forma, al poseer una esperanza de vida muy alta, la zona elevada de la distribución, donde se concentra la población más envejecida, presenta una gran densidad [2].

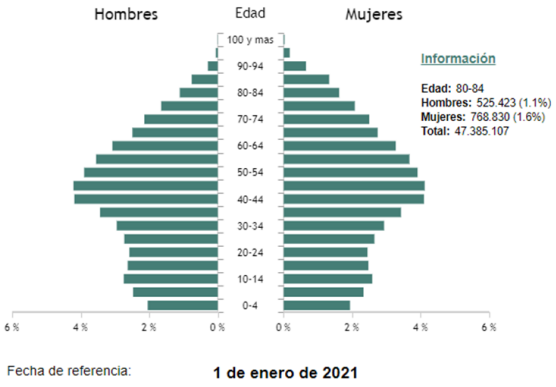


Figura 1 Pirámide poblacional española en 2021

Esta inversión en la pirámide es debida, fundamentalmente, a que la tasa de natalidad en los últimos 20-30 años ha menguado en gran medida comparada con años anteriores como vemos en la Figura 2.

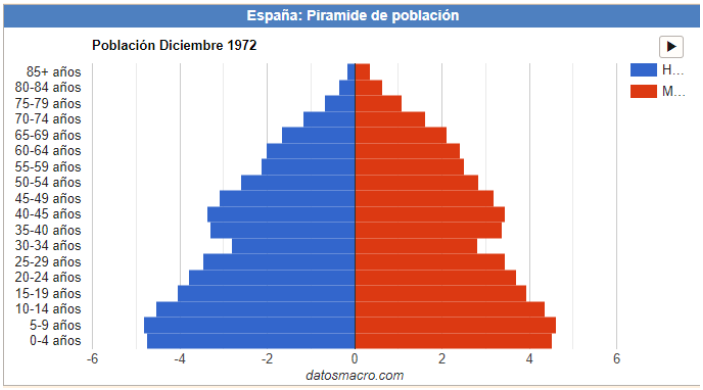


Figura 2 Pirámide poblacional española en 1972

Este hecho se suma a que, en la actualidad, gracias a los grandes avances que ha habido en diversos campos como la medicina, ha permitido que la esperanza de vida [3] de la población española haya aumentado en más de 10 años en la última mitad de siglo.

Dada esta situación en España hay cada vez una mayor cantidad de personas que pertenecen al grupo denominado de la “tercera edad”, mayores de 64 años, por contraposición el porcentaje de jóvenes

comprendidos entre los 0 y 14 años, que están en mínimo histórico de acuerdo con los últimos informes del INE.

Toda esta situación lleva al mayor uso de residencias, o trabajadores especializados en cuidado de personas mayores y dependientes, por parte de los españoles. La primera modalidad presenta un cuidado indirecto de la persona, con la consiguiente reducción de costes, si el cuidado se realiza durante las 24 horas, en comparación con la contratación privada de un cuidador. La segunda opción representa un cuidado más personal, pero por contraposición más costosa. Es por lo que los españoles, por norma general, suelen elegir residencias, tanto privadas como públicas, para ingresar en ellas a las personas con dependencia. Esto conlleva a que las residencias, en muchos casos, se encuentren desbordadas teniendo pocos trabajadores para hacer frente a toda la tarea que tienen que realizar. Por esta razón el cuidado suele empeorar.

Si nos fijamos ahora en los datos que nos aporta el CSIC (Consejo Superior de Investigaciones Científicas) podemos encontrar los siguientes resultados [4]:

Tenemos un total de 5567 residencias, 3925 privadas y 1642 públicas, con un total de 384251 plazas residenciales, 281332 privadas y 102919 públicas, distribuidas de la siguiente manera por el territorio nacional [5] como se puede ver en la Figura 3.

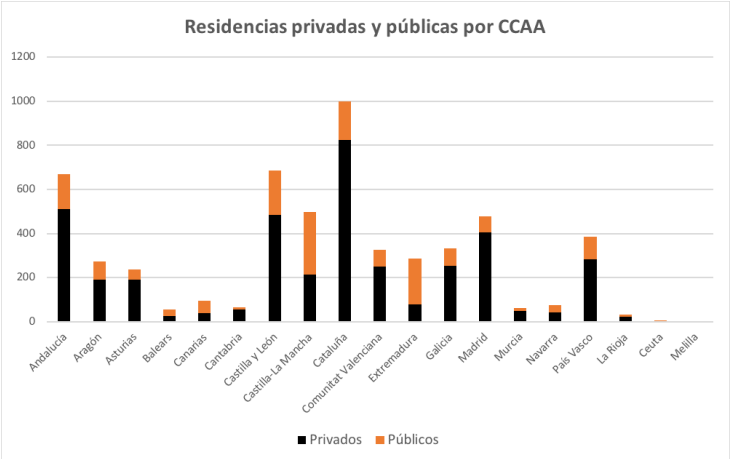


Figura 3 Distribución de residencias en el territorio español

Con estos datos podemos deducir que el número de residentes y residencias no dejará de aumentar en los años venideros. Por ello se pretende plantear un sistema de asistencia, tanto para un usuario privado como para una institución pública o privada.

Ante las dimensiones que está cobrando los problemas asociados al envejecimiento, y las necesidades de atención a personas dependientes, los avances tecnológicos y, más en concreto, en la informática,

contribuyen a plantear soluciones cada vez más sofisticadas y asequibles. La gran mayoría de la población mundial lleva consigo un dispositivo móvil capaz de realizar tareas que hace 10 años eran impensables. A su vez los sensores y dispositivos IoT (*Internet of Things*, Internet de las cosas) han sufrido fuertes evoluciones en los últimos años. Esto ha permitido que los costes se reduzcan y que todas las personas puedan acceder a ellos.

Es en este marco en el que surge la idea fundamental del presente TFG: Utilizar la tecnología existente, con unos costes asequibles, para mejorar la vida de las personas, tanto las dependientes como los cuidadores.

Este proyecto plantea crear un sistema que ayude de manera inmediata y sin retrasos a la persona responsable con la intención de solucionar un posible problema en el menor tiempo posible. Se propone un sistema que monitorice los cambios bruscos que se pueden dar dentro de una habitación y que, mediante el uso de tecnología, puedan ser rápidamente comunicados. En resumen: Un servicio de **coste reducido, instantáneo, fiable y escalable**.

- **De coste reducido**, para ser implementado tanto en un hogar, como en una residencia con gran cantidad de habitaciones.
- **Instantáneo,** ya que es de suma importancia conocer de manera rápida si se ha producido una situación con posibilidad de ser de gravedad, para intentar reducir la importancia y auxiliar con la mayor brevedad posible.
- **Fiable**, ya que debe dar la información necesaria, en el momento en el que se produzca la situación indeseada, se tiene que disponer de sensores o sistemas capaces de distinguir entre falsos positivos y negativos.
- **Escalable**, ya que, como se comentó con anterioridad, es importante que sirva, tanto para supervisar a un único individuo, como a varios de pacientes en una gran residencia.

Para llevar a cabo el proyecto se han seleccionado sensores que permiten medir y controlar datos y situaciones, obteniendo información del entorno para alertar al responsable cuando esos datos arrojen una situación de peligro. Esta información es controlada por tarjetas Arduino que permiten el envío a servicios en la nube como se trata de las plataformas Google Cloud, Microsoft Azure o Amazon Web Services. Posteriormente estos datos deben ser procesados para su recogida mediante una plataforma que permita ver los datos de una manera clara y visual. Para que cualquier persona sea capaz de distinguir que situación ha ocurrido.

Se van a usar los siguientes dispositivos para el desarrollo de este proyecto, exponiéndose otros más adelante por los que puedan ser sustituidos de cara a la implementación en una habitación o residencia real.

Entre los dispositivos encontramos:

- Tarjeta ESP-8266: Dispositivo encargado de enviar la información que recoge de los sensores a los servicios de Google Cloud. Elegida por sus características (dispone de varios puertos digitales, así como conexión WIFI).
- Tarjeta Arduino Uno: Encargado de obtener información de los sensores y enviarla mediante comunicación serie a un dispositivo con capacidad para enviarlo a la nube.
- Sensor de presencia: Encargado de controlar si una persona se encuentra en una posición determinada. Permite controlar por ejemplo si ha cruzado la puerta o si se ha caído.
- Sensor de luminosidad / fotorresistencia: Elemento encargado de controlar los niveles de luminosidad que haya tanto en una habitación como en un cuarto de baño. Con la intención de alertar de un uso inadecuado o prologando.
- Sensor de fuerza/presión: Dispositivo destinado a comprobar si una persona se encuentra en una estancia determinada o usando un elemento de la habitación.
- Giroscopio + Botón del pánico: Módulo inalámbrico encargado de controlar posibles situaciones que requieran atención inmediata, como es una caída o el pulsado del botón del pánico.

A continuación, se presenta una descripción de los capítulos que constituyen la presente memoria, además de los anexos que la acompañan:

Capítulo 2. Objetivos: Se describirán los principales objetivos del proyecto. Indicando tanto los funcionales del sistema como los personales del alumno.

Capítulo 3. Conceptos teóricos: Se detallarán los distintos datos e información recopilada antes, durante y después del desarrollo del proyecto. Así como las nociones de conocimiento relacionadas con los sensores y herramientas empleadas.

Capítulo 4. Técnicas y herramientas: Se indicará las herramientas y hardware empleado. Se realizará un hincapié en los distintos modelos de sensores y equipamientos disponibles, justificando la elección de los usados en el proyecto.

Capítulo 5. Aspectos relevantes del desarrollo. En este apartado se dará una visión global del proyecto, así como los problemas y limitaciones encontrados. Para clarificar el proyecto se mostrará un mapa de como el dato se transmite desde que se obtiene en los sensores hasta que se muestra en la pantalla del ordenador.

Capítulo 6. Descripción funcional: Se realizará una demostración del funcionamiento del sistema, así como los resultados del desarrollo.

Capítulo 7. Conclusiones y líneas de trabajo futuro: En este apartado se indicarán los resultados y conclusiones a las que se ha llegado tras el trabajo y realización del proyecto. También se presentarán los distintos desarrollos posibles con el proyecto, indicando que cambios se realizarían para profesionalizar y mejorar el producto.

Glosario y bibliografía. Se explicarán conceptos técnicos usados durante la memoria. Así como se presentarán las fuentes empleadas para el desarrollo del proyecto.

Anexos. En estos apartados se mostrará de manera detallada algunos aspectos que no se han comentado en la memoria. Entre estos documentos podemos encontrar:

Anexo I. Plan del proyecto software: Este documento se realizará una descripción de la estimación de costes y una planificación temporal del proyecto. Recogiendo de manera gráfica todas las actividades que se van a llevar a cabo en el proyecto.

Anexo II. Especificación de requisitos del software: El objetivo de este anexo es recoger todos los requisitos del sistema software a construir. Se mostrarán tanto los de información, funcionales y no funcionales.

Anexo III. Especificación de diseño: Generalmente este apartado está destinado a describir el diseño de datos, arquitectónico, de la interfaz y procedimental. En el caso de este proyecto se ha destinado a la especificación de elementos hardware además de software y de la disposición de los datos.

Anexo IV. Documentación técnica de programación: En este apartado se describirán las bibliotecas empleadas en el proyecto. También se dedicará un capítulo al código fuente de la aplicación. Para este apartado se empleará pseudocódigo para facilitar la legibilidad.

Anexo V. Manuales de usuario: En este anexo se recogen todos los tutoriales, manuales y guías de usuario que sean necesarios para el correcto manejo de la aplicación. En el caso de este proyecto se describirán también las recomendaciones para realizar una instalación adecuada de los sensores en una explotación particular.

2. Objetivos

Antes del desarrollo del proyecto se presentan una serie de objetivos funcionales y personales que se describirán a continuación.

2.1 Objetivos funcionales del proyecto

El objetivo principal del proyecto es crear un sistema que permita la supervisión adecuada de una persona dependiente dentro de una habitación, de manera que se pueda facilitar la tarea al cuidador o responsable, empleando tecnología y hardware disponible para todos los usuarios y de coste reducido.

De este objetivo principal podemos encontrar una serie de objetivos funcionales que se buscan cumplir en el desarrollo del proyecto:

- Ayuda: El sistema nace con la idea de ayudar tanto a las personas dependientes como a sus cuidadores en el día a día. Para mejorar la calidad de vida y hacer que situaciones potencialmente peligrosas reduzcan su gravedad.
- No sustitución de personal: Otro de los pilares del proyecto y estrechamente relacionado con el anterior, es el de crear un sistema de apoyo, y no de sustitución de atención de un responsable. El sistema asistirá y nunca será empleado como remplazo de la atención personalizada que una persona puede realizar. Este hecho puede suponer una mejora de la calidad de vida tanto del paciente como del responsable, pero no supone un cambio o mejora en la rentabilidad.
- Coste reducido: Otro de los objetivos principales es mantener un coste reducido en la implementación del sistema. Para ello se usarán dispositivos comunes, fácilmente adquiribles y que, en esencia, presenten un precio suficientemente bajo, si se implementa tanto para una única persona, como para varias. Además, se buscará que el software necesario para hacer funcionar el proyecto tampoco implique un coste para los usuarios de este.
- Velocidad de respuesta: Dadas las situaciones que pueden desarrollarse dentro de una habitación, es importante que los datos se muestren de manera rápida con el objetivo de resolver cuanto antes situaciones potencialmente peligrosas.
- Integridad y gestión de los datos: Otro de los objetivos del proyecto es conseguir que los datos e información obtenida por los sensores sea lo más segura posible. Además, se buscará que esta información sea rápida y fácilmente accesible por el responsable.

- Escalabilidad: Se pretende crear un sistema que funcione de la misma manera tanto para una única habitación como para varias. Para ello se buscará usar un sistema que permite controlar y centralizar toda la información.
- Facilidad de uso: Se buscará presentar al usuario de un sistema que sea fácilmente utilizable y accesible. Mostrando la información de una manera simple y visual. Indicando claramente las distintas situaciones que pueden darse.
- Robustez ante distintas situaciones: Como objetivo importante se buscará crear un sistema que reporte adecuadamente de las situaciones que se desarrollen dentro de una habitación. Indicando claramente cuando se producen, sin generar casos de falsos positivos o negativos. Así como reportar adecuadamente de las que sean situaciones reales sin inundar al usuario con información.

2.2 Objetivos personales

Entre los objetivos que como alumno me planteo es la adquisición de nuevos conocimientos, descubrir nuevas tecnologías y hardware que hasta el momento no he explorado.

También intentaré aprender a realizar las tareas de desarrollo hardware y software desde la concepción de la idea, pasando por el desarrollo y finalización del producto.

Como parte de este proyecto también se buscará poner en práctica todos los conocimientos adquiridos durante los cuatro años de formación académica en la universidad. También se intentará ampliar estos conocimientos y aprender las diferentes tecnologías existentes, que pueden servir para desarrollar y llevar a cabo diversidad de proyectos.

Otro objetivo importante es que el sistema complete adecuadamente los objetivos planteados en su concepción, intentando que el proyecto funcione acorde a las características indicadas anteriormente.

Para finalizar, quiero recalcar que el nacimiento de la idea se debe a intentar solucionar un problema existente y que está presente en la vida de muchas personas. Es claro que este proyecto no tiene el alcance para solucionar todos esos problemas. Pero se intentará presentar una posible solución con unas ciertas características (escalable, rápido y de coste reducido) que puede llevar a proyectos posteriores a mejorar la calidad de vida de las personas que lo necesitan.

3. Conceptos teóricos

A continuación, se discutirán aspectos y conceptos de importancia del proyecto. Entre estos podremos encontrar los datos obtenidos para comprobar la viabilidad del proyecto, que sensores se han utilizado, que protocolos se usan para describir cada situación y como se envía la información correspondiente a la nube de Google.

3.1 Datos

Como ya se adelantó en anteriores apartados, si nos regimos por la evolución de la pirámide poblacional española, y de casi cualquier país europeo, en los últimos 50 años se ha producido una gran transformación sociológica. Las personas viven más años gracias a los avances en medicina y cuidados, si atendemos al siguiente gráfico (Figura 4):

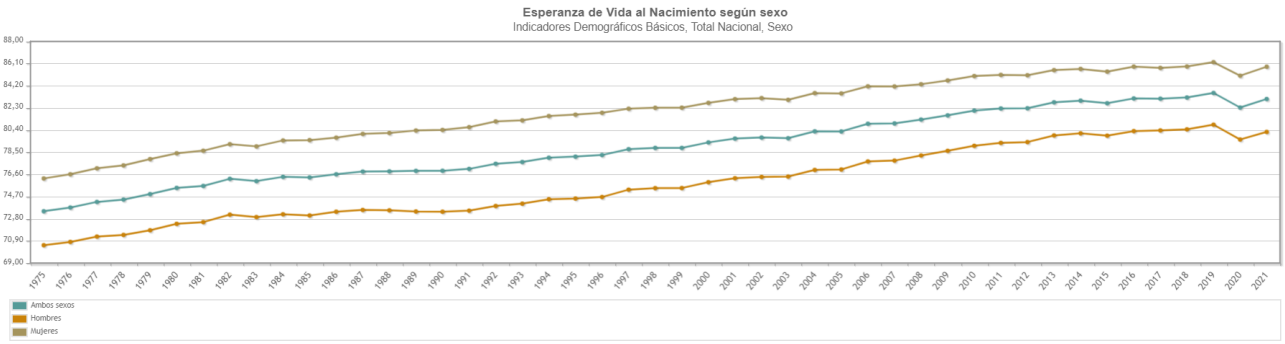


Figura 4 Comparativa de esperanza de vida en España

Podemos observar como la esperanza de vida ha aumentado varios puntos en los últimos años. Este dato nos indica que las personas viven más, de media, por lo que requerirán más cuidados al dejar de ser autónomos, de manera general, en los últimos momentos de sus vidas.

También debemos tener en cuenta que según la Figura 5 podemos comprobar como la tasa de natalidad se ha reducido en gran medida en los últimos años. Esto nos lleva a la siguiente conclusión: Cada vez hay menos personas jóvenes que puedan hacerse cargo de las personas mayores o que tengan algún tipo de dependencia.

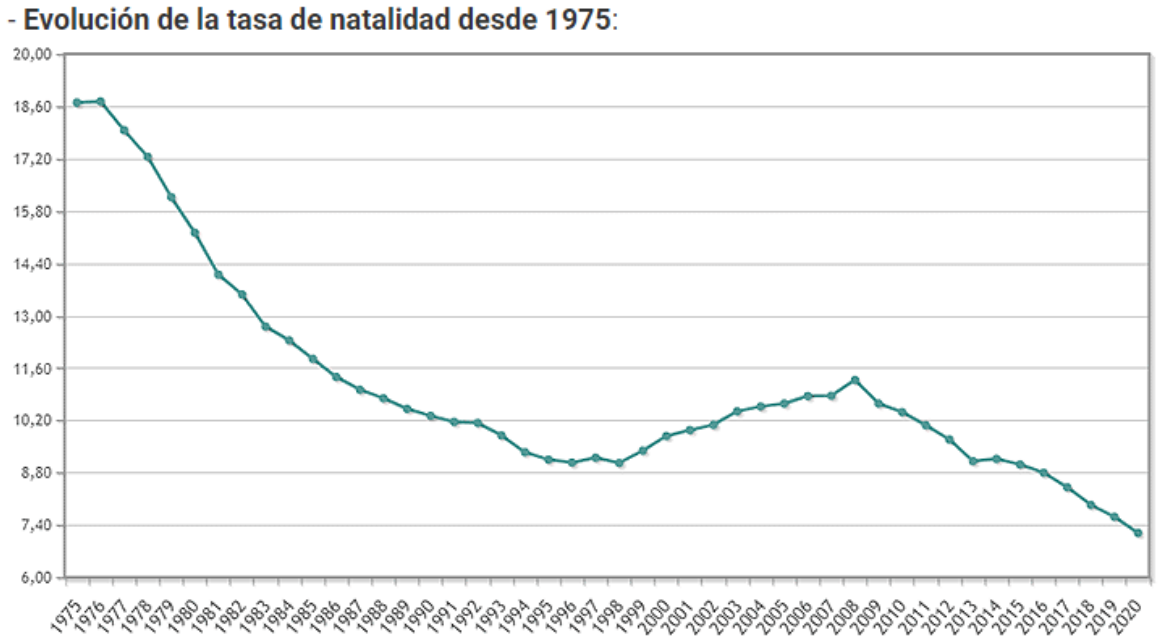


Figura 5 Tasa de Natalidad (Nacidos por mil habitantes)

Por esta razón cada vez es más común internar a la persona dependiente en una residencia para que este tenga un mejor cuidado. Otra alternativa es la contratación de una persona que esté durante una parte del día a cargo de la persona.

Como podemos ver en la Figura 6, las residencias cuentan cada vez con mayor cantidad de internos. Si a esto le sumamos que, en muchos de los casos, los puestos de trabajo se han reducido por parte de las residencias con la idea de abaratar costes, podemos concluir que las personas internas, pese a tener un control y un cuidado, en muchas ocasiones no están recibiendo el trato más adecuado.

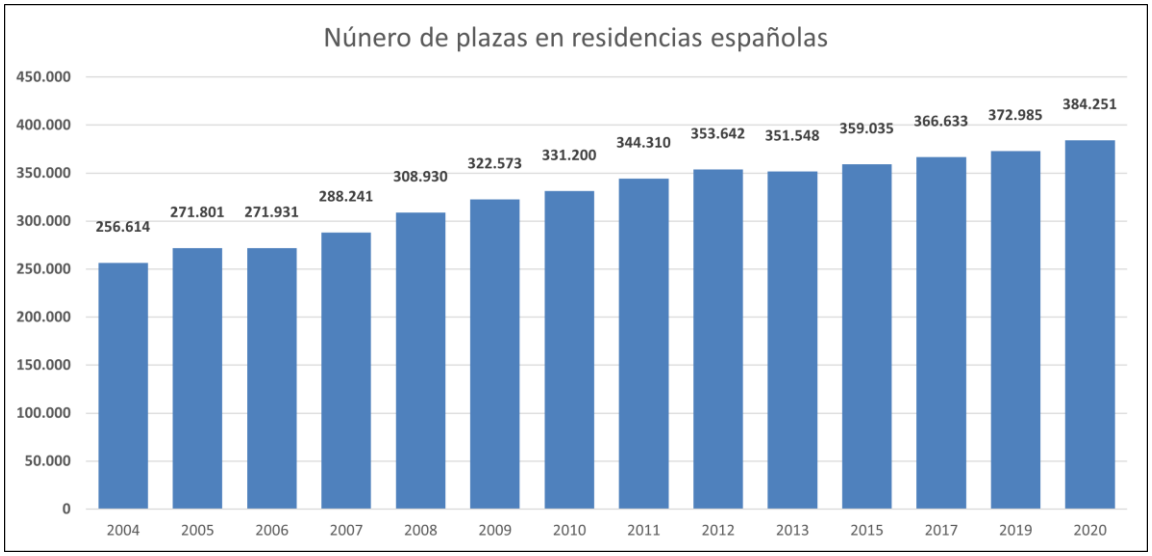


Figura 6 Número de plazas de residencias españolas entre 2004 y 2020

Por todas estas razones se plantea la idea de crear un sistema de asistencia y ayuda para que las personas dependientes tengan una mejor calidad de vida, haciendo que ciertas situaciones sean indicadas a la persona que supervisa de manera instantánea.

3.2 Descripción del sistema

Antes de comenzar a explicar en detalle los sensores, dispositivos y tecnologías empleadas se van a explicar los elementos intervinientes y el funcionamiento del sistema empleando el diagrama de la Figura 7.

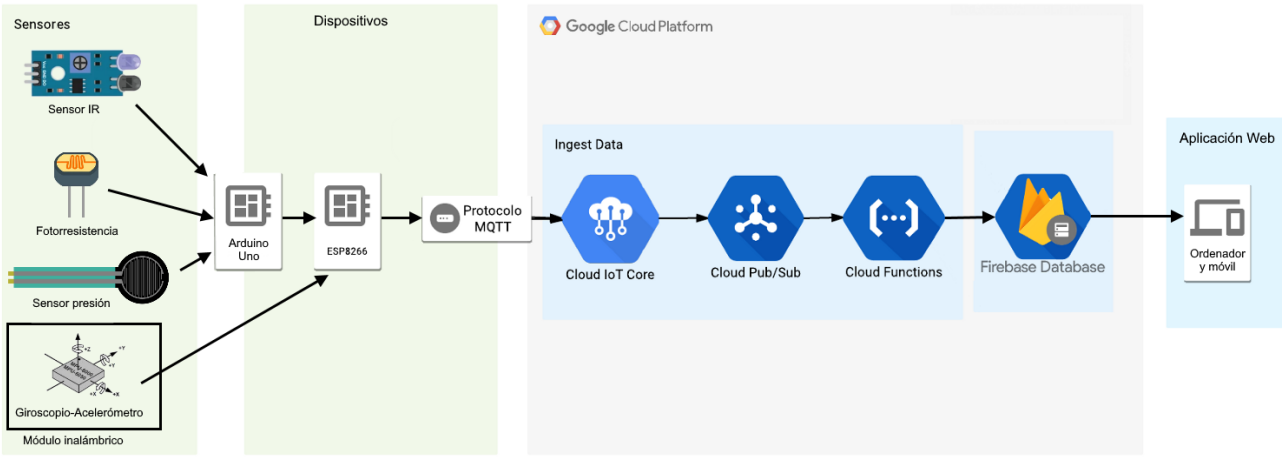


Figura 7 Descripción del sistema

El origen de los datos proviene de los sensores, encargándose de recopilar información del medio. Estos datos serán procesados mediante los dispositivos ESP8266 y Arduino Uno, en el caso del módulo inalámbrico solamente serán procesados por el módulo ESP8266. Estas tarjetas realizarán las comprobaciones de los datos recibidos y formarán mensajes para ser enviados mediante MQTT (*Message Queuing Telemetry Transport*) a la plataforma de Google Cloud.

Una vez que lleguen los mensajes a Cloud IoT Core, que hace función de *broker* de MQTT, serán distribuidos al servicio de colas de Pub/Sub. Cuando este sistema redirija los mensajes, a las suscripciones indicadas, el servicio de Cloud Functions se encargará de obtener la información que recopilaron los sensores y almacenarlo en FireStore (servicio interno de Firebase).

Por último, estos datos almacenados en la base de datos no relacional de FireStore serán recopilados mediante código JavaScript y mostrados al cliente en un panel de control web. Desde esta página el responsable contará con la información necesaria para actuar de la manera más ágil y eficiente posible.

3.3 Sensores

Como se comentó en apartados anteriores, se van a usar diversos sensores para obtener información del medio y así poder tomar decisiones acordes a lo que esté sucediendo. Estos sensores han sido escogidos con la idea de plantear una prueba de concepto usando una habitación prototipo de tamaño reducido. Existen una gran variedad de sensores que mejoran las características de los elegidos, pero para esta prueba no son requeridos. De todos modos, aparte de comentar los aspectos teóricos relacionados con estos sensores, se van a comentar alternativas a estos.

3.3.1 Fotorresistencia

También conocido como LDR (*Light-Dependent Resistor*, Resistencia dependiente de la luz [6]) es un componente electrónico cuya resistencia es modificada con la variación de magnitud de intensidad lumínica que incide sobre el mismo. Generalmente está compuesta de una célula fotorreceptora, con dos patillas para la conexión.

La base de funcionamiento de una fotorresistencia está en el componente principal del que se encuentra formada, sulfuro de cadmio, CdS. Este semiconductor tiene la capacidad de variar su resistencia eléctrica dependiendo de la cantidad de luz incidente. Cuanta mayor intensidad de luz incide sobre la célula fotorreceptora, más baja es la resistencia que presenta, es decir, pasará más corriente por el circuito al que se encuentre conectado.

Existen dos tipos de resistencia dependiendo de la manera en la que se polarizan. Las lineales, llamadas fotodiodos, que polarizan a la inversa de la fuente donde se conecte, y las no lineales, en las cuales la polaridad no dependerá de la fuente a la que se conecte. Por esto último se ha elegido las fotorresistencias no lineales.

Como se puede ver en la Figura 8 dependiendo de la cantidad de luminosidad, la fotorresistencia ofrecerá una mayor o menor resistencia. Generalmente, la resistencia con mucha luminosidad es de 100Ω, mientras que en situaciones de alta oscuridad ofrecerá una resistencia de 1MΩ.

Además, cabe destacar que el tiempo de respuesta, dependiendo de la aplicación para la que se use, puede resultar lento, en torno a 1 décima de segundo cuando se produce variación de luminosidad. Esta tecnología es de muy bajo coste y extremadamente sencilla. Por estas razones está presente en gran cantidad de dispositivos que podemos encontrar en nuestro día a día, como alarmas o sistemas de alumbrado. El uso de fotorresistencias presenta una serie de ventajas e inconvenientes. Entre las

ventajas podemos encontrar la facilidad de uso, el reducido coste y la capacidad de abarcar distintas superficies.

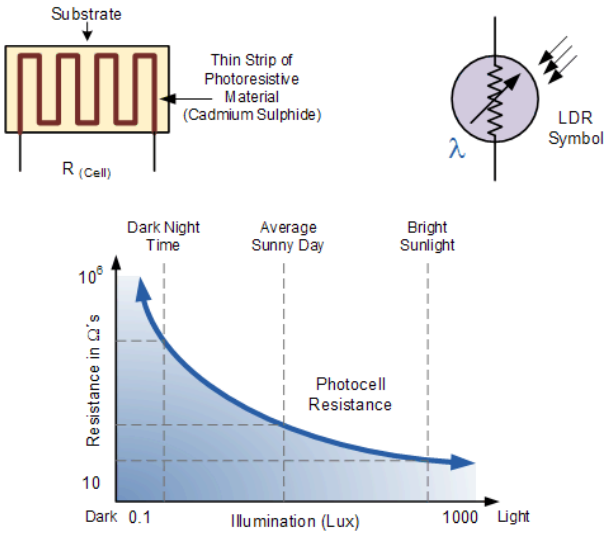


Figura 8 Fotorresistencia no lineal

Por contraposición debemos tener en cuenta las desventajas que presenta el uso de esta tecnología. Una de las más importantes, como se pudo observar en la Figura 8, es que la variación de la resistencia no es lineal, por lo que habrá que controlarla adecuadamente, y el tiempo de respuesta es lento. También habrá que elegir la resistencia adecuada dependiendo del tipo de radiación lumínica que se quiera medir, ya sea infrarrojo, espectro visible o ultravioleta. Por último, hay que tener en cuenta el efecto denominado histéresis. Según la Real Academia Española, es un "fenómeno por el que el estado de un material depende de su historia previa y que se manifiesta por el retraso del efecto sobre la causa que lo produce". Es decir, que si no existe ningún estímulo el material tenderá a conservar sus propiedades.

Teniendo en cuenta estas características se ha optado por elegir para la realización del proyecto una simple fotorresistencia no lineal. Similar a la que encontramos en la Figura 9.

Figura 9 Fotorresistencia empleada

Ahora demostraremos con un simple código el funcionamiento de la fotorresistencia.

Como podemos ver en la Figura 10, en la que se representa un circuito simplificado, vemos que el valor que devuelve la lectura analógica es menor cuanto tenemos poca luminosidad [7].

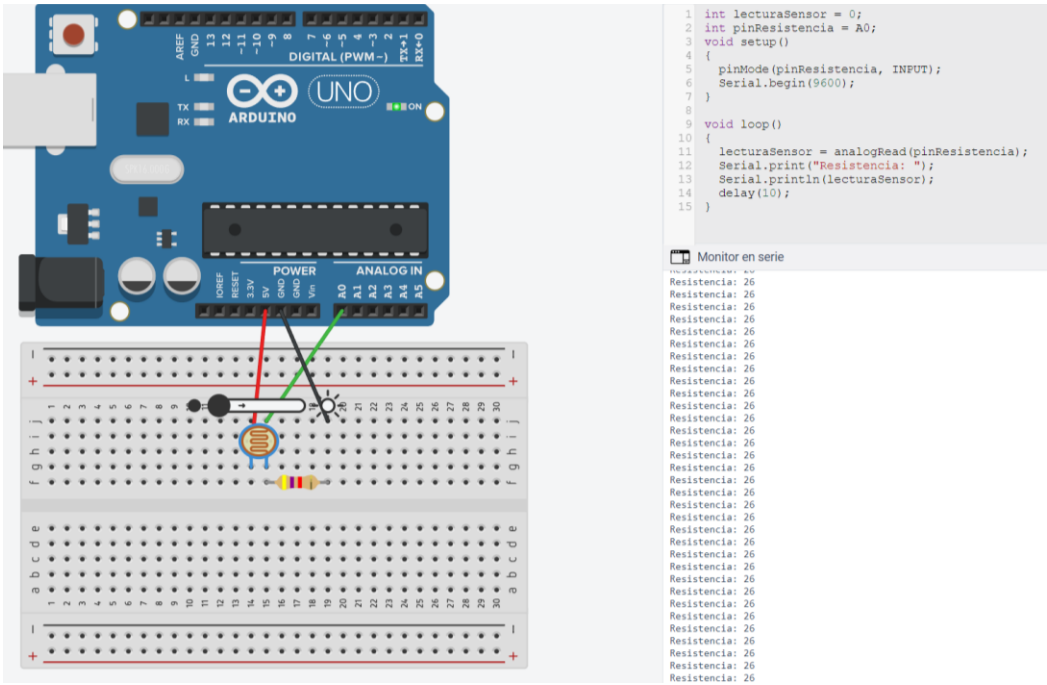


Figura 10 Ejemplo código fotorresistencia – Luminosidad reducida

Si aumentáramos el valor de luminosidad veremos como el valor de lectura se incrementa consecutivamente.

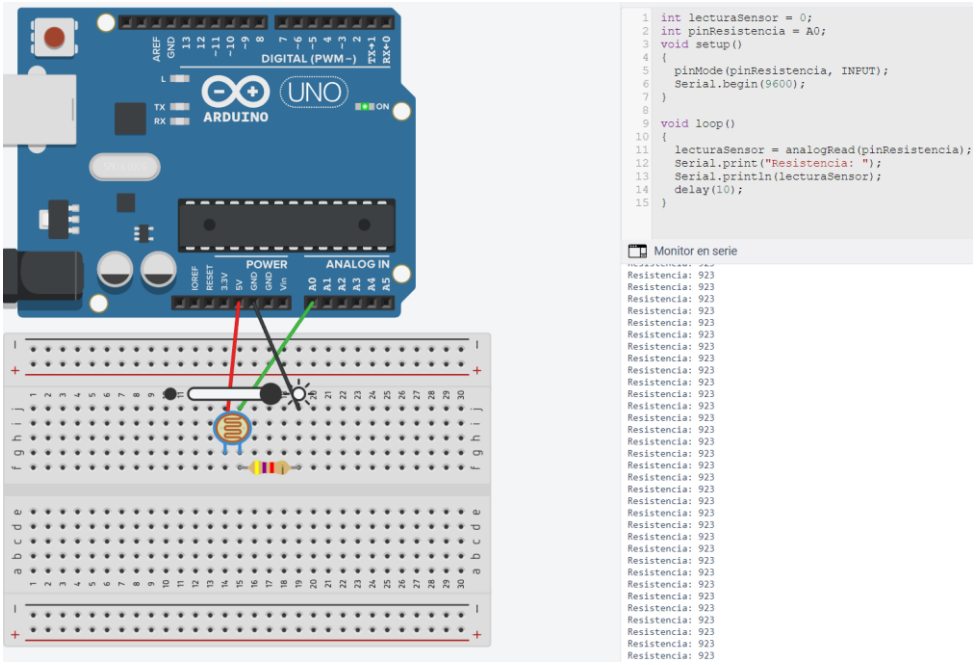


Figura 11 Ejemplo código fotorresistencia - Luminosidad elevada

Para entender esta diferencia de valores de lectura se debe explicar que los valores de lectura analógica en Arduino comprenden una horquilla de valores comprendidos entre 0 y 1023 [8]. Al cambiar el valor de luminosidad en la fotorresistencia obtenemos una lectura con distintos valores. En el caso del ejemplo empleado en la Figura 10 obtenemos el dato de 26, debido a la reducida luminosidad, mientras que en la Figura 11 es de 923. Estos valores representan la cantidad de corriente que deja pasar la fotorresistencia, o lo que es lo mismo, la resistencia que presenta al paso de corriente [9].

En cuanto a las alternativas que se podían haber usado el sensor de luz ambiental BH1750.

Figura 12 Sensor BH1750, alternativa a la fotorresistencia

Este sensor presenta mejoras como una resistencia al espectro infrarrojo, alta resolución de lectura y la habilidad de convertir los datos analógicos a digital sin usar ninguna parte externa. El principal problema, y la razón principal por la que no se plantea su uso para este proyecto, es su coste. Ya que, por lo general su precio es de diez veces mayor al de la fotorresistencia usada. Dado que esto es una prueba de concepto y no el producto final se ha optado por el módulo que representa unos menores costes.

A continuación, se van a detallar los aspectos más relevantes del sensor elegido:

- Nombre: SEN-09088.
- Fabricante: Sparkfun.
- URL – Ficha técnica: https://www.sparkfun.com/datasheets/Sensors/Imaging/SEN-09088-datasheet.pdf
- Precio: 1.40€ - 1.60€ unidad.

3.3.2 Giroscopio acelerómetro

Este sensor es un dispositivo capaz de medir y mantener la orientación y velocidad angular de un objeto [10]. Un acelerómetro convencional solo puede medir el desplazamiento lineal. Mientras que un giroscopio-acelerómetro puede medir tanto la inclinación como orientación lateral.

Como se ha dicho anteriormente, permiten obtener la velocidad angular, en grados por segundo. La velocidad angular es el cambio en el ángulo de rotación de un objeto por unidad de tiempo. Aparte de medir esta magnitud, son capaces de determinar el movimiento de un objeto. Normalmente el sistema completo consta de diversos acelerómetros. Dependiendo de la dirección existen tres tipos de mediciones de la velocidad angular:

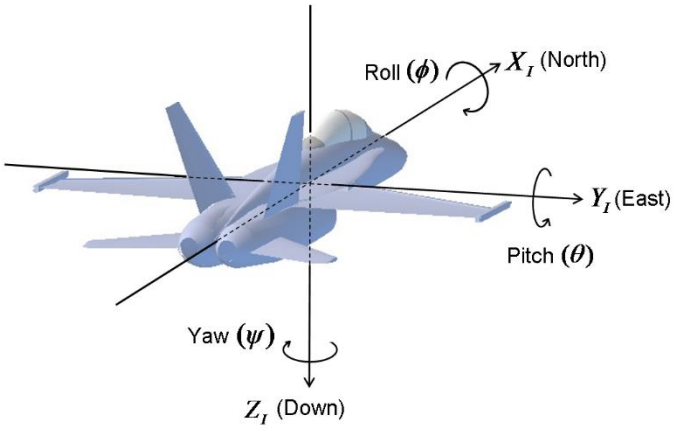


Figura 13 Representación de los distintas magnitudes de velocidad angular

El eje normal: Una rotación de guiñada es un movimiento alrededor del eje de guiñada de un cuerpo rígido que cambia la dirección hacia la que apunta, a la izquierda o a la derecha de su dirección de movimiento. La tasa de guiñada o velocidad de guiñada de un coche, avión, proyectil u otro cuerpo rígido es la velocidad angular de esta rotación, o tasa de cambio del ángulo de dirección cuando el avión está horizontal. Se mide en grados por segundo o en radianes por segundo.

El eje de cabeceo, también llamado eje transversal o lateral, tiene su origen en el centro de gravedad y está dirigido a la derecha, paralelo a una línea trazada de punta a punta del ala. El movimiento en torno a este eje se denomina cabeceo. Un movimiento de cabeceo positivo eleva el morro del avión y baja la cola. Los elevadores son el control principal del cabeceo.

El eje de alabeo, también conocido como eje longitudinal tiene su origen en el centro de gravedad y está dirigido hacia delante, paralelo a la línea de referencia del fuselaje. El movimiento en torno a

este eje se denomina balanceo. Un desplazamiento angular alrededor de eje se llama inclinación. Un movimiento de balanceo positivo eleva el ala izquierda y baja el ala derecha.

También hay que tener en cuenta el efecto Coriolis. Este es el efecto que se observa en un sistema de referencia en rotación cuando un cuerpo se encuentra en movimiento respecto de dicho sistema de referencia. Este efecto consiste en la existencia de una aceleración relativa del cuerpo en dicho sistema en rotación. Esta aceleración es siempre perpendicular al eje de rotación del sistema y a las componentes radial y tangencial de la velocidad del cuerpo.

Teniendo en cuenta lo expuesto antelación, no se explicarán en mayor detalle cómo funciona internamente un giroscopio. Se pasará entonces a detallar el funcionamiento del sensor elegido. Como se comentó con anterioridad este sensor se va a usar para detectar posibles caídas de los pacientes, de manera que vamos a explicar su funcionamiento.

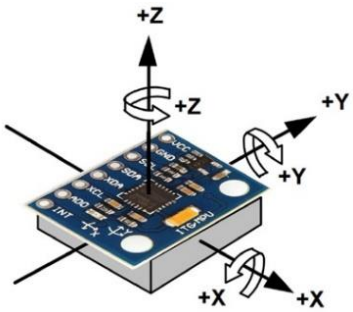


Figura 14 Sensor MPU6050 empleado en el proyecto

El sensor MPU6050 nos permite con una sola orden obtener tanto los valores de las tres dimensiones para el acelerómetro y el giroscopio. Una vez obtenidos los datos son necesarios calibrarlos. Con dichos datos obtendremos el vector amplitud del acelerómetro.

Generalmente en una caída, una persona sigue el siguiente comportamiento:

- Primeramente, existe una caída libre momentánea, con una reducción de la aceleración.
- Seguidamente, se da un gran pico de aceleración.
- Por último, se cambia la orientación en la que se encuentra. Ahora si esta nueva orientación es mantenida por una determinada cantidad de tiempo, se entiende que la persona se ha caído.

Si no se cumplen todas las condiciones anteriores no es posible detectar una caída, ya que un aumento de la aceleración indicaría una caída obteniendo gran cantidad de falsos positivos. Por ende, la barrera que discrimina si se ha producido una caída es que se haya generado una gran fuerza de aceleración partiendo de una mínima y además se haya cambiado la orientación. Es por esta razón por lo que es

completamente necesario el uso de un giroscopio, dado que sin él sería imposible comprobar la orientación de la persona.

Para la obtención de esta información se ha tomado como referencia el Trabajo de Fin de Grado "Desarrollo de un sistema de detección de caídas" [11].

El principal problema que presenta este enfoque es que es necesario que exista una lectura anterior a la caída para poder comprobar la diferencia en aceleración y orientación. Se precisan una serie de datos para detectarlo adecuadamente, es decir, para que la lectura sea precisa el giroscopio debe estar en funcionamiento con anterioridad.

En cuanto a posibles sensores que sustituyan al elegido, podemos encontrar el 10 DOF IMU de la Figura 15, que es un sensor de unidad de medición inercial. Estos emplean magnetómetros y sensores de presión, además de giroscopio y acelerómetros, para obtener unas medidas más exactas.

Además, permite medir temperatura y requiere mayor cantidad de energía para funcionar adecuadamente. Es usado, generalmente, para drones.

Pese a sus cualidades no ha sido elegido por su elevado coste. Dado que suele costar unas 10 o 20 veces más que el MPU6050, dependiendo del modelo que se elija.

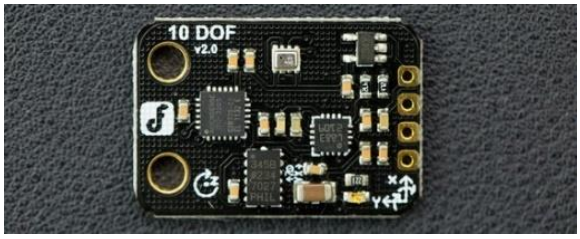


Figura 15 Sensor 10 DOF IMU, alternativa al sensor MPU6050

A continuación, se van a detallar los aspectos más relevantes del sensor seleccionado:

- Nombre: MPU-6050.
- Fabricante: TDK InvenSense.
- URL – Ficha técnica: https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf
- Precio: 4€ - 6€ unidad.

3.3.3 Sensor de presión

También conocido como FSR *(Force Sensitive Resistor,* Resistencia Sensible a la Fuerza [12]*)*. Son sensores de bajo coste y fácil uso. Son diseñados para detectar presión física y peso. Pese a que son capaces de medir la presión existente, no están destinados para medir la fuerza, solamente detecta si está siendo pulsado o no.

El principio físico detrás de estos sensores es similar al caso que se puede ver en las fotorresistencias. Básicamente, es una resistencia que varía dependiendo de la presión que está siendo aplicada sobre el mismo.

Generalmente el FSR está formado por 2 capas separadas por un espaciador, como se ve en la Figura 16. Cuanto más se presiona, más puntos del elemento activo tocan el semiconductor y eso hace que la resistencia baje como se puede apreciar en la Figura 17.

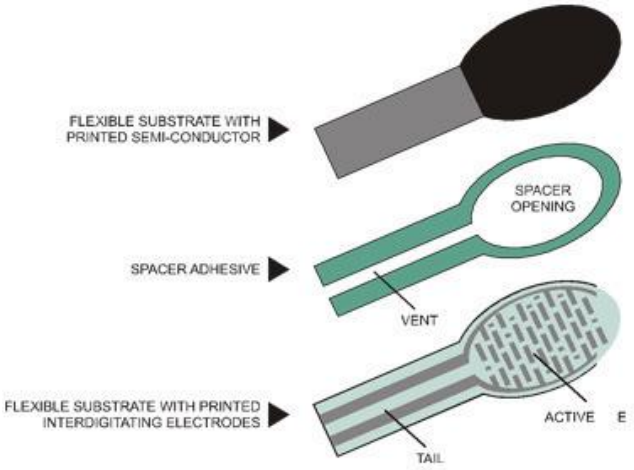


Figura 16 Composición de sensor de presión

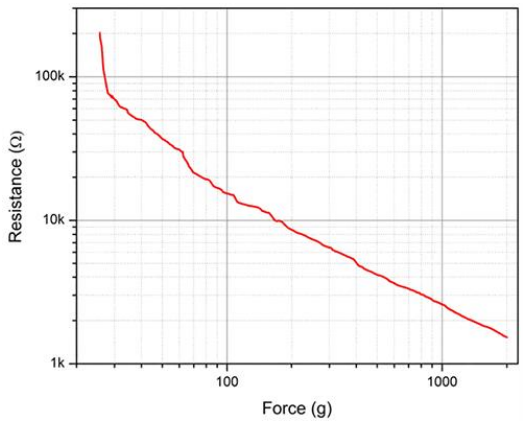


Figura 17 Variación de la resistencia dependiendo de la fuerza aplicada

Dado que este sensor, por tamaño y fuerza necesaria para usarlo adecuadamente, es difícil adaptarlo al proyecto que se plantea, se ha decidido crear uno. Para este desarrollo se ha empleado una lámina delgada de aluminio y un par de cables.

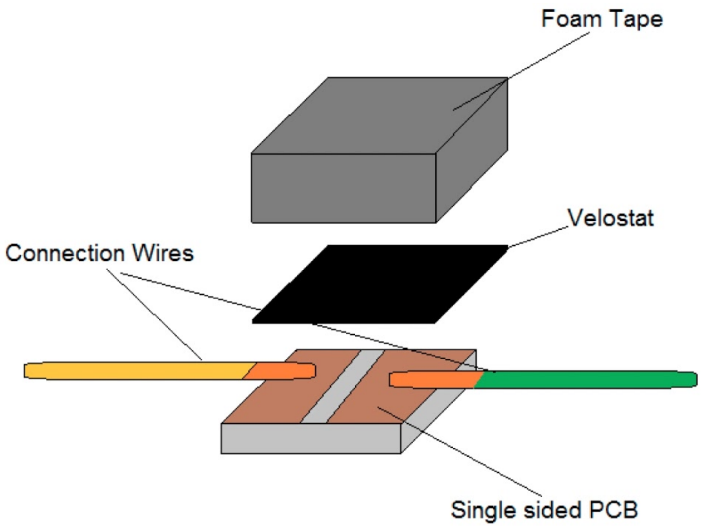


Figura 18 Esquema de la implementación realizada

El principio de funcionamiento es el mismo que el sensor anterior, teniendo en cuenta la gran reducción en coste y la modularidad que nos permite este diseño. Como podemos ver en la Figura 18, podemos usar el tamaño que sea necesario mientras los dos cables estén conectados de la manera indicada.

Existe un problema con este diseño, la obligación de realizar lecturas analógicas. Esto se debe fundamentalmente a que no tenemos ningún circuito que pueda convertir las lecturas en datos digitales. Por ello para simplificar el proceso a la tarjeta Arduino llegará un valor comprendido entre 0 y 1023 dependiendo de si hay o no contacto entre los dos cables.

El principio físico es simple: Si no existe conexión la lectura de datos que recibe la tarjeta es cercana a cero, y, por el contrario, si existe presión la lectura será cercana a 1023. Una vez afinado el diseño y el control mediante software, tenemos un sensor que ha sido diseñado y funciona de forma adecuada.

Hay que tener en cuenta que se ha decidido usar este planteamiento únicamente porque se trata de una prueba de concepto y la habitación del prototipo va a tener unas dimensiones reducidas. Si este proyecto fuera a implantarse en una habitación real se debería probar a ajustar el diseño o usar un sensor de presión como los mencionados con anterioridad. Otra opción es emplear el detector que se encuentra en las básculas digitales para determinar si una persona se encuentra sobre una superficie.

Este sistema permitiría individualizar más la supervisión, ya que se podría hacer control por peso, para así eliminar la posibilidad de lecturas erróneas.

También hay que indicar que se podría plantear el desarrollo o adquisición de un sensor por mapa de presión. Este dispositivo permite detectar donde se está desarrollando la presión y la forma del cuerpo que realiza la fuerza. Esto permitiría reconocer patrones, como disponerse en pie encima del sensor, tumbado o de rodillas como podemos ver en la Figura 19.

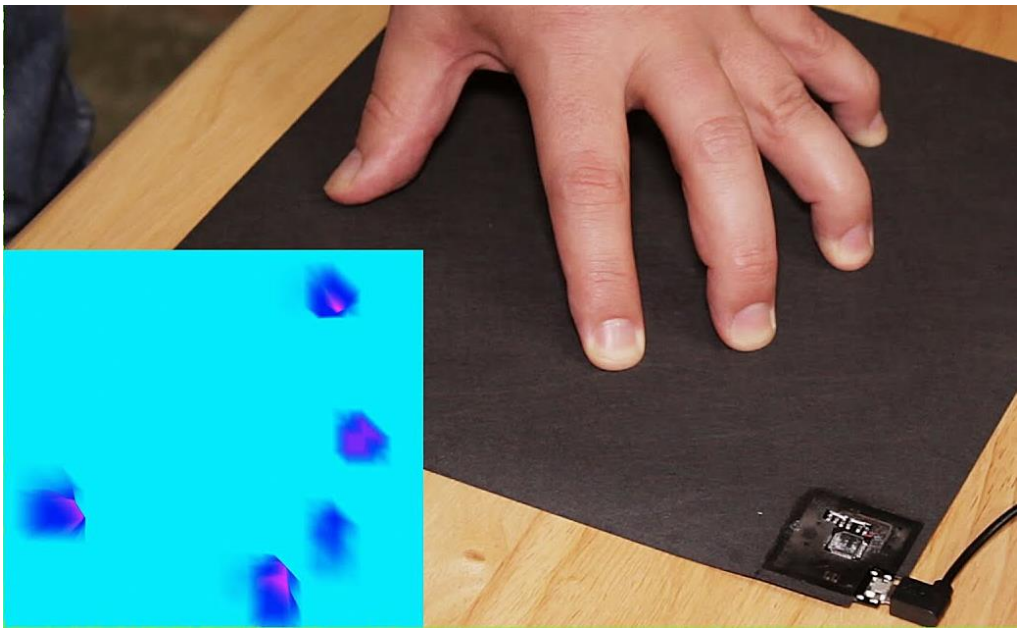


Figura 19 Ejemplo de Mapa de presión

Teniendo en cuenta que el factor monetario y la simplicidad son importantes, se seguirá desarrollando el proyecto con el sensor diseñado dadas las ventajas expuestas con anterioridad.

3.3.4 Sensor Infrarrojo

Esta tecnología [13] basada en la detección de las radiaciones fuera del espectro visible es usada en distintos campos diariamente. Por ejemplo, los mandos de televisión usan sensores infrarrojos que envían señales al receptor de la televisión.

Entre las ventajas principales de los sensores infrarrojos encontramos su reducido coste, tanto energético como económico, su simple diseño y funcionamiento.

Como se ha dicho anteriormente, la radiación empleada por estos dispositivos no es visible por el ojo humano. Generalmente las longitudes de onda se encuentran entre 0,7 µm 5 µm y 1000µm. El espectro de radiaciones infrarrojas suele dividirse en tres regiones: Infrarrojo cercano, medio y lejano.

La longitud de onda de la región del infrarrojo cercano va de 0,75 a 3µm, la del medio va de 3µm a 6µm y la del lejano es superior a 6µm.

Un sensor de infrarrojos o IR es un dispositivo electrónico que emite radiaciones para detectar algunos elementos del entorno. Un sensor IR puede medir el calor de un objeto, así como detectar el movimiento. Este dispositivo, funcionan como detectores, denominado sensor IR pasivo Figura 20. Normalmente, en el espectro infrarrojo, todos los objetos irradian alguna forma de radiación térmica.

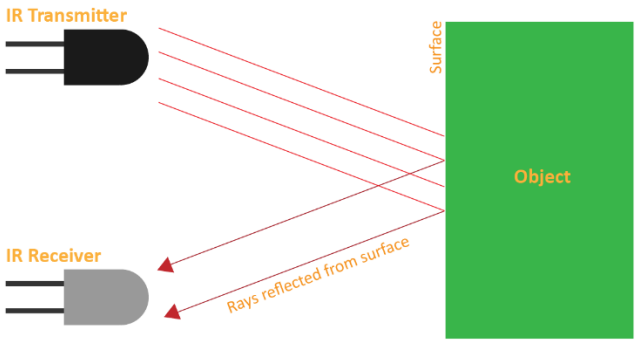


Figura 20 Ejemplo de emisor-receptor infrarrojo

El emisor es un LED IR (Diodo Emisor de Luz) y el detector es un fotodiodo IR que es sensible a la luz IR de la misma longitud de onda que la emitida por el LED IR. Cuando la luz IR incide sobre el fotodiodo, las resistencias y las tensiones de salida cambiarán en proporción a la magnitud de la luz IR recibida.

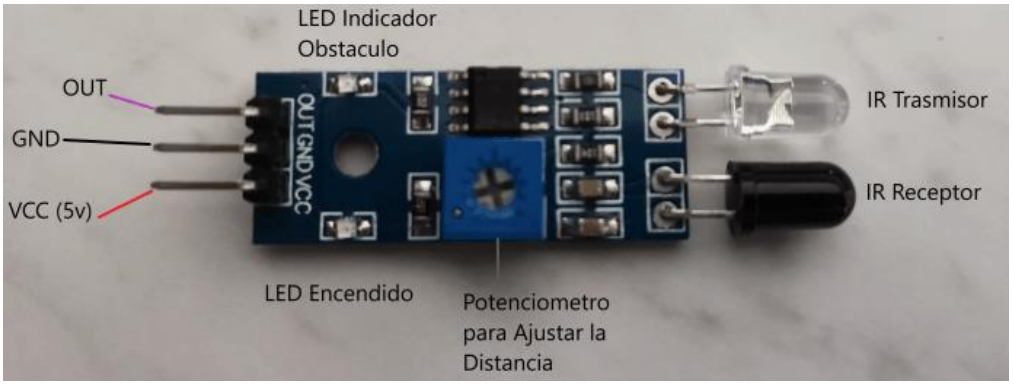


Figura 21 Diagrama del sensor utilizado

El LED IR es un tipo de transmisor que emite radiaciones IR. Este LED tiene un aspecto similar al de un LED estándar y la radiación que genera no es visible para el ojo humano. Los receptores de infrarrojos detectan la radiación Estos receptores de infrarrojos están disponibles en forma de fotodiodos. Los fotodiodos IR son diferentes a los habituales porque detectan simplemente la

radiación IR. Existen diferentes tipos de receptores de infrarrojos, principalmente en función de la tensión, la longitud de onda, el paquete, etc.

Una vez que el transmisor de infrarrojos genera la emisión, esta llega al objeto y parte de la señal se refleja hacia el receptor. La salida del emisor puede ser decidida por el receptor en función de la intensidad de la respuesta.

En cuanto a las ventajas encontramos que este sensor requiere de poca energía para funcionar, no es necesario que exista contacto con la superficie del cuerpo para detectarlo adecuadamente, además, no se ven afectados por la oxidación o corrosión y también presentan una alta inmunidad al ruido. En contraposición encontramos que es necesario que exista línea de visión entre el sensor y el objeto, su alcance es limitado y puede verse afectado por diversos factores como la niebla, la lluvia o la luz solar.

Existen dos tipos de sensores IR, como podemos ver en la Figura 22, tanto activos como pasivos.

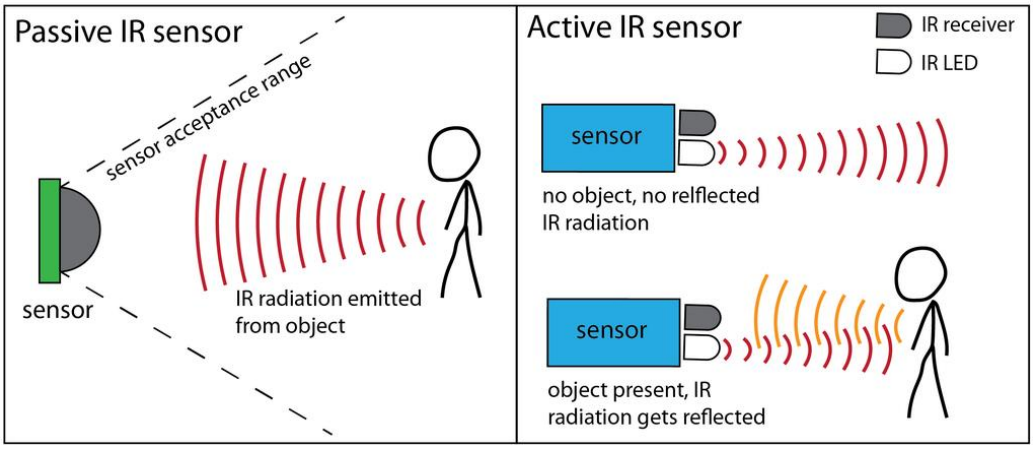


Figura 22 Diferencias entre sensor pasivo y activo

Los sensores IR pasivos son usados generalmente para detectar movimiento, mientras que los activos se usan para comprobar la ruptura de un haz de luz.

Una vez que se ha explicado el principio físico y el funcionamiento de estos sensores, os quedara detallar el empleado y las razones de su elección.

Como hemos visto anteriormente los sensores IR pasivos son sensores que nos permiten detectar presencia y movimiento de un cuerpo. Pero su principal desventaja y es que necesitan de un determinado espacio para funcionar. Como ya se ha indicado con anterioridad, se va a desarrollar un prototipo de una habitación con dimensiones reducidas. Por esta razón se ha descartado el uso de sensores pasivos como el HC-SR501, dado que el rango de trabajo es de varios metros (generalmente

entre 3 y 7). Por ello se va a emplear un sensor activo como el que se ve en la Figura 22Figura 21, ya que permite detectar obstáculos desde 3 a 15 cm. Esta distancia representa una magnitud ideal para la implementación de este sensor en una maqueta, dado el tamaño reducido que presenta la representación tridimensional diseñada.

A continuación, se van a detallar los aspectos más relevantes del sensor elegido:

- Nombre: Sensor de proximidad por infrarrojos, para evitar obstáculos FC51.
- Fabricante: Waveshare.
- URL – Ficha técnica: http://www.dmf.unisalento.it/~denunzio/allow_listing/ARDUINO/FC51.pdf
- Precio: 1€ - 2€ unidad.

3.4 Dispositivos

En este apartado se va a detallar los dispositivos encargado del envío de mensajes. Para ello, y como se verá en apartados posteriores, se han elegido dos modelos de tarjetas en concreto.

3.4.1 NodeMCU ESP-8266

Es la tarjeta principal del sistema y es la encargada del envío de datos mediante el módulo WIFI (ESP-8266) del que dispone. Como podemos ver en la Figura 23, la tarjeta dispone de una gran variedad de puertos que permiten la conexión con otros dispositivos. Entre estos destacan el elevado número de pines digitales (DO-D8). Esta tarjeta se puede alimentar empleando un cable micro USB o directamente mediante el pin denominado *Vin* (*Voltage Input*, Entrada de tensión). Cabe destacar que la alimentación que proporciona la tarjeta es de 3.3V. Para dispositivos que requieran mayor voltaje será necesario un conversor de nivel lógico. Otros aspecto relevantes de la tarjeta son que permite la instalación en el tablero de pruebas dejando los pines libres, así como su compatibilidad con código Arduino empleando un *plugin*. La razón principal para la elección de este dispositivo es el que emplea un módulo WIFI 8266 que permite conexiones a internet de una manera simple y rápida.

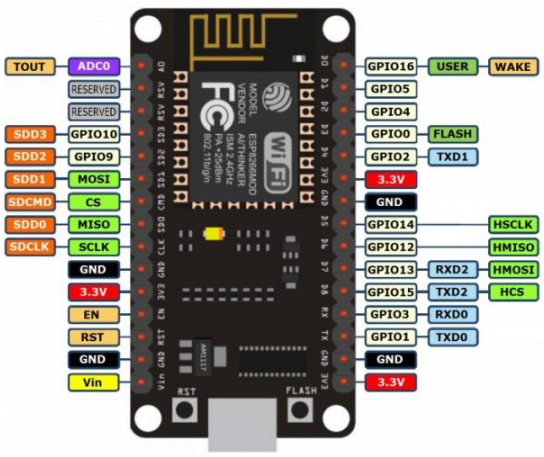


Figura 23 Salida de pines ESP-8266

Por último, hay que resaltar su reducido consumo. En cuanto a los aspectos negativos, hay que indicar que, la tarjeta únicamente dispone de un único puerto para lecturas analógicas, por lo que las instalaciones que requieran de más pines deberán emplear un multiplexor u otro dispositivo y transferir los datos mediante conexión serie. Para esta conexión dispone de los pines RX y TX que se pueden ver en la parte inferior derecha de la Figura 23.

En cuanto a alternativas disponemos de una gran número de dispositivos que pueden hacer un trabajo similar al que realiza la tarjeta elegida, como la tarjeta ESP32. En cuanto a las características principales encontramos que dispone de mayor número de pines, como se puede apreciar en la Figura 24, mayor potencia, ya que cuenta con un procesador más moderno, y mejor conexión WIFI. En contraposición presenta un mayor tamaño, por lo que no entraría en algunos tableros de prueba, un coste más elevado (de dos a tres veces mayor) y un mayor consumo.

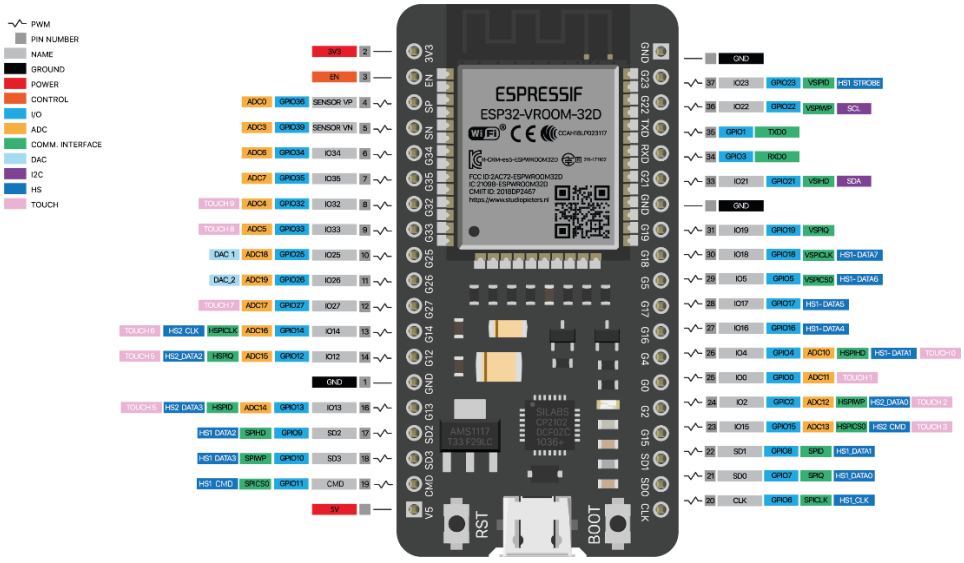


Figura 24 Salida de pines ESP-32

Las diferencias principales entre la anterior y la ESP8266 es el número pines y la potencia, pero esos factores no son relevantes para el envío de información que es necesario para este proyecto. Por estas razones, se ha decidido emplear la tarjeta con el módulo ESP8266 para el desarrollo del proyecto, ya que presenta un menor coste y cuenta con la potencia suficiente para realizar las tareas de procesado y envío de mensajes.

A continuación, se van a detallar los aspectos más relevantes de la tarjeta escogida:

- Nombre: NodeMCU ESP-8266 1.0.
- Fabricante: Amica.
- URL – Ficha técnica: https://www.elecrow.com/download/ESP8266_Specifications_English.pdf
- Precio: 4€ - 6€ unidad.

3.4.2 Arduino Uno

El dispositivo complementario empleado en el proyecto es la tarjeta Arduino Uno, ya que dispone de los pines necesarios para realizar las conexiones con el módulo ESP-8266. Las características principales de este dispositivo son:

Coste reducido, voltaje de operación de 5V y 3.3V, 14 pines digitales (6 de los cuales proporcionan salida con modulación por ancho de pulsos), múltiples pines analógicos, una alta potencia de procesamiento y amplia documentación en línea. Como podemos ver en la Figura 25, esta tarjeta sí que cuenta con las conexiones necesarias para las lecturas analógicas que se van a necesitar en el proyecto. Cabe destacar que existen revisiones de esta tarjeta con conexión a internet integrada, pero tienen un coste mayor. Por esta razón y unida a que se dispone con anterioridad de esta tarjeta se ha decidido emplearla para el proyecto.

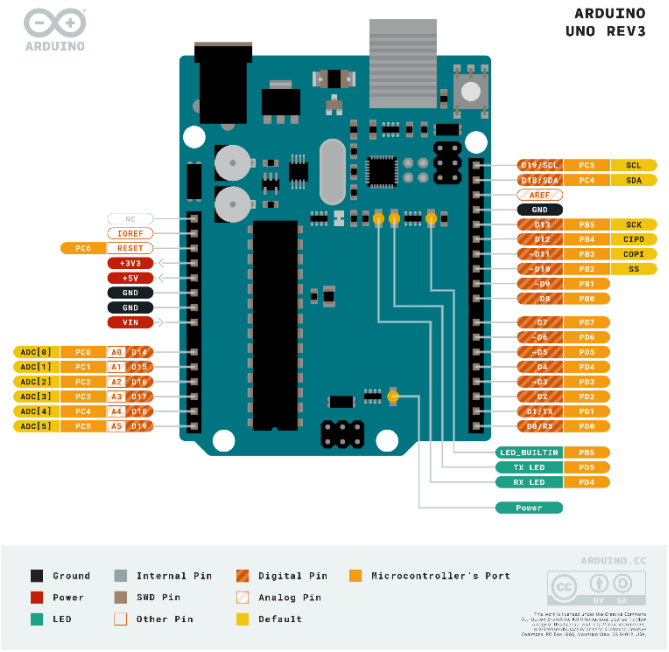


Figura 25 Salida de pines Arduino Uno

A continuación, se van a detallar los aspectos más relevantes de la tarjeta elegida:

- Nombre Arduino Uno Rev3.
- Fabricante: Arduino.
- URL – Ficha técnica: https://docs.arduino.cc/static/383392a830d886a9b694712d7d9f4dd0/A000066-datasheet.pdf
- Precio: Oficial 22€ unidad, clones 8€ - 15€ unidad.

3.5 Protocolos de uso

Una vez que se han explicado los sensores que se van a emplear, y su funcionamiento, vamos a detallar cuáles van a ser los distintos protocolos que se van a seguir para detectar diferentes comportamientos. Para ello se van a definir los estados que se pueden producir dentro de la habitación y las condiciones para que se generen las situaciones que se van a analizar.

3.5.1 Caída

Como se ha contado en apartados anteriores, uno de los aspectos importantes a tener en cuenta en el desarrollo del proyecto es el control ante las caídas. Para ello se diferencian dos tipos de caídas, un tipo de caída que se da llevando el dispositivo inalámbrico y otro que se da sin utilizarlo.

En los siguientes apartados se van a describir las situaciones que pueden darse dentro de una habitación y de los mensajes en respuesta a esas opciones que obtiene el responsable en el panel de control. Estos mensajes variarán dependiendo de la gravedad de la acción y emplearán distintos colores para ser rápidamente reconocibles. Los mensajes, información que se indica y los colores son los siguientes:

- Mensaje de información: Se enviará información relativa a acciones que no presentan peligro, como por ejemplo "Usuario usando la cama". Este mensaje irá representado con el color verde. El mensaje se reproducirá durante un determinado periodo de tiempo para luego desaparecer. Se indicará el estado de la estancia con el color verde una vez que el mensaje deje de mostrarse.
- Mensaje de aviso: Las acciones de las que informan este tipo de mensajes representan un grado mayor de peligro que los anteriores. Se emplearán para acciones como "Cruce de puerta" o "Usuario ha dejado de usar la cama". Estos mensajes son importantes para que el responsable sepa qué situación ha ocurrido dentro de la estancia. Irán acompañados del color amarillo. También desaparecerán pasados un tiempo.
- Mensaje de alerta: Estos mensajes representarán las acciones potencialmente peligrosas. Requieren de atención inmediata por parte del responsable, para ello se utiliza el color rojo que resalte. Además, este mensaje requerirá de una acción por parte del responsable para que desaparezca. Las acciones que activan este tipo de mensajes son, por ejemplo, las caídas o salidas de la habitación en un horario inadecuado.

Una vez explicadas los distintos mensajes que se van a mostrar, vamos a describir detalladamente las distintas situaciones que pueden darse.

3.5.1.1 Caída libre

Este tipo de accidente se presenta si el usuario lleva consigo el módulo inalámbrico. Este módulo está compuesto de una tarjeta ESP8266, un Giroscopio-acelerómetro, un altavoz y un botón del pánico.

Este módulo permite detectar, usando un algoritmo, como se indicó con anterioridad, las posibles caídas de la persona que lo lleve incorporado. En el caso de que se produzca una caída, el sistema emitirá por medio del altavoz una alerta para ser fácilmente detectado. Además de la alerta sonora se enviará un mensaje al puesto de control alertando de esta situación.

Este dispositivo ofrece una reacción análoga cuando se acciona el botón del pánico. Este equipamiento está pensado para ser llevado en todo momento por parte del usuario.

Como puede haber momentos en los que la persona no lleve puesto el dispositivo descrito, se proponen dos sistemas adicionales para detectar posibles caídas, que se describirán a continuación.

3.5.1.1 Caída sobre alfombra

Para controlar otras posibles caídas se presenta una superficie que controla la presión mediante un sensor a la vez que se usa un par de sensores IR para comprobar qué situación se ha producido. El dispositivo quedaría de la siguiente manera:

En el suelo una alfombra, con la funcionalidad de sensor de presión explicada anteriormente, por debajo de las rodillas un sensor IR, que llamaremos IR-Abajo, y por último otro sensor IR a la altura del pecho, que denominaremos IR-Arriba.

Como se emplean tres sensores distintos existirán varias situaciones a controlar dependiendo de las lecturas de estos. En total existirán ocho diferentes. Es importante usar solo las lecturas que sean de interés y descartar el resto, para facilitar la comunicación y la velocidad de esta. Una vez aclarado esto tendremos tres situaciones de interés y una extra que puede tenerse en cuenta:

1. Alfombra con presión + IR-Abajo activado + IR-Arriba activado: Esta es la situación donde todos los sensores se encuentran activos. Si detectamos presión en la alfombra nos indica que hay una persona en ella, pero no sabemos la forma en la que se encuentra encima. Si detectamos en los dos sensores IR, tanto a baja como a elevada altura, nos indica que la persona se encuentra de pie encima de la alfombra. Situación que puede darse al levantarse o sentarse en la cama.
2. Alfombra con presión + IR-Abajo activado + IR-Arriba desactivado: Contraria a la anterior. Esto se debe a que si detectamos presión en la alfombra y el IR-Abajo esta activo, pero no detectamos nada en altura, indica que la persona se encuentra encima de la alfombra, pero sin estar de pie. Podemos asumir que si se encuentra en la alfombra tumbado está así después de sufrir una caída, esto puede darse tanto si se cae desde la cama, como si la caída se produce estando de pie.
3. Alfombra sin presión + IR-Abajo desactivado + IR-Arriba desactivado: Situación en la que el usuario no se encuentra en la zona de la alfombra. Es importante tenerlo en cuenta, ya que va a ser el comportamiento más habitual del dispositivo, por ello esta situación no debe comunicarse.
4. Alfombra con presión + IR-Abajo desactivado + IR-Arriba desactivado: Esta es la situación extra que se tiene en cuenta. Se produce cuando únicamente se encuentra accionada la alfombra. Esto puede deberse a dos situaciones, la primera es que el usuario acabe de empezar el contacto

con la misma y por ello el sensor IR-Abajo no se encuentre en rango. Y la segunda situación es que puede haber un objeto encima de la alfombra y por ello esté activada. Esta situación es importante tenerla en cuenta y calibrar los dispositivos para obtener mensajes adecuados si se precisan.

3.5.1.1 Caída baño

Dado que la caída también se puede producir en el baño, se ha decidido emplear el mismo sistema. Cumple los mismos principios que el caso anterior, únicamente con los sensores adaptados a este entorno.

3.5.2 Cruce puerta

Un aspecto importante a tener en cuenta es las entradas y salidas de las diferentes estancias, teniendo claro unas condiciones. Estas pueden ser el tipo de dependencia de la persona o la hora a la que se produzcan. Teniendo esto en cuenta podemos distinguir cuatro posibles situaciones. Para discernir entre estos escenarios se van a emplear dos sensores IR, uno cercano a la puerta y otro un poco más alejado, de manera que nos permita diferenciar entre sí la persona ha entrado o ha salido de la estancia. Por ejemplo, si se activa primero el sensor más próximo a la puerta y posteriormente el que se encuentra más alejado nos indica que la persona ha entrado, en el caso es el contrario entendemos que la persona ha salido.

Un apunte importe a recalcar es que el sistema como está realizado detectará la entrada y salida en la habitación de cualquier persona. Para el caso explicativo se describirán las acciones correspondientes a la persona dependiente que usa la habitación. En futuras revisiones se pueden crear sistemas de control del aforo empleando tarjetas *RFID* (*Radio Frequency Identification*, Identificación por radiofrecuencia), eliminando así posibles lecturas que no correspondan con el usuario.

3.5.2.1 Salida habitación

Como se ha comentado en el párrafo anterior, para detectar esta situación debemos tener en cuenta que el sensor que se encuentra más alejado debe haberse activado previamente. Si se cumplen la condición anteriormente descrita y que se active el sensor más cercano a la puerta, se determina una salida de la habitación. Esta situación hará que se envíe un mensaje al panel de control indicando que

se ha producido una salida de la estancia. Este mensaje tendrá que ser interpretado de distinta manera, teniendo en cuenta, entre otras cosas, la hora a la que se produce. Si se hace durante el día entonces se trata de una salida habitual de la habitación, pero por la noche puede indicar una situación peligrosa y tendrá que ser mostrada como tal.

3.5.2.2 Entrada habitación

Situación inversa a la anterior, pero se debe tener en cuenta los mismos parámetros. En este caso primero se activaría el sensor más próximo a la puerta y después el más alejado, indicando la entrada en la habitación. También se deberá tener en cuenta el horario en el que se producen estas acciones, al igual que el caso anterior.

3.5.2.3 Salida baño

En cuanto al baño se producen situaciones análogas. La única diferencia es que la salida es independiente, de la hora a la que se produzca, solamente se realizará un aviso y nunca una alerta. Esto permitirá que se controle las entradas y salidas, pero con independencia del horario.

3.5.2.4 Entrada baño

En cuanto a la entrada se considera el mismo planteamiento que la salida. Se supondrán estas situaciones como avisos. También son importante tenerlas en cuenta para controlar el sistema de iluminación si así se requiriera.

3.5.3 Iluminación

El sistema de iluminación es otro aspecto importante en el control de la estancia. Debemos tener en cuenta los horarios de estas, así como comprobar si una luz ha quedado encendida, algo que es bastante habitual. Para ello usaremos fotorresistencias para comprobar si la iluminación se encuentra apagada o no. Este control se hará en dos estancias, la habitación y el baño, pudiendo ampliarse en un futuro.

3.5.3.1 Luz encendida habitación

Esta situación se produce si la fotorresistencia que se encuentra en la habitación detecta una cantidad de luz superior a una determinada cantidad. Esta enviará mensaje al panel de control indicando esta situación.

3.5.3.2 Luz apagada habitación

En cuanto al caso contrario también tenemos que tener en cuenta cuando se apague la luz, indicando que la situación ha sido solucionada. Por ello si la fotorresistencia no detecta iluminación por debajo de un margen enviará el correspondiente mensaje.

Se podría realizar un control mucho más exhaustivo de las horas a las que se producen estos comportamientos. Por sencillez de la demostración se ha decidido no realizar.

3.5.3.3 Luz encendida baño

Estas dos situaciones son similares a los casos anteriores. Incluyendo el control por horas si fuese necesario.

3.5.3.4 Luz apagada baño

Como se comentó en su caso en la habitación, tendremos que poner igual de atención a los momentos en los que se enciende la luz como a los que se apaga.

3.5.4 Cama

En cuanto al control de la cama se plantea realizar la misma implementación que para las alfombras. Es decir, crear un sensor de presión que permita comprobar si la persona está encima de la cama o no. Para simplificar nuestro sistema únicamente vamos a comprobar si el usuario está usando la cama o no. Una posible mejora, de mayor complejidad, sería la posibilidad de detectar la posición de sentado o tumbado.

3.5.4.1 Presencia cama

Si el sensor de presión se encuentra activo indicaría que el usuario está usando la cama. La primera vez que se avise se mostrará el mensaje correspondiente. Cuando el usuario ya esté en la cama tan solo se mostrará la superficie iluminada. En principio el mensaje que se muestra es de información únicamente. Ya que se trata de una situación habitual, ya que se podría si se quisiera hacer también un control por horas.

3.5.4.2 Ausencia cama

Caso contrario al anterior, y se produce cuando el sensor de presión deja de estar activo. En esta situación, sí se emitirá aviso, ya que será una situación anormal. El control por horas también se puede llevar a cabo.

3.5.5 Botón del pánico

En cuanto al botón del pánico, implementado en el módulo inalámbrico, contaremos con un sistema de alerta. Este dispositivo busca emular los botones del pánico actuales, que además de emitir la alerta, emitirá un sonido y mostrará el correspondiente mensaje en la consola de administración.

3.6 IoT Core

El servicio de Google Cloud Platform IoT Core es uno de los aspectos más relevantes del desarrollo del proyecto. Gracias a este servicio se puede descentralizar, reducir costes y escalar los dispositivos conectados. IoT Core permite, entre otras muchas funciones, conectar una serie de equipos mediante WIFI usando el protocolo MQTT. Este protocolo permite el envío de mensajes de información recogida por los sensores para su posterior procesado. A continuación, se va a hablar de los aspectos más importantes de este servicio.

3.6.1 Funcionamiento

Como se ha comentado, IoT Core sirve como intermediario de los mensajes con el panel de control final. Para llegar al usuario los datos siguen el camino que podemos ver en la Figura 26.

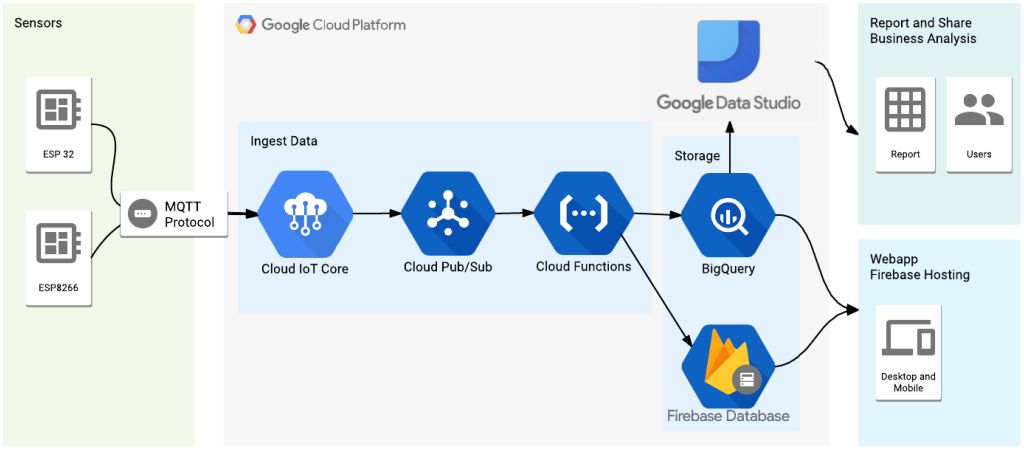


Figura 26 Funcionamiento de IoT Core de Google

Primeramente, los sensores recogen datos del entorno, los cuales son procesados por los Arduinos correspondientes. Dependiendo de la lógica que se quiera implementar se enviará un mensaje u otro, que se envía mediante MQTT. Este es un servicio de mensajería push con patrón publicador/suscriptor, es decir, una infraestructura en la que los clientes se conectan con un servidor central denominado broker, en este caso IoT Core, como se puede apreciar en la Figura 26.

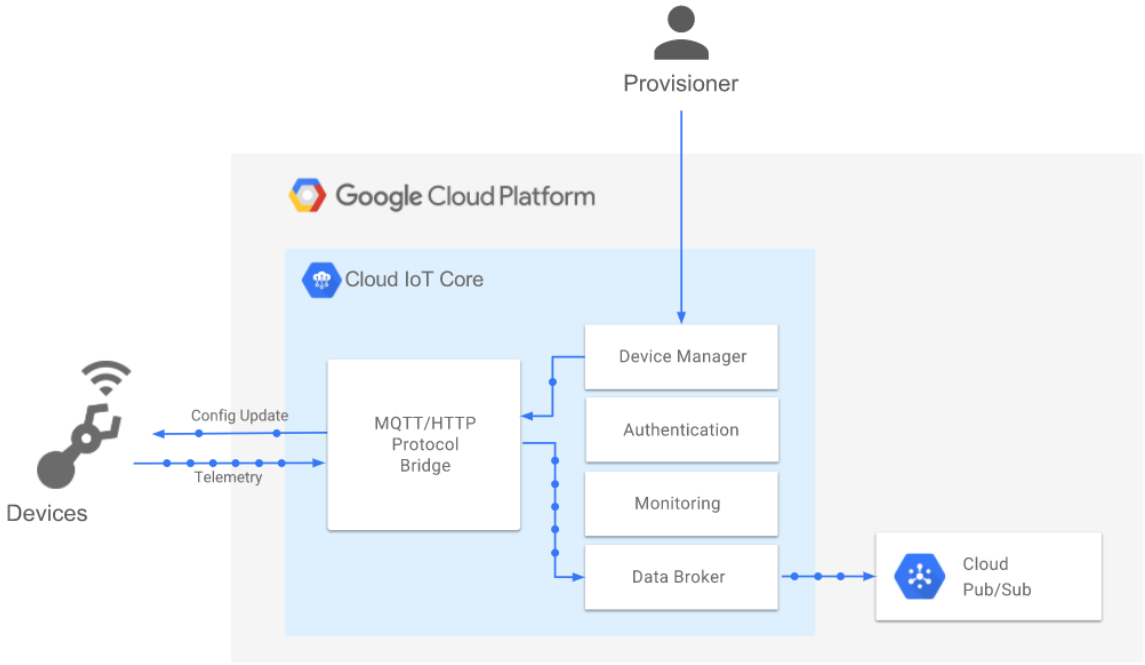


Figura 27 Funcionamiento del bróker

Como también podemos apreciar en la Figura 27, para permitir la conexión se debe registrar y autenticar la tarjeta usando claves públicas y privadas, así como cargar el software necesario para que la tarjeta sea reconocida.

Usando las bibliotecas que Google presenta en su documentación tan solo basta con realizar la conexión WIFI de la tarjeta y con una orden podemos enviar los datos que queramos.

Seguidamente, el dato le llega a IoT Core. Como se ha dicho anteriormente, se debe registrar un dispositivo para su futura conexión. Para ello se debe crear un registro, cada registro puede almacenar una gran cantidad de dispositivos diferentes. Una vez que IoT Core determina a que registro pertenece el dispositivo que ha enviado el mensaje lo reenviará a Pub/Sub.

Pub/Sub es un patrón de mensajería en el que los emisores de mensajes, llamados editores, no programan los mensajes para que se envíen directamente a receptores específicos, llamados suscriptores, sino que clasifican los mensajes publicados en temas sin saber qué suscriptores puede haber, si los hay. Del mismo modo, los suscriptores expresan su interés por uno o varios temas y sólo reciben los mensajes que les interesan, sin saber qué editores hay, si es que los hay.

Es decir, Pub/Sub es un servicio de Google Cloud Platform que permite el envío de mensajes de forma asíncrona mediante la creación de temas y subscripciones. Cada vez que le llegue uno a IoT Core, este enviará al tema correspondiente. En este tema el mensaje quedará almacenado durante un periodo de tiempo determinado. Aparte de almacenarse será reenviado a la subscripción que corresponda.

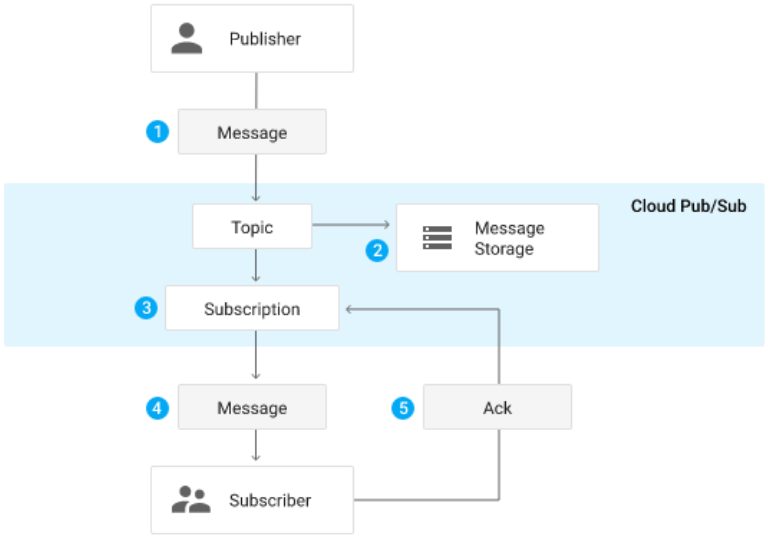


Figura 28 Funcionamiento de Pub/Sub

Dentro de este tema pueden existir distintas Suscripciones que permiten manejar los datos que se hayan recibido, permitiendo realizar diversidad de tareas con el mismo mensaje.

Una vez que el mensaje le llega a la Subscripción se introducirá en una cola con todos los que se hayan enviado. Como pudimos ver en la Figura 26, ahora entra en juego Cloud Functions o Funciones en la nube.

Google Cloud Functions es un entorno de ejecución sin servidores para compilar y conectar servicios en la nube. Con Cloud Functions, se puede programar funciones simples de un solo propósito vinculadas a eventos emitidos desde una determinada infraestructura y servicios en la nube.

Esta función se activa cuando ocurre un evento que está bajo observación. Posteriormente se ejecuta un código en un entorno completamente administrado, sin necesidad de aprovisionar infraestructura ni que exista necesidad por administrar servidores. Es decir, estas funciones se pueden emplear para diversas tareas. En este caso se encargarán de que cada vez que, llegue un nuevo mensaje a la cola de Pub/Sub, obtendrán los datos e información relevante del mensaje y lo enviarán a un sistema de almacenamiento.

Para ello se puede emplear código en diversos lenguajes, que permitirá la transformación y envío a los distintos servicios que dispone Google, como Firebase, Firestore o BigQuery.

Para este proyecto se va a emplear Firestore, que es una base de datos de documentos NoSQL que permite almacenar, sincronizar y consultar fácilmente datos en páginas web y dispositivos móviles a escala global.

Se trata de una base de datos no relacional, que presenta la información en forma de colecciones con variedad de documentos. Una vez que cada dato se almacena dentro del documento elegido, dentro de la colección indicada, el dato podrá ser accesible desde cualquier equipo o momento.

Firestore permite la conexión a gran diversidad de equipos ya sea usando Java, Node.js, C++, Unity o REST, así como a los distintos sistemas operativos de móviles. Para el desarrollo del proyecto se va a emplear JavaScript para obtener los datos en el cliente.

Para realizar este paso, únicamente incluimos la clave que nos proporciona Firestore y usando una serie de funciones asíncronas podemos recuperar la información que tengamos en el documento que queramos.

Todos estos servicios permiten que un sensor conectado a un Arduino envíe un dato y este sea leído por cualquier persona que se encuentre en posesión del panel de control que lea de la base de datos. Esta arquitectura ofrece una escalabilidad prácticamente infinita y una rapidez y sencillez que con otros estándares no serían posibles.

Teniendo esto en cuenta, podemos tener los dispositivos en otro país y supervisarlos desde cualquier parte del mundo casi al instante.

Pero toda esta tecnología tiene un coste y si es demasiado elevado desvirtuaría el propósito del proyecto. A continuación, se discutirá sobre los distintos precios y costes del uso de esta tecnología.

3.6.2 Costes

Como se indicó en los objetivos del proyecto uno de los aspectos más importantes es la escalabilidad y el coste reducido del sistema. Como hemos visto anteriormente Google Cloud Platform permite la primera parte de escalabilidad, fiabilidad y rapidez, ahora falta comprobar los costos que conlleva el uso de esta plataforma.

Para este apartado vamos a describir los distintos costes de cada funcionalidad siguiendo el orden del apartado anterior. También se indicará la cuantía tanto para una instalación individual como en una residencia. Cabe destacar que existe un desglose de todos los costes, así como del uso en Google Cloud.

IoT Core

Como podemos ver en la Tabla 1, y si no se supera los 250MB el coste es 0 $, en el caso de superarlo el precio por MB es muy reducido. En las pruebas realizadas durante el desarrollo del proyecto no se ha superado en ningún momento más de 3KB. Por lo que podemos asumir que el coste mensual se mantendrá a 0 € hasta que se utilicen muchos dispositivos. En una instalación individual el coste será de 0 € en este apartado, mientras que en una instalación donde existan cientos de dispositivos el coste no será 0€ pero será contenido. Para comprobarlo vamos a realizar el cálculo.

Volumen de datos mensual	Precio por MB	Dispositivos registrados	Cargo mínimo*
Hasta 250 MB	0,00 €	Ilimitados	1024 bytes
De 250 MB a 250 GB	0,0045 €	Ilimitados	1024 bytes
De 250 GB a 5 TB	0,0020 €	Ilimitados	1024 bytes
5 TB y más	0,00045 €	Ilimitados	1024 bytes

Tabla 1 Desglose precios IoT Core

Vamos a realizar un ejemplo para un usuario particular y una residencia con unas 100 camas que es superior a la media de España, alrededor de 70.

Usuario individual:

1 dispositivo * 96 PINGREQ * /dispositivo día * 30 días * 1024 *bytes* = 2.9MB

Coste = 0€

Residencia 100 habitaciones:

100 dispositivo * 96 PINGREQ * /dispositivo día * 30 días * 1024 *bytes* = 294MB

Coste: (294MB -250MB) * 0.0045 = 0.198 €

*PINGREQ es un mensaje enviado a Cloud IoT Core para mantener activa la conexión MQTT.

Como podemos comprobar pese a tener 100 dispositivos el coste sería menor a 20 céntimos mensuales en este apartado.

Pub/sub

En cuanto a Pub/Sub los primeros 10GiB de rendimiento de una cuenta por mes natural son gratuitos, después de esto son 40€ por TiB. Como podemos observar por los valores de mensajes anteriores el coste, independientemente de si se trata de usuario individual o una residencia, será 0€ siempre.

Cloud Functions

Cloud Functions ofrece un precio gratuito para siempre de los recursos de tiempo de procesamiento, que incluye la asignación de GB por segundo y GHz por segundo. Este nivel incluye 2 millones de invocaciones gratuitas, así como 400.000 GB por segundo, 200.000 GHz por segundo de tiempo de procesamiento y 5 GB de tráfico de salida a Internet al mes. Teniendo en cuenta estos datos, el coste para un usuario individual será de 0 € y en el caso de residencias también lo será, a menos que se trate de una de gran tamaño. En las pruebas realizadas durante el proyecto el uso diario no llega a 50 *bytes* por segundo.

FireStore

En cuanto a FireStore encontramos la siguiente información representada en la Tabla 2:

	Cuota gratuita al día	Precio tras superar la cuota gratuita (por unidad)	Unidad de precio
Operaciones de lectura de documentos	50.000	0,06 €	por 100.000 documentos
Operaciones de escritura de documentos	20.000	0,18 €	por 100.000 documentos
Operaciones de eliminación de documentos	20.000	0,02 €	por 100.000 documentos
Datos almacenados	1 GB de almacenamiento	0,18 €	GB al mes

Tabla 2 Desglose precios FireStore

Como podemos ver tenemos una serie de operaciones gratuitas para cada cuenta que usemos en este servicio. Si realizamos el cálculo al igual que con IoT Core obtendremos lo siguiente:

Usuario individual – Coste diario:

- **O. lectura**: 1 operación cada dos segundos = 43200 < 50000 -> **Coste 0€**
- **O. escritura**: 200 operaciones cada hora = 4800 < 20000 -> **Coste 0€**
- **O. eliminación**: 200 operaciones cada hora = 4800 < 20000 -> **Coste 0€**

Como podemos comprobar para el usuario individual con este número de operaciones tendríamos un coste de 0€ mensuales. Aunque hay que tener en cuenta que el servicio precisa de Google Storage y esto conlleva un coste medio de 0.01€ a 0.05€ dependiendo de la necesidad de Cloud Storage.

Residencia 100 habitaciones – Coste diario:

- **O. lectura**: 100 habitaciones * 1 operación cada 4 segundos = 2160000.
  - Coste: (2160000– 50000) * (0.06€ / 100000) = **1.266€** día.
- **O. escritura**: 100 * 100 operaciones cada hora = 240000
  - Coste: (240000 - 20000) * (0.18€ / 100000) = **0.396€** día.
- **O. eliminación**: 100 * 100 operaciones cada hora = 240000
  - Coste (240000 - 20000) * (0.02€ / 100000) = **0.044€** día.

Coste mensual

- O. lectura: 1.266€ día * 30 días = 37.98€.
- O. escritura: 0.396€ día * 30días = 11.88€.
- O. eliminación: 0.044€ día * 30 días = 1.32€.

Total: 51.18€ por 100 habitaciones -> 0.5118€ por habitación al mes.

Como podemos comprobar en este cálculo, el coste para una residencia de 100 habitaciones queda sobre los 50€ mensuales. Hay que tener en cuenta los cobros que pueda realizar Cloud Storage, que estarán en el rango de los céntimos.

También hay que reseñar que el cálculo se ha hecho sobre el total, pero se podría tener una cuenta por cada planta o diferenciar la distribución de habitaciones de alguna otra manera, lo que haría reducir los costes.

Otro aspecto importante es que Cloud Platform realiza los cálculos de manera interna y por ello estos pueden estar alejados de la realidad. También hay que destacar que las operaciones de lectura son las que están calculadas exactamente, el resto de las operaciones son aproximaciones teniendo en cuenta el uso actual.

A modo de resumen, para un usuario individual el coste mensual que podemos esperar es menor a 0.10€ mensuales. Aproximadamente podemos fijar el coste anual en 1€.

En cuanto a una residencia podemos fijar el coste mensual por debajo de los 75€. Recalcar que se trata de una residencia de 100 habitaciones, habrá que tenerlo en cuenta para residencias con más o menos estancias. En cuanto el coste anual lo situaremos por debajo de 1000€. Como se comentó anteriormente este se puede reducir si se divide la residencia en plantas o secciones y usamos distintas cuentas.

4. Técnicas y herramientas

En este apartado vamos a hablar brevemente del software utilizado para el desarrollo del proyecto, así como el hardware y lo necesario para el desarrollo de la maqueta.

4.1 Software

Antes de comenzar a nombrar el software utilizado, es importante recalcar, que dado que se trata de un proyecto Arduino con funcionalidad de Google Cloud Platform, se ha tenido que usar programas o funcionalidades propietarias.

Arduino IDE

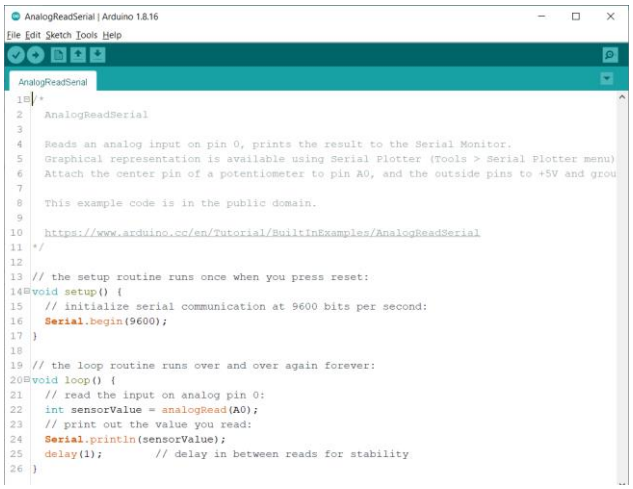


Figura 29 Ejemplo código en el IDE Arduino

Nombre: Arduino IDE.

Versión: 1.8.16.

Desarrollador: Arduino LLC.

URL: https://www.arduino.cc/en/software

Descripción: Se ha utilizado el IDE propietario de Arduino para el desarrollo del código de estos dispositivos. Como se han utilizado distintas tarjetas y sensores se han instalado distintas librerías y *plugins* para llevarlo a cabo.

Alternativas: En cuanto a alternativas se podría haber usado tanto la versión 2.0 del mismo desarrollador o Visual Studio Code con la extensión de PlataformIO. Al final la elección fue por el IDE de Arduino, dada la sencillez del programa y de la experiencia previa con el mismo.

Licencia: *GNU General Public License*, versión 2.0 o posterior.

Google Cloud Platform y Google Cloud SDK

Figura 30 Diagrama con los distintos servicios de Google

Nombre: Google Cloud Platform y Google Cloud SDK.

Versión: 391.0.0.

Desarrollador: Google Inc.

URL: https://cloud.google.com/sdk?hl=es

Descripción: Como se ha comentado en apartados anteriores se ha usado el software propietario de Google para el desarrollo del proyecto. Para todo el apartado de IoT Core se ha empleado la plataforma web de Google Cloud [14]. Así como el SDK para la creación y de Google SDK Software Development Kit "Kit de desarrollo software".

Alternativas: Existen alternativas a esta plataforma como por ejemplo Amazon Web Service o Azure, se eligió esta plataforma ya que cuenta con librerías para las tarjetas empleadas.

Licencia: *Apache License v. 2.0.*

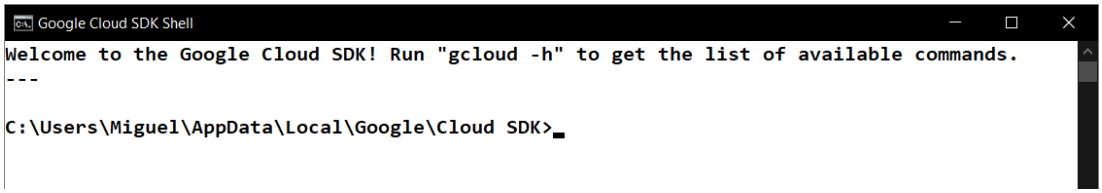


Figura 31 Consola de Google SDK

WebStorm

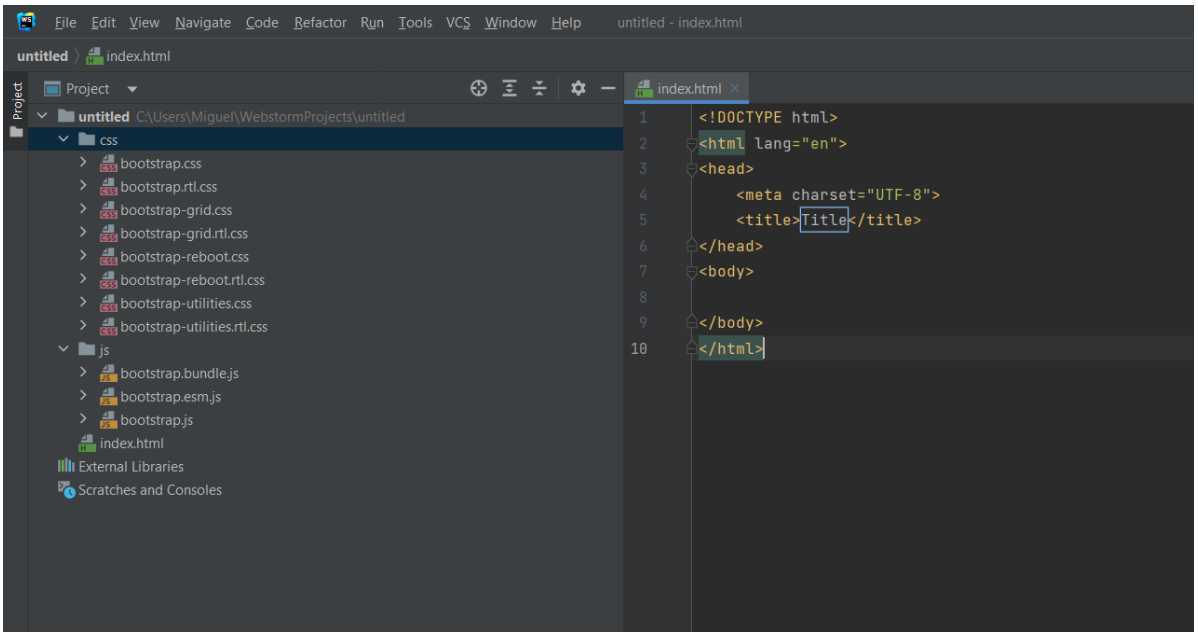


Figura 32 Pantalla de inicio de WebStorm

Nombre: WebStorm.

Versión: 2022.1.

Desarrollador: JetBrains s.r.o.

URL: https://www.jetbrains.com/es-es/webstorm/

Descripción: Esta aplicación es el IDE de JetBrains pensado para el desarrollo en JavaScript y creación de aplicaciones web. Cuenta con diversas funcionalidades, como especialización en JavaScript, autocompletado, herramientas de desarrollo integradas, vista previa de páginas web integrada. Otra ventaja importante es que no tiene coste para los alumnos de universidades, dado que se trata de una aplicación de pago.

Alternativas: Se podría haber usado distintos programas para el desarrollo Web como SublimeText, Visual Studio Code o Atom. Al final se escogió este programa por las ventajas indicadas anteriormente.

Licencia: gratuita para uso educativo, de pago para uso particular o comercial.

Fritzing

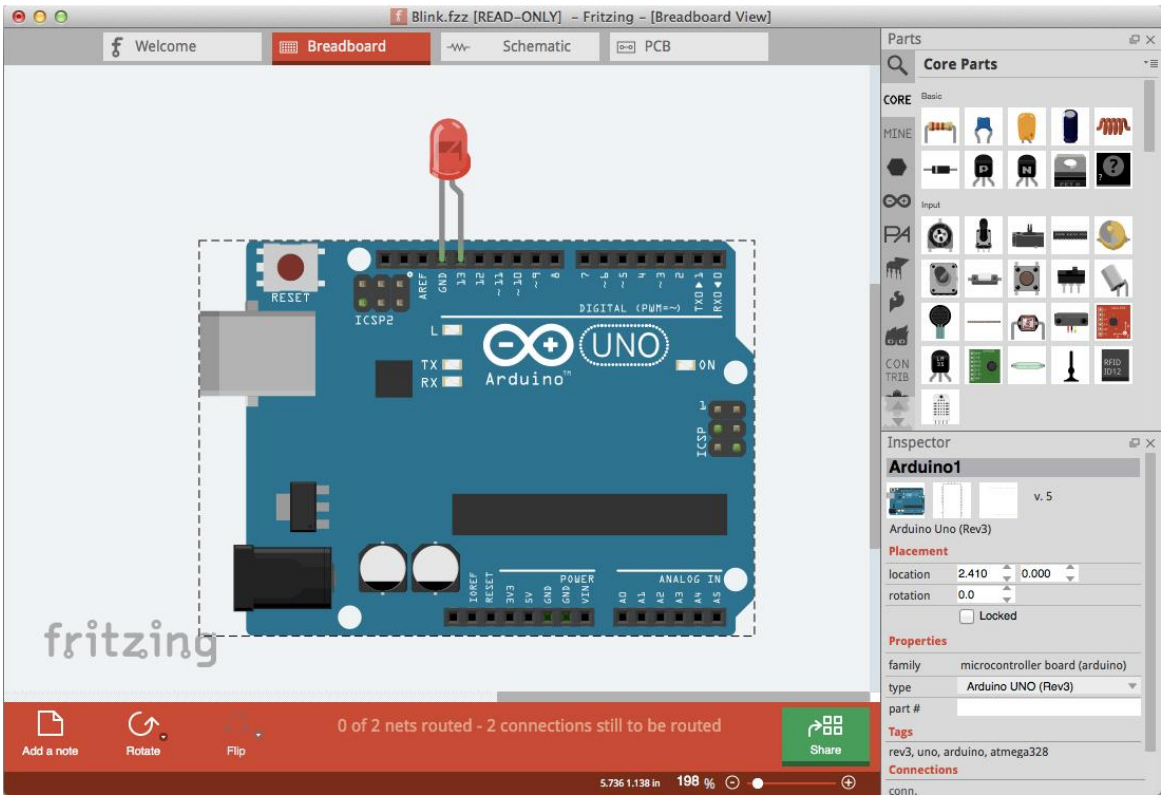


Figura 33 Ejemplo de circuito en Fritzing

Nombre: Fritzing.

Versión: 0.9.9.

Desarrollador: Interaction Design Lab.

URL: https://fritzing.org/

Descripción: Para el diseño de los circuitos y diagramas se ha empleado Fritzing, ya que permite la importación de distintos sensores y tarjetas. También tiene diferentes maneras para ver el esquema o la *PCB* (*Printed Circuit Board*, placa de circuito impreso).

Alternativas: Otra que se podía haber empleado Autodesk Tinkercad, pero cuenta con menor cantidad de dispositivos.

Licencia: *GNU GPL v3*, gratuita.

Autodesk Tinkercad

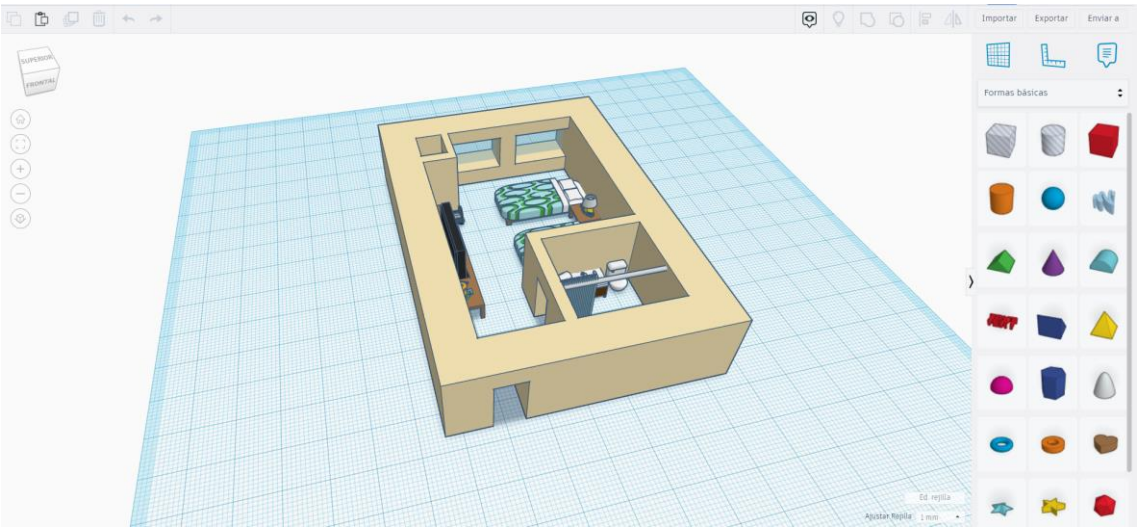


Figura 34 Ejemplo de diseño 3D de Tinkercad

Nombre: AutoDesk Tinkercad.

Versión: 2019_10_14.

Desarrollador: Autodesk Inc.

URL: https://www.tinkercad.com

Descripción: Plataforma en línea destinada a la creación de Diseños en 3D, circuitos y código en bloque. Para este proyecto se ha empleado como herramienta de diseño de la maqueta en 3D. La principal razón para esto es que cuenta con una gran biblioteca de diseños que pueden ser utilizados, como el que se ve en la Figura 34.

Alternativas: Se podrían haber utilizado distintos programas para el modelado 3D como Blender, AutoCAD o AutoDesk.

Licencia: Gratuita.

Programas de diseño 3D

Parte del desarrollo del proyecto ha consistido en la creación de diferentes estructuras y maquetas en 3D, donde se han utilizado diversos programas destinados a estas tareas que se describen con mayor detalle en los siguientes apartados:

OpenSCAD

Nombre: OpenSCAD.

Versión: 2021.01

Desarrolladores: Marius Kintel, Clifford Wolf.

URL: https://openscad.org/

Descripción: OpenSCAD es una aplicación libre para crear objetos sólidos de CAD. No es un editor interactivo sino un compilador 3D basado en un lenguaje de descripción textual. Permite la modificación de diseños mediante cambios en el código.

Licencia: GPLv2.

Meshmixer

Nombre: Autodesk Meshmixer.

Versión: 3.5.

Desarrollador: Autodesk.

URL: https://www.meshmixer.com/

Descripción: Se trata de un software puntero para trabajar con mallas de triángulos, o como lo describen sus desarrolladores, la "navaja suiza" de la edición de archivos STL y mallas 3D.

Licencia: Gratuita.

Thingiverse

Nombre: Thingiverse.

Versión: -

Desarrollador: MakerBot Industries.

URL: https://www.thingiverse.com/

Descripción: sitio web dedicado a compartir archivos de diseño digital creados por los usuarios. Ofrece principalmente diseños de hardware gratuitos y de código abierto.

Licencia: *GNU General Public License*, se debe citar al creador.

PrusaSlicer

Nombre: PrusaSlicer.

Versión: 2.4.2.

Desarrollador: Prusa Research.

URL: https://www.prusa3d.com/es/

Descripción: PrusaSlicer es una herramienta de código abierto, rica en características y frecuentemente actualizada, que contiene todo lo que se necesita para exportar los archivos de impresión a una impresora de la marca.

Licencia: *GNU Affero General Public License v3.0*.

Programas para cortadora láser

Se ha utilizado software para el desarrollo de los planos de la base de la y para el posterior cortado mediante el láser.

Inkscape

Nombre: Inkscape.

Versión: 1.2.

Desarrollador: Equipo Inkscape.

URL: https://inkscape.org/

Descripción: editor de gráficos vectoriales gratuito y de código abierto que se utiliza para crear imágenes vectoriales, principalmente en formato Scalable Vector Graphics (SVG). Se pueden importar y exportar otros formatos.

Licencia: *GPLv3*.

Lightburn

Nombre: Lightburn.

Versión: 1.1.04.

Desarrollador: LightBurn Software.

URL: https://lightburnsoftware.com/

Descripción: Lightburn es un software de diseño, edición y control para su cortadora láser. Con lightburn se puede: Importar material gráfico, organizar, editar y enviar el resultado directamente a su cortadora láser.

Licencia: De pago, https://lightburnsoftware.com/pages/how-the-lightburn-license-works.

Otras herramientas

También es importante recalcar que se han empleado gran variedad de aplicaciones como:

Google Chrome: Navegador empleado para usar las diferentes plataformas web, así como el control y verificación de la plataforma web del proyecto. La verificación se ha realizado también en Firefox y en Microsoft Edge.

Paint.net: Programa de edición de imagen empleado para modificar distintos archivos creados para la plataforma web.

4.2 Leguajes

Los lenguajes empleados para el desarrollo del proyecto son:

- **C++** adaptado a Arduino: Lenguaje usado para la lectura y envío de mensajes a la plataforma de Google. No es C++ puro, sino que es una adaptación que provienen de avr-libc, que aporta una librería de C de alta calidad para usar con GCC en los microcontroladores AVR de Atmel y muchas funciones específicas para los MCU AVR de Atmel.
- **HTML** (*HyperText Markup Language*, Lenguaje de Marcas de Hipertexto): Componente más básico de la Web, encargado de definir el significado y la estructura del contenido web.

- **CSS** *(Cascading Style Sheets*, Hojas de estilo en cascada): Lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en lenguaje de marcado como por ejemplo HTML.
- **JavaScript:** Lenguaje de programación interpretado orientado a objetos, basado en prototipos y débilmente tipado que se utiliza para dar funcionalidad las páginas web.
- **Node.js:** Entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript. Es asíncrono, orientado a eventos y basado en el motor V8 de Google. Empleado para el desarrollo de las CloudFunctions.
- **JSON** *(JavaScript Object Notation*, Notación de Objeto de JavaScript): Es un formato de texto destinado al intercambio de información. Permite representar los datos de manera estructurada usando la sintaxis de objeto de JavaScript. Empleado la comunicación de datos entre los sensores y la plataforma de Google, así como la lectura de datos desde la base de datos Firestore por parte del cliente local.

5. Aspectos relevantes del desarrollo

El proyecto se ha realizado bajo el paradigma del desarrollo de proyectos software del Proceso Unificado (Figura 35).

Este es un marco de trabajo genérico adaptable a una gran variedad de aplicaciones, proyectos y sistemas software. Se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser interactivo e incremental.

Estas últimas características quieren decir que se divide en una serie de fases, y estas a su vez se descomponen en iteraciones. Dichas iteraciones dan como resultado un incremento del producto desarrollado, en el cual tenemos mejoras en las funcionalidades con respecto a la anterior iteración.

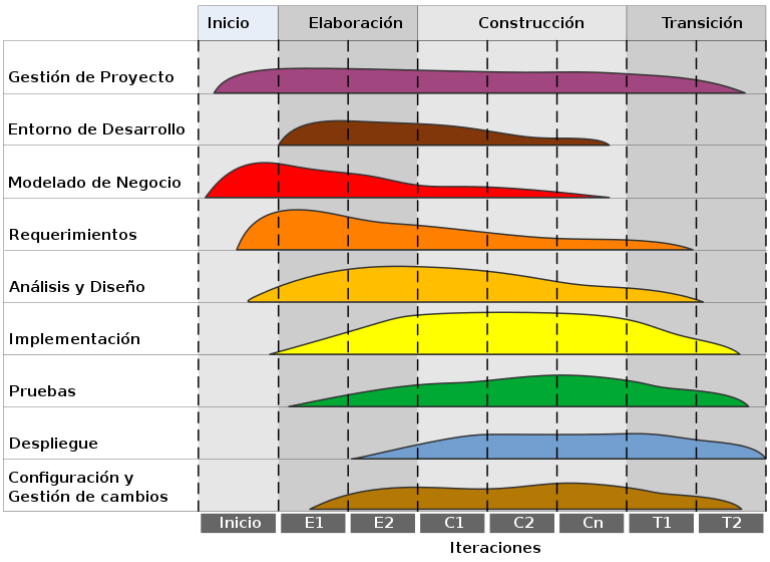


Figura 35 Proceso Unificado

Como se ha indicado anteriormente todo proyecto desarrollado usando el proceso unificado contará de varias fases, generalmente son las siguientes:

- Inicio: Se define el alcance del proyecto y en algunos proyectos grandes puede tener varias iteraciones.
- Elaboración: Se obtiene una visión minuciosa del proyecto que se va a ejecutar. Se añaden nuevos requisitos y se ajustan las estimaciones.
- Construcción: Abarca la evolución hasta convertirse en un producto listo.
- Transición: Es la fase final del proyecto. El producto debe estar listo, funcionando y probado.

Una vez aclarado el marco de trabajo que se va a emplear, se comentará los aspectos más relevantes del desarrollo del proyecto. Se incluirán todas las iteraciones y distintas modificaciones que ha sufrido a lo largo de su desarrollo.

Antes de comenzar con el desarrollo que se ha llevado a cabo es importante destacar los conceptos que se tenían antes de empezar a trabajar en el proyecto.

Inicio

Como se comentará más adelante, la idea de este proyecto parte de un problema real y de la necesidad de solucionarlo. Se plantea un sistema que pueda implantarse en una habitación y que controle distintos aspectos de esta, para que una persona, responsable del usuario, pueda conocer cualquier posible situación potencialmente peligrosa y estar informado. Además de esto, se plantea crear un módulo inalámbrico que permita conocer si una persona ha sufrido una caída y poder comunicárselo al responsable, para que se lleve a cabo una acción lo más rápidamente posible.

Una vez con la idea se plantea el gráfico de la Figura 36.

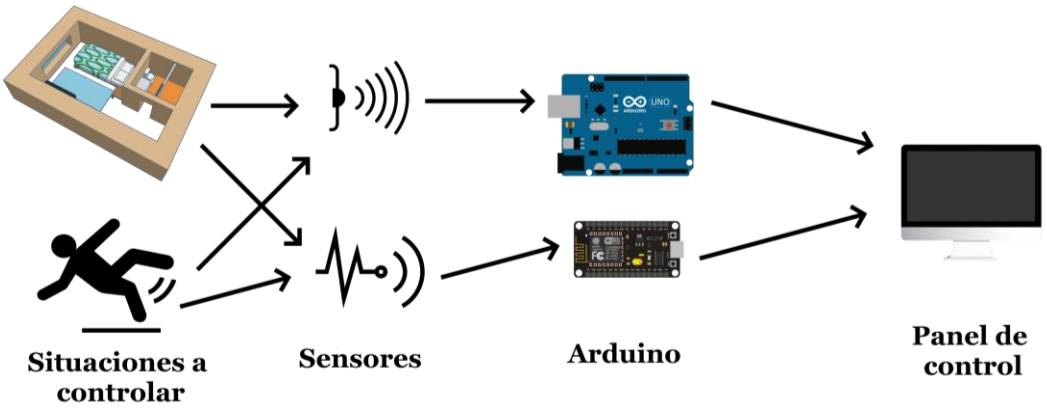


Figura 36 Diagrama de la idea de desarrollo

Desarrollo

A continuación, se introducirá el desarrollo que se ha realizado, con las distintas iteraciones que han tenido lugar, para pasar de la idea inicial al proyecto obtenido al final de la última iteración. En esta sección se indicarán los problemas que se han encontrado, las limitaciones y soluciones planteadas.

En la primera fase se planteó el tema del proyecto, que en este caso nació de una necesidad real. Esto se debe a que un familiar cercano se encontraba en una residencia y debido a que padece demencia, salía de la habitación continuamente. Debido a este problema se me ocurrió la idea de crear un sistema que controle y ayude a las personas que se encuentren en situaciones similares.

Para llevar a cabo este proyecto era necesario el uso de una serie de sensores y poder centralizarlos en algún equipo o sistema para su rápida lectura y actuación.

Con esta premisa se adquirieron una serie de sensores, los indicados en capítulos anteriores, y varias tarjetas Arduino. El modelo elegido al final del proceso fue el ESP-8266. Esta tarjeta permite conexión WIFI, dado que el adaptador viene incluido, y requiere de poca energía para su funcionamiento.

Una vez con todos los sensores y dispositivos necesarios se empezaron a buscar posibles vías para desarrollar el sistema. A continuación, se describirán las distintas iteraciones por las que ha pasado el sistema, así como las razones para elegir o rechazar cada una de ellas. En este apartado se va a mostrar y explicar el código desarrollado para cada una de las implementaciones, para que sea legible y que el contenido no sea abrumador, se ha decido emplear pseudocódigo en lugar de código.

1ª Iteración

En esta fase se planteó lo siguiente:

Usar una de las tarjetas ESP8266 como cliente y otra tarjeta como servidor. La tarjeta cliente se encargaría de recopilar todos los datos mediante los sensores, mientras que la tarjeta servidor se encargaría de almacenar los datos y de mostrarlos a una página web.

Para ello se siguió el siguiente ejemplo [15]. Se realizaron una serie de modificaciones tanto al código como el funcionamiento del sistema. Con ello se consiguió comunicar la información de los sensores mediante envío de peticiones HTTP GET, como se puede apreciar en la Figura 37.

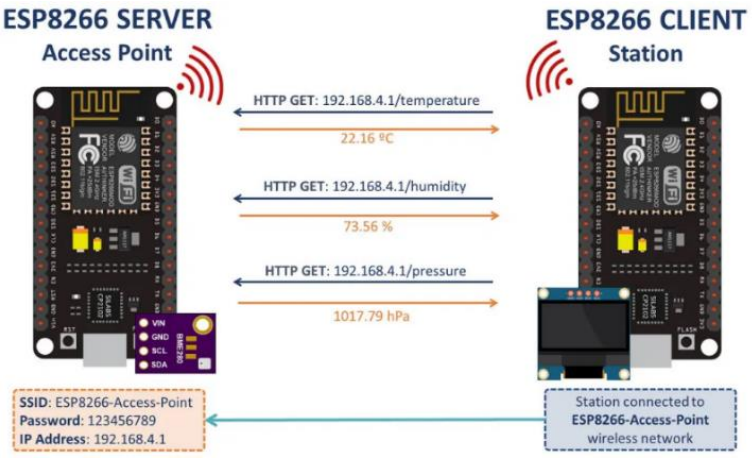


Figura 37 Detalle funcionamiento Cliente-Servidor

Durante el desarrollo de este sistema el pseudocódigo empleado era el siguiente:

```
ESP-8266 Servidor
      Declaración de librerías utilizadas.
      Declaración de variables.
      Declaración de funciones para obtener los datos.
      Configuración
            Elección de la conexión.
            Conexión WIFI.
            Configuración de la tarjeta como punto de acceso.
            Asignación de rutas para la escucha de peticiones.
      Bucle
            //No hacer nada

ESP-8266 Cliente
      Declaración de librerías utilizadas.
      Declaración de variables.
      Configuración.
            Elección de la conexión.
            Conexión WIFI.
      Bucle
            Obtención de datos de los sensores.
            Generación de peticiones HTTP Get.
```

Como podemos observar en el pseudocódigo el cliente se encargaba de obtener los datos y el servidor se encontraba a la escucha de estos envíos. Pese a que este sistema permita el envío de datos lo realiza utilizando el método de envío HTTP GET. Este método muestra los datos que se envían en la URL y se guardan sin cifrar en la caché del servidor. Solo permiten caracteres ASCII y tiene una longitud limitada de 2048 caracteres.

La razón principal para descartar esta idea fue que el servidor era el que tenía que nutrir de datos una página web. Esto exigía un par de tarjetas para cada implementación. Esto no supone un problema para los casos de instalaciones individuales, pero para grandes instalaciones lo hace inviable. Otro factor que no se tuvo en cuenta, pero que posteriormente se resolvió, es la escasa cantidad de puertos analógicos que disponen estas tarjetas. Así como el principal problema de almacenar todos los datos que se van enviando, dado que, sí que existe comunicación y envío de datos entre los dos dispositivos,

pero con la solución presente no podemos recuperar los datos que vamos enviando. Por lo que pese a la existencia de paso de información esta no puede ser utilizada de una manera adecuada.

Dados estos inconvenientes se descarta esta posible solución, teniendo en cuenta que, si nuestro objetivo es enviar datos entre dos dispositivos y mostrarlos en uno de ellos es simple, pero se encuentra muy limitado a posteriores mejoras. Por ello se consideró al siguiente modelo:

2ª Iteración

En esta segunda iteración se solucionaron los problemas que presentaban el sistema anterior. Para ello se siguió el modelo que podemos ver en la Figura 38.

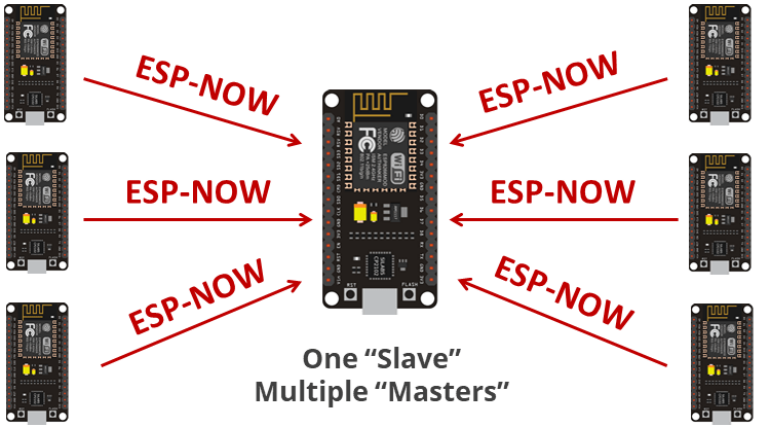


Figura 38 Detalle funcionamiento un servidor varios clientes

Siguiendo la guía [16] obtenemos un nuevo sistema. Para ello se utilizaría un único servidor y múltiples clientes. Siguiendo un procedimiento similar al anterior, podemos conseguir que el problema de tener un par de tarjetas por cada estancia se reduzca, ya que con este sistema podremos controlar una gran cantidad de habitaciones usando un único servidor.

En este caso en vez de realizar las conexiones usando HTTP GET, se van a realizar envíos usando la dirección MAC (Media Access Control, Control de Acceso a los Medios) de cada dispositivo. La dirección MAC es generalmente asignada cuando un dispositivo es creado y, generalmente y a diferencia de las direcciones IP, no cambian cuando se mueven entre redes, es decir, se consideran estáticas. Para ello se emplea el siguiente código:

```
Obtención de dirección MAC
Declaración de variables
      Configuración.
      Inicialización de comunicación.
      Imprimir MAC de cada equipo.
```

Mediante este fragmento de código podremos obtener cada una de las direcciones MAC de los dispositivos que queremos emplear en la comunicación. A continuación, habrá que desarrollar código para las tarjetas que realizan el envío y para la que hace de servidor.

ESP-8266 Cliente

```
Declaración de librerías utilizadas.
Declaración de variables.
Configuración
      Elección de la conexión.
      Conexión WIFI.
      Registro del identificador único de cada tarjeta.
Bucle
      Envío de datos.
```

ESP-8266 Servidor

```
Declaración de librerías utilizadas.
Declaración de variables.
Declaración de funciones para obtener los datos.
Configuración
      Elección de la conexión.
      Conexión WIFI.
      Configuración de la tarjeta para la escucha de mensajes.
Bucle

      //No hacer nada
```

La biblioteca ESPNOW nos permite enviar información desde diversas tarjetas y recibirlas desde una única actuando como servidor.

Pese a todas las ventajas que presenta esta solución frente al caso anterior, siguen existiendo problemas al tener varias tarjetas conectándose a un único servidor. Según la documentación [17] de ESPNOW el máximo de tarjetas soportadas por servidor es de veinte, por lo que no es altamente escalable. Además, como únicamente se mandan datos desde una única tarjeta por estancia, el número de sensores sería limitado, por lo que habría que emplear varias tarjetas por estancia, reduciéndose así las ventajas que presenta. También hay que tener en cuenta el almacenaje de la información por parte del servidor y cómo va a ser mostrada al cliente.

Como se indicó en los objetivos del sistema, se requiere un sistema altamente escalable y que muestre la información de una manera rápida y concisa. Con la solución que se plantea no es posible realizar una instalación para más de veinte equipos, se podría subsanar el problema empleando múltiples servidores, pero debería buscarse el modo de centralizar los datos entre todos los servidores y la comunicación entre ellos. Como esta iteración genera más problemas, se decide buscar otra posible solución para las dificultades de escalabilidad y comunicación.

3ª Iteración

Para esta iteración se decidió dar un paso atrás y en vez de enviar los datos a un servidor usando una única tarjeta como servidor, se decidió usar un sistema externo para recibir los datos. Para ello se siguió el ejemplo [18]. En esta guía se explica como enviar datos desde una tarjeta ESP8266 a una base de datos. Esta última puede estar creada tanto en un ordenador externo como en una tarjeta Raspberry.

Otra de las ventajas que permite este método es el de mostrar rápidamente en una página web los datos que se han recolectado mediante los sensores. Como se puede ver por la Figura 39, los sensores recopilan información, envían una petición HTTP POST con toda la información recopilada en un determinado periodo. Posteriormente el servidor, en este caso un ordenador, recibe los datos mediante un servidor y, usando un script PHP, se almacenan los datos en una base de datos relacional. Posteriormente con otro script PHP se recorren las tablas con los datos y se muestran en una página web para el posterior análisis y visualización del cliente.

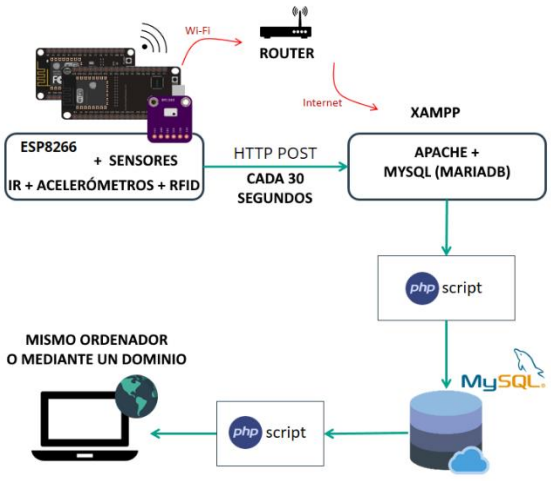


Figura 39 Detalle de funcionamiento Arduino - Servidor

Previo al desarrollo del código es necesario instalar y configurar MySQL, se puede realizar tanto en un equipo local como en un servidor web que permita esta tecnología. En el caso de esta implementación se decidió emplear XAMPP. Cuando se encuentre instalada la base de datos se deberá crear el sistema de tablas para almacenar la información del proyecto. También habrá que desarrollar los scripts de PHP para pasar los valores a la base de datos. Los scripts tendrán un aspecto similar a este:

Script PHP para insertar información en la base de datos

```
Declaración de variables de conexión.
Declaración de credenciales de conexión.
Si se produce una petición POST
      Obtención de los datos de la petición.
      Conexión con la base de datos.
            Consulta para añadir los datos.
      Terminación de conexión.
Si no se produce petición POST
      //No hacer nada.
```

Script PHP para mostrar los datos en el un servidor web

```
Declaración de variables de conexión.
Declaración de credenciales de conexión.
Crear conexión con la base de datos.
Creación de consulta.
Si la consulta retorna valores
      Almacenamiento temporal en variables para cada elemento en forma
      de tabla.
      Mostrar los elementos en el archivo HTML.
      Cerrar conexión.
Si la consulta no retorna valores
      Cerrar conexión.
```

Con estos dos documentos podremos introducir y recuperar información en una base de datos. Ahora faltará desarrollar el código Arduino para enviar los datos al servidor:

```
ESP-8266 Cliente
      Declaración de librerías utilizadas.
      Declaración de variables.
      Declaración de variables de conexión al servidor.
      Configuración
            Elección de la conexión.
            Conexión WIFI.
      Bucle
            Si existe conexión WIFI
                  Inicialización de conexión HTTP.
                        Recopilación de datos.
                        Creación de consulta HTTP POST.
                        Comprobación de errores.
                  Cierre de conexión HTTP.
            Si no existe conexión WIFI
                  Mostrar mensaje.
            Retraso de dos segundos.
```

Con este código podemos enviar los datos que recopilen los sensores cada dos segundos. La petición HTTP POST llegará a nuestro servidor. Desde aquí podremos mandar la información mediante uno de los scripts anteriormente descritos. Para llevar la información al cliente usaremos el otro script para consultar la base de datos.

Como podemos ver se han solucionado gran parte de los problemas presentes en las anteriores iteraciones. Por ejemplo, con los nuevos cambios los datos quedan almacenados de manera segura en una base de datos. Otra ventaja es que se están empleando peticiones de tipo POST. Este tipo de envío de información es más seguro, permite datos binarios, es invisible para el usuario la conexión que se está realizando, además de posibilitar un envío ilimitado de datos en cada petición.

Pero no todo son ventajas, dado que se precisa un servidor central que contenga la base de datos, la escalabilidad se ve mermada. Además, como las peticiones deben procesarse mediante scripts y, si se realiza con mucha asiduidad el envío de mensajes, puede hacer que equipos de poco rendimiento, o que las tarjetas tipo Raspberry no puedan soportar la carga a la que se les está sometiendo. Otro problema es el uso de una base de datos relacional, dado que al emplear este sistema se debería crear una tabla para cada una de las estancias, con la consecuente complicación en la creación y mantenimiento, tanto de la base de datos como de los scripts. También hay que tener en cuenta que,

como las peticiones se realizan mediante un script PHP, la funcionalidad de la página puede quedar limitada. Pero, sin lugar a duda, el aspecto más problemático de este enfoque es que el acceso a los datos se encuentra limitado únicamente para el equipo que haga de servidor o para equipos de la misma red. Si se plantea hacer conexión desde cualquier otro lugar serán necesarios conexiones mediante VPN o conexiones remotas.

Otra vertiente para solucionar estos problemas sería emplear un servidor externo y un nombre de dominio para conectarse a este. Si se quisiera llevar a cabo el sistema utilizando esta tecnología habría que realizar una gran tarea de encriptación y protección de los datos, pero esta solución es viable. Pese a esto, se ha decidido explorar ideas que permitan realizan las conexiones de manera más eficientes y seguras, así como no precisar de servidores externos o locales mediante conexiones remotas que dificulten el proceso.

Por todas estas razones se descarta este sistema y se busca una nueva solución que permita una escalabilidad indefinida, bajo coste y que posibilite una lectura rápida de los datos independientemente de la cantidad de estancias que existan.

4ª Iteración

Antes de comenzar con este apartado de la memoria es importante enfatizar sobre ciertas características del sistema. Se propone un sistema seguro, confidencial y que no permita a terceros el acceso a datos personales. Para ello se emplea la tecnología disponible en Google Cloud. Permite en todo momento encriptar tanto los datos que se envían como los que se almacenan, así como proteger los accesos mediante cuentas de usuario.

Durante el desarrollo del prototipo del sistema se ha decidido no emplear encriptación con el fin de mostrar de una manera adecuada y fácil de entender los datos que se están procesando y almacenando. En el caso de querer comercializar este producto, se realizarían las tareas necesarias para conseguir los niveles de seguridad necesarios.

Otro punto importante es que el panel de control se va a destinar únicamente a un equipo que permita el control, pero como se ha indicado anteriormente, al estar gestionado y controlado por las cuentas de Google no es necesario generar mayor control. Para posteriores prototipos y desarrollos se puede plantear la gestión de usuarios y control de accesos, pero dado la naturaleza mono puesto del servicio, se ha decidido no llevarla a cabo.

En este nuevo modelo se plantea el esquema de la Figura 40. La tarjeta ESP-8266, por medio de los sensores que recogerán información, tendrá conexión con el servicio Cloud IoT Core de Google. Para esta conexión únicamente se precisará de una red WIFI.

Figura 40 Detalle de funcionamiento de Google IoT Core

Tras enviarle los datos, la plataforma se encargará de enviarlos al sistema de colas Pub/Sub, como se indicó en el apartado 3.6.1 Funcionamiento. Posteriormente se utilizaría una Cloud Functions que se encargue de enviar estos a una base de datos. Con este método de trabajo se solucionan los problemas anteriores, dado que se eliminan las restricciones por número de dispositivos, ya que IoT Core permite una escalabilidad infinita. También se soluciona los problemas de acceso a los datos, ya que se emplea una base de datos no relacional basada en documentos y que es accesible con simple JavaScript desde cualquier parte del mundo. Esto permite que podamos estar controlando la estancia desde cualquier lugar. Para llevar a cabo este enfoque se ha seguido la guía oficial para conexión de dispositivos IoT de Google, disponible en el GitHub oficial [19]. Durante el proceso de creación de los dispositivos no existieron problemas reseñables, debida a la claridad de la guía. Al finalizar el trabajo con ella se tenía un sistema que enviaba mensajes desde una tarjeta ESP8266 hasta el sistema de colas Pub/Sub.

Siguiendo la guía oficial del GitHub [19] se nos presentan tres documentos: Un archivo de código Arduino y dos archivos de cabeceras que deberemos modificar acorde a nuestros requerimientos. Con el fin de clarificar la memoria, únicamente se describirá el código de Arduino, dejando la descripción del resto de documentos para el Anexo IV. Documentación Técnica de Programación.

El código empleado para esta solución es el siguiente:

```
Declaración de variables
Inicialización (que se ejecutará una única vez)
      Inicialización de la comunicación serie.
      Llamada a la creación de variables globales para MQTT.
Bucle (que se ejecutará ininterrumpidamente después de la
inicialización)
```

```
        Conexión MQTT
        Retraso de 10 milisegundos para establecer la conexión.
    Código para la publicación de telemetría.
```

Empleando este código conseguiremos publicar la información que recopilen los sensores y llevarla hasta el servicio de colas Pub/Sub.

Entonces sería ahora el momento de obtener estos datos y almacenarlos en la base de datos Firestore. Fue en este instante cuando empezaron los problemas y complicaciones. Esto se debe a que, pese a que exista una amplia guía y detallados videos explicando cómo realizar la conexión y envío de datos desde las tarjetas a IoT Core, no es así para el caso del desarrollo de una Cloud Functions que almacene datos en Firestore. Se entiende que es porque son casos muy específicos con unos datos muy particulares. Para otras plataformas y sistemas de almacenajes sí que existen guías. Pese a ello se decidió crear una Cloud Functions usando Node.js. La función fue realizada en los servidores de Europa, para así reducir la posible latencia en el envío y recepción de datos. Tras implementarla y solucionar los problemas se llegó a la conclusión de que el sistema era incapaz de detectar la función por lo que los datos no llegaban a ella. Después de un largo periodo de pruebas se consiguió solucionar este problema mediante la creación de la función en el servidor de Estados Unidos, usando la consola de desarrollo del SDK de IoT Core. Tras realizarlo de esta manera, la función empezó a recibir los datos y almacenarlos en los documentos asignados.

El código Node.js que se ha utilizado en la Cloud Functions es el siguiente:

```
Declaración de variables.
Inicialización de la aplicación y de la función.
Exportar información de Pub/Sub.
Obtención de información del mensaje JSON.
Tratamiento de la información.
Publicación de la información en las colecciones y documentos
pertinentes.
```

Con este código podemos obtener la información de Pub/Sub y transferirla a la base de datos para que posteriormente sea consultada fácilmente.

Empleando como referencia la Figura 41 se va a explicar en detalle el funcionamiento del sistema. Los sensores recopilan información del medio, en este caso la estancia en la que se encuentren instalados. Esta información será procesada por la tarjeta ESP-8266. Dependiendo de las lecturas obtenidas por los sensores la tarjeta enviará un mensaje en formato JSON con la acción que ha

ocurrido. Esta acción se envía mediante MQTT a Google Cloud IoT Core. Los mensajes se irán almacenando en una cola temporal. Entre los mensajes que se encuentran en esta cola temporal encontramos tanto los mensajes que se han enviado como los propios que usa MQTT para establecer conexión o los mensajes de control.

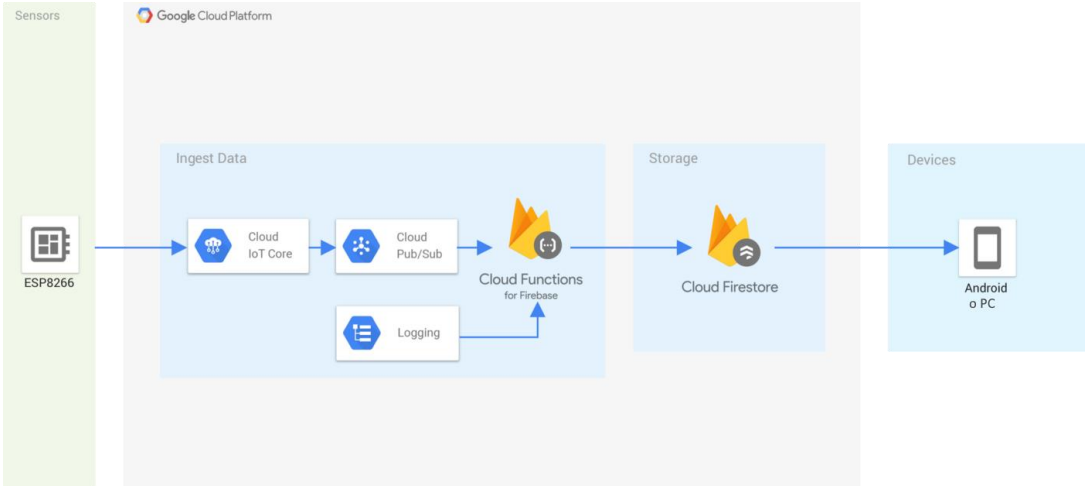


Figura 41 Envío de mensajes desde el dispositivo a la base de datos

De esta cola temporal pasan al sistema de colas de Pub/Sub. Aquí únicamente pasarán los mensajes que hemos enviado desde las tarjetas. Los mensajes de control serán filtrados. Ahora con los mensajes filtrados se pasará a reenviarlos al tema en concreto. El sistema leerá cada mensaje, uno tras otro, y los irá distribuyendo. Cuando lleguen al tema de la subscripción en concreto se almacenan hasta ser leídos por el sistema de Cloud Functions.

Este servicio se encargará de ir leyendo uno tras otro los mensajes que se encuentran en la subscripción y los procesará. Para ello, usando código Node.js, se accederá a los datos que se enviaron en el mensaje y a la metainformación del mensaje (fecha y hora del envío). Una vez leídos los datos se creará una estructura en JSON con el formato deseado. En este caso indicando el sensor que se ha activado, la acción que la ha activado y la hora a la que se ha producido la acción. Con la estructura formada se almacenarán los datos en dos colecciones distintas en FireStore. La primera será la colección correspondiente a la estancia en particular y la segunda el histórico con todos los mensajes enviados.

Tras finalizar este proceso tendremos los datos que se han leído en los sensores almacenados y procesados en una base de datos no relacional que se pueda acceder de manera segura desde cualquier lugar del mundo.

El siguiente desafío era obtener la información de la base de datos y mostrarla adecuadamente en una página web. Existe amplia documentación [20] sobre cómo realizar este apartado usando distintas tecnologías, como son Swift, Flutter o Node.js. Pero existe un problema, dado que se quiere tener total descentralización del sistema y que este pueda ser accedido desde cualquier dispositivo, se descarta la idea de utilizar Node.js como servidor para obtener esta información. Por ello el código para la realización de las consultas y mostrado de datos se llevarán a cabo mediante JavaScript. Por esta razón se aumentó la dificultad del sistema, dando lugar que parte del código encontrado en la documentación fuese inservible. Llegados a este punto, se realizó una amplia tarea de pruebas hasta conseguir el código necesario para obtener la información de la base de datos.

El código JavaScript para realizar estas funciones es el siguiente:

```
Importación de bibliotecas y funciones de Firebase/FireStore.
    Configuración de FireStore.
    Declaración e inicialización de variables.
    Funciones para el tratamiento de información.
        Función para la obtención del último dato de la base de datos.
            Tratamiento de información.
    Función para recolectar información cada periodo.
    Control de la funcionalidad de los botones.
```

Una vez conseguida la forma de realizar estas consultas se pasó a efectuar el panel de control para mostrar estos datos. Se partió de una página web desarrollada para la asignatura Interacción Persona Ordenador, cursada durante la carrera, que se puede observar en la Figura 42.

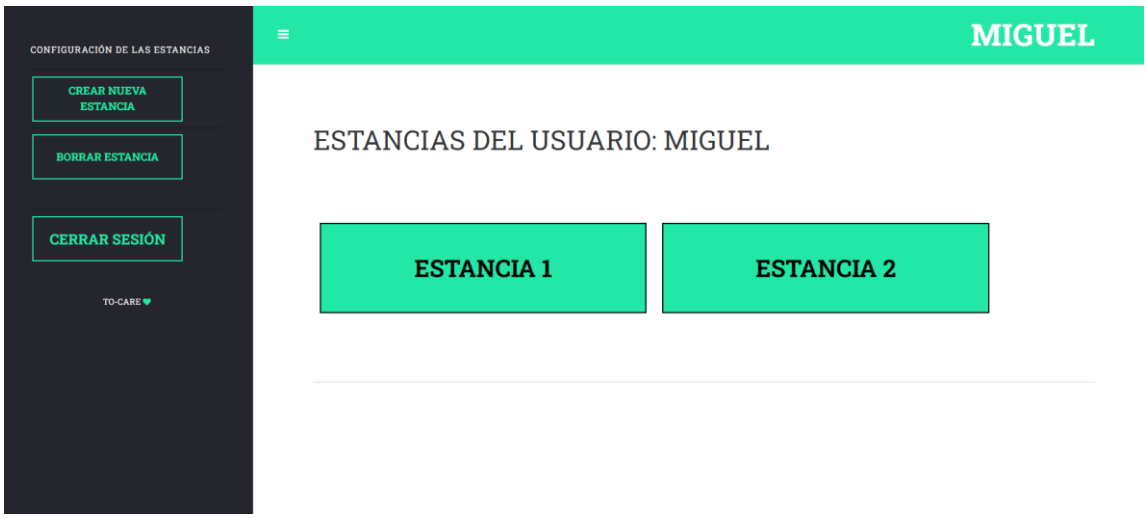


Figura 42 Página principal

En esta página se mostraban las diferentes estancias asignadas y se recuperaba la información haciendo clic sobre cada una de ellas. Obteniendo la información como se muestra en la Figura 43. Cabe destacar que esta página web era parte del proyecto final de la asignatura. En él había que realizar una interfaz imaginada. El trabajo versaba sobre una tarjeta que con conectarla a la red era capaz de obtener toda esta información del ambiente. Por ello estas figuras son orientativas.

Figura 43 Detalle estancia

Pese a todos los avances que se habían conseguido con este nuevo enfoque, se descubrió un problema importante que debía ser solucionado. Las tarjetas ESP8266 presentan un único conector para lectura de datos analógicos. Esto, en condiciones normales, no presenta ningún problema. Pero en este caso, y dado que es preciso obtener información analógica de los sensores de las fotorresistencias y de los sensores de presión, hacían inviable esta solución. Además, es importante recalcar que la página web desarrollada presentaba la información, pero de una manera poco visual. Por esta razón se decidió realizar cambios también en este apartado.

5ª Iteración

Antes de comenzar con los cambios, se desarrolla el aspecto del sistema. Como podemos ver en la Figura 44 los sensores se conectan a la tarjeta ESP-8266 y esta enviaba los datos a IoT Core.

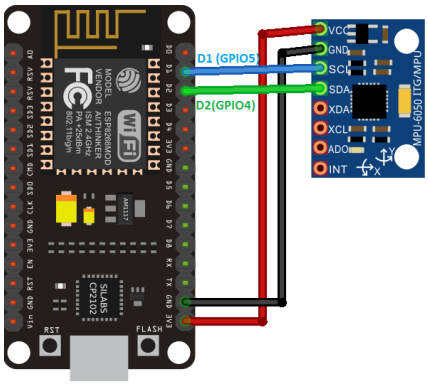


Figura 44 Detalle tarjeta y sensor empleado

Como se indicó en la iteración anterior, esto supone un problema dada la existencia de una única entrada analógica en la tarjeta. Para solucionar este problema se podían optar por dos posibles soluciones:

La primera consistía en utilizar un multiplexor de señales analógicas como el que se puede ver en la Figura 45.

Figura 45 Multiplexor de señales analógicas

Este enfoque permite utilizar hasta 16 dispositivos analógicos con tan solo una entrada. Pese a que esta solución funciona, y es útil, se descartó porque requería complicar la lógica del código debido al uso del multiplexor, la adaptación de corriente para los sensores a otro voltaje (usando un conversor de 5V a 3.3V) y por último la obligatoria adquisición de este nuevo dispositivo. Por estas razones se descartó esta solución y se buscó un nuevo enfoque.

Dado que cuando se realizó la compra de todos los dispositivos también se adquirió un Arduino Uno se decidió probar con la conexión entre las dos tarjetas para solucionar el problema.

Para realizar la conexión se realizó una investigación empleando recursos en línea [21]. Por lo general, se puede establecer conexión con cualquier par de pines, aunque es recomendable realizarlo usando los pines destinados a ello, RX y TX.

Estos dos pines son los destinados a la conexión serie entre la tarjeta y cualquier equipo que se le conecta. El pin RX lleva los datos y el pin TX la señal de reloj.

Un detalle importante a tener en cuenta para esta conexión es que los pines TX y RX deben de estar conectados a la inversa como se puede apreciar en la Figura 46.

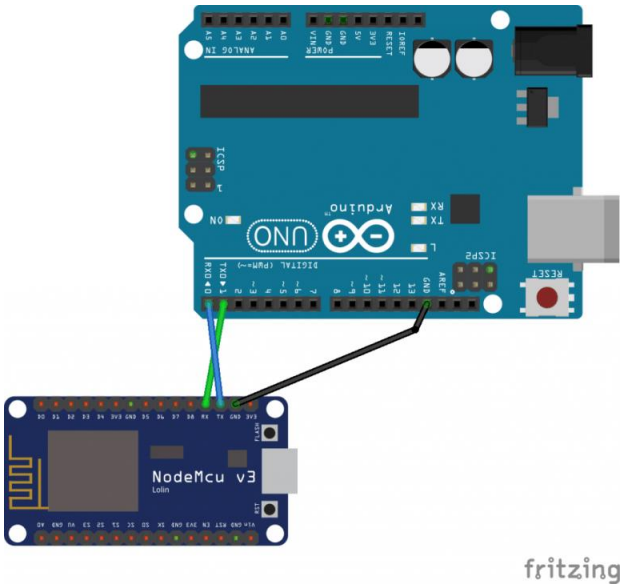


Figura 46 Conexión entre Arduino Uno y ESP8266

Si no se realiza así, es imposible que exista conexión. Otro aspecto relevante es unir la línea de tierra entre los dos dispositivos. También cabe recalcar que, si se quiere actualizar el código instalado en las tarjetas, esta conexión cableada no puede estar realizada. Para poder pasar el código debemos deshacer la conexión y posteriormente rehacerla. Estos aspectos son de alta importancia para la comunicación serie y son motivo de muchos problemas si no se controlan adecuadamente. El aspecto más relevante es la conexión entre tierras, dado la diferencia de voltaje entre el Arduino Uno y la tarjeta ESP8266.

Teniendo en cuenta estos factores, y después de solucionar los problemas, se consigue la comunicación entre las dos tarjetas. Con esto el Arduino Uno puede recibir todos los datos, procedentes de las lecturas del entorno de los sensores, y se los transmite de una manera rápida y ágil al módulo ESP8266. Este dispositivo, posteriormente, los hace llegar a IoT Core y, por ende, al panel de control del sistema.

El código para la comunicación será distinto, ya que primero necesitaremos recopilar los datos en la tarjeta Arduino Uno y posteriormente enviarlos mediante la ESP-8266. A continuación, se mostrará un pseudocódigo explicando este proceso:

Código Arduino Uno

```
Declaración de variables.
Configuración
      Conexión serie y conexión de SoftwareSerial.
      Inicialización de los distintos modos de los pines de la placa.
Bucle
      Lectura de los sensores.
      Comprobación de los valores
            Si superan la comprobación se envía documento JSON mediante
            comunicación serie al ESP8266.
            Si no superan la comprobación se repite lecturas.
            Retraso de 500 milisegundos entre lecturas del bucle.
```

Código ESP-8266

```
Declaración de variables.
Configuración
      Inicialización de la comunicación serie.
      Llamada a la creación de variables globales para MQTT.
Bucle
      Creación de documento JSON.
      Conexión MQTT.
      Retraso de 10 milisegundos para establecer la conexión.
      Lectura de la conexión serie.
      Comprobación de la información recibida.
      Código para la publicación de telemetría.
```

Con estos dos fragmentos de código tenemos un sistema similar al que se generó en la iteración anterior, con las modificaciones para permitir la comunicación entre las dos tarjetas.

Con esto ya tenemos solucionados los últimos problemas presentes con la anterior solución. Pasamos entonces a comentar el resto de los procesos que se realizaron durante el desarrollo.

Cabe destacar que el desarrollo del módulo inalámbrico, con botón del pánico, se implementó directamente sin ningún problema, ya que presenta un código y un circuito más simple. Además, sigue el mismo patrón de funcionamiento que el expuesto anteriormente.

En cuanto al desarrollo del panel de control web se tuvo en cuenta los cambios necesarios con el modelo anterior. Se mostraba la información, pero se realizaba de manera poco visual.

Se decide solucionar este problema realizando un modelo de la habitación mediante HTML y CSS. Para ello se parte de la Figura 47.

Figura 47 Habitación de referencia

Como se indicó en apartados anteriores se llegó al diseño de la Figura 48.

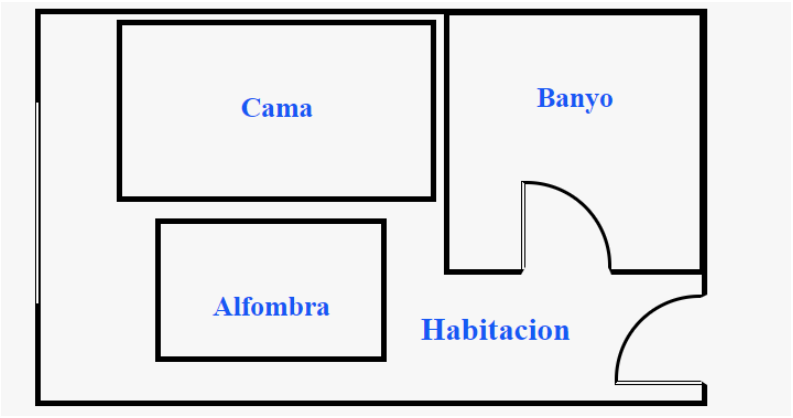


Figura 48 Primera iteración del diseño de la estancia

Teniendo en cuenta este modelo se decidió mejorar el aspecto visual de la página, realizando cambios tanto en las hojas de estilos, como en la funcionalidad con JavaScript. Esto último permite actualizar el estado de la habitación según las situaciones que estén dándose dentro de ellas. Los cambios, sobre todo visuales, se pueden apreciar en la Figura 49.

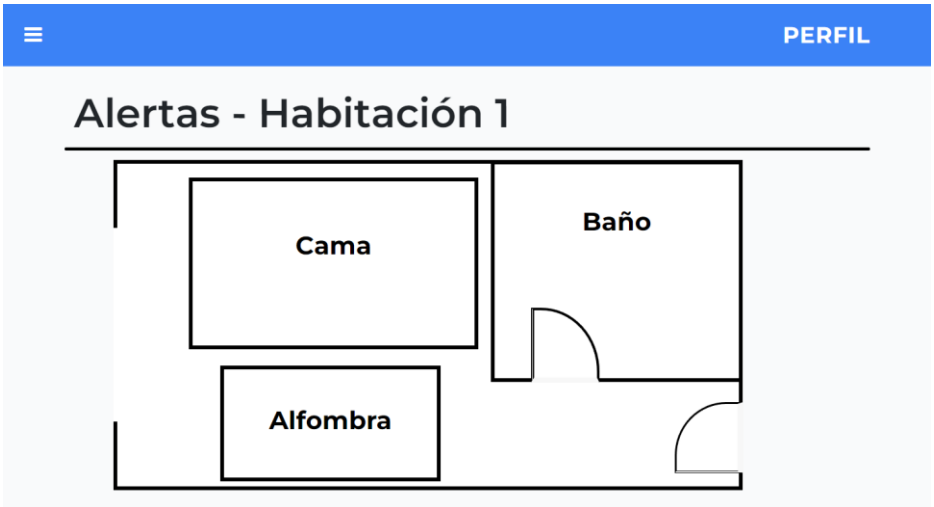


Figura 49 Diseño final de la habitación en la plataforma web

Ahora la habitación cuenta con una descripción de cada aspecto de esta, así como un sistema de alertas que indican la gravedad de cada incidente. Esto permite de una manera visual y clara conocer el estado de la habitación, como se puede apreciar en la Figura 50. Todo este funcionamiento se explicará en detalle posteriormente.

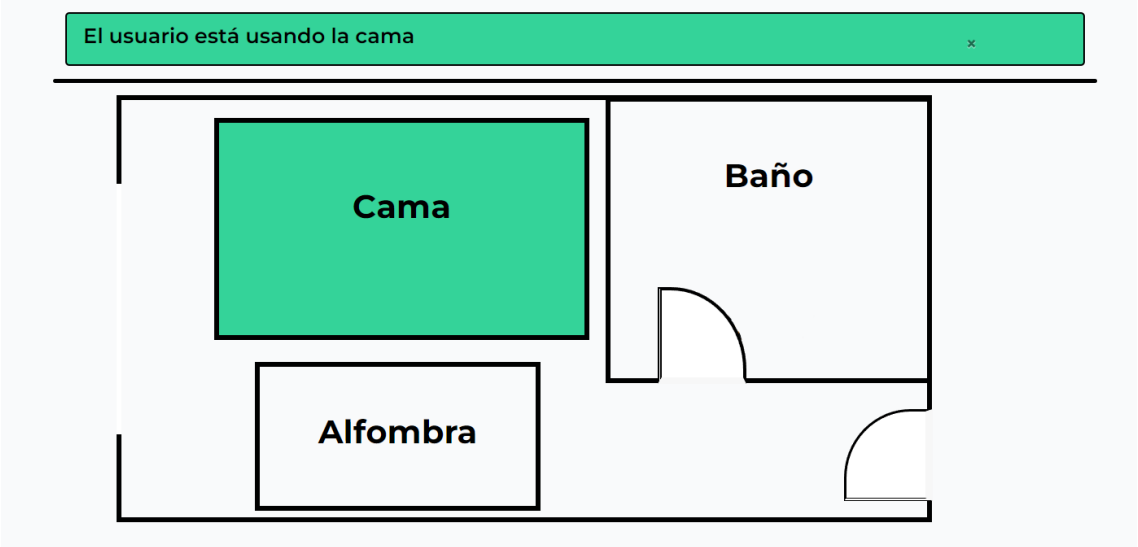


Figura 50 Detalle funcionamiento del sistema de alertas

El código para este apartado será similar al de la iteración anterior, añadiendo las modificaciones estéticas y el sistema de alertas, dadas las similitudes se omite en este apartado, pero será explicado y desarrollado en el Anexo IV. Documentación Técnica de Programación.

El siguiente reto era adaptar el panel de control para una gran cantidad de habitaciones, permitiendo conocer el estado de ellas con un simple vistazo. Para ello se implementó la siguiente interfaz (Figura

51), en ella podemos ver todas las estancias que tenemos dadas de alta. En este caso son diez, pero pueden ser más, o incluso distribuirse por plantas.

Figura 51 Plataforma con 10 habitaciones configuradas

En la Figura 52 vemos los distintos eventos que suceden en las estancias, quedándonos claro los distintos estados y alertas que están sucediendo en ellas.

Figura 52 Sistema de alertas con 10 habitaciones

El código para este sistema es similar al presente para una única estancia, con una serie de modificaciones conseguiremos obtener información de todas las estancias y modificar estéticamente la página, como se puede ver en la Figura 52.

En cuanto al circuito que se ha creado para el funcionamiento se puede apreciar de manera esquemática en la Figura 53.

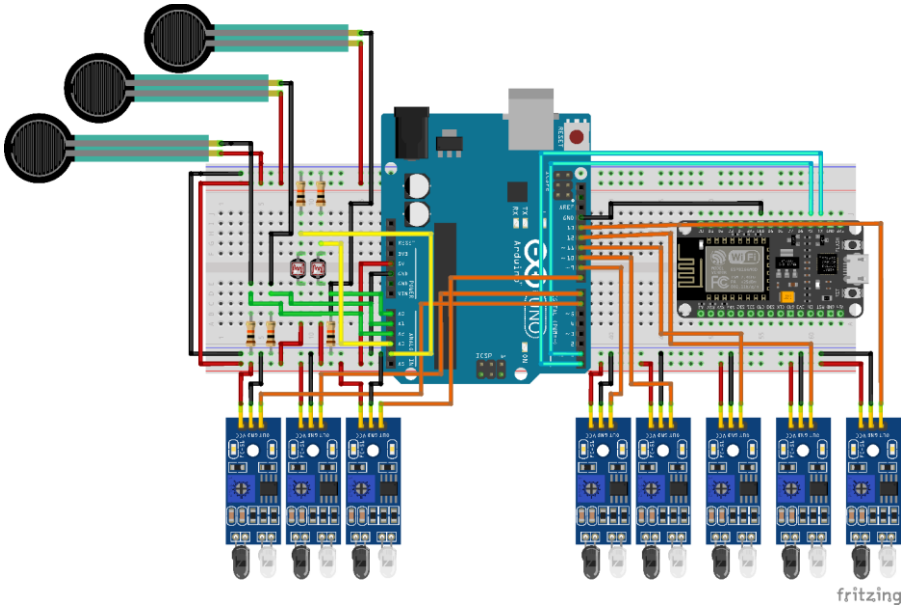


Figura 53 Esquema circuito habitación.

Este circuito se ha desarrollado en una maqueta, que se verá más adelante, pero para las pruebas se utilizó un prototipo que se puede apreciar en la Figura 54:

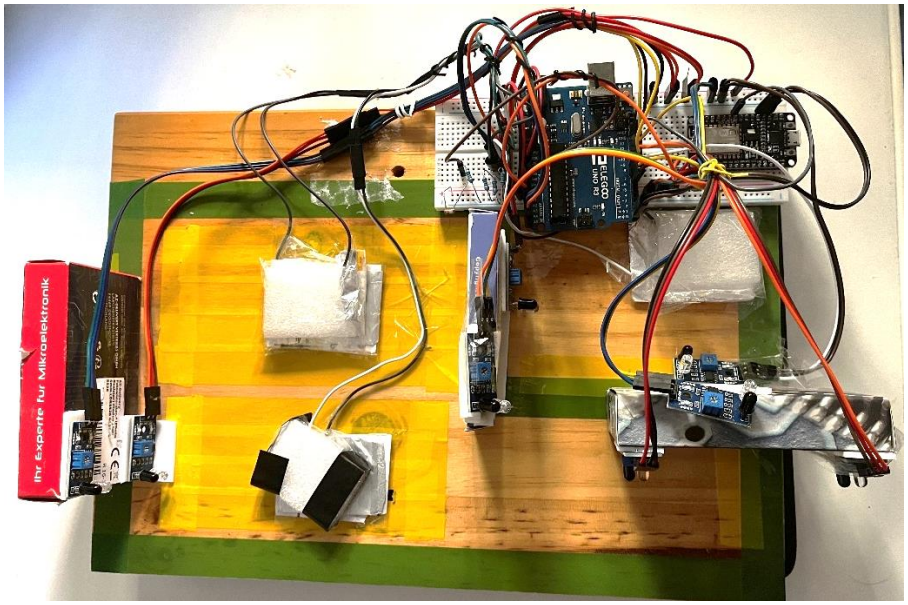


Figura 54 Circuito desarrollado en el primer prototipo.

Este prototipo ha servido para realizar las pruebas y comprobar el funcionamiento del circuito planteado.

En cuanto al módulo inalámbrico se planteó una idea similar al de la habitación, empleando un único ESP-8266 para enviar los datos como se ve en la Figura 55.

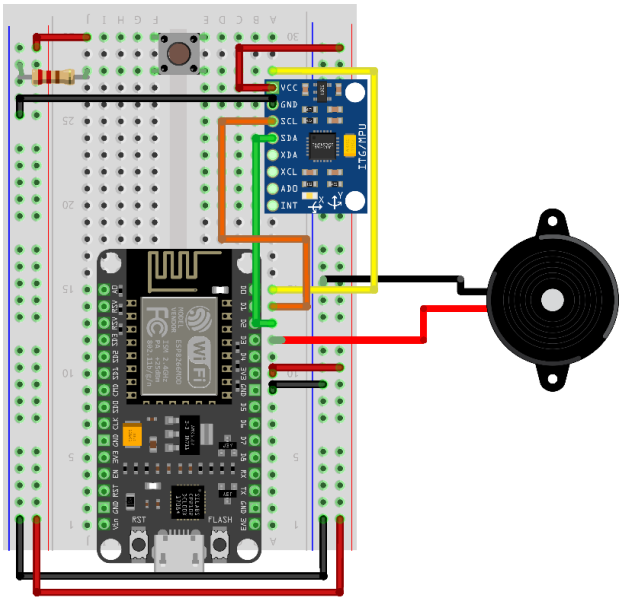


Figura 55 Módulo inalámbrico

Este dispositivo cuenta de un zumbador, un botón del pánico y la tarjeta ESP-8266. Su implementación la podemos ver en la Figura 56.

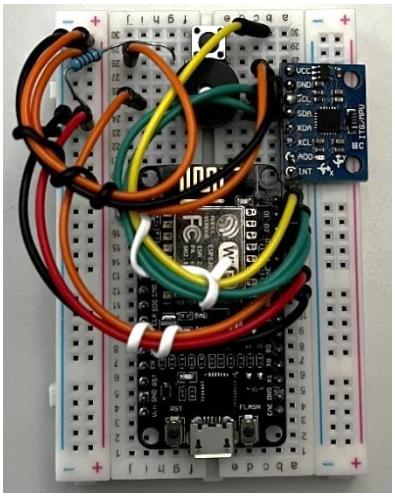


Figura 56 Implementación módulo inalámbrico

Estos circuitos se podrán apreciar de una manera más detallada en el Anexo III. Especificación de diseño. Con estos dos dispositivos tendremos un sistema capaz de supervisar una habitación y detectar situaciones de riesgo.

Maqueta 3D

Como se ha comentado en capítulos anteriores, para el desarrollo del prototipo de este proyecto se ha creado una maqueta con ayuda de FabLab, de la Universidad de Salamanca. FabLab USAL es un laboratorio de creación y fabricación digital, enfocado al diseño de objetos con tecnologías digitales que está abierto a toda la comunidad universitaria. Para ello, dispone de un equipamiento técnico que aúna las características de los talleres digitales y tradicionales. Entre el equipo del que dispone FabLab se puede encontrar una máquina de mecanizado CNC (Computer Numerical Control, Control Numérico por Ordenador), así como varias impresoras de impresión 3D por filamento fundido.

Tras una serie de reuniones con los responsables de FabLab, se llegó a un acuerdo por el cual el alumno realizaría las maquetas que posteriormente sería impresas usando el equipamiento que se dispone en el laboratorio.

Para este desarrollo se me indicó buscar modelos disponibles en las muchas páginas web destinadas a compartir diseños 3D. Una vez encontré uno similar al deseado realicé las modificaciones necesarias con un programa destinado a ello. A continuación, se verán los distintos seleccionados y los cambios realizados para cumplir los requisitos impuestos.

Diseño habitación

Antes de comenzar con la creación de la habitación se plantearon diversos modelos. Entre todos ellos se consideró uno similar al de la Figura 57.

Figura 57 Diseño de habitación usada como referencia

Se decidió simplificar el modelo eliminado todos los elementos sobrantes quedando un diseño similar al que se puede ver en la Figura 58.

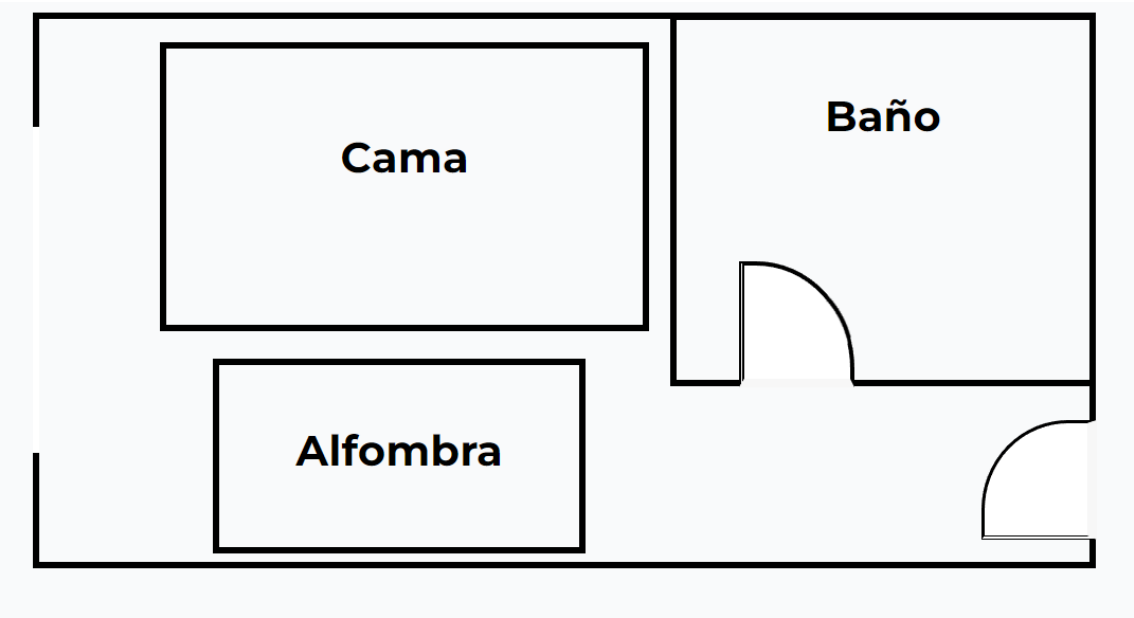


Figura 58 Diseño adaptado de la habitación

Una vez elegido el diseño, como punto de partida para el modelo de la estancia, se decidió usar uno disponible en Tinkercad dado que tenía un alto grado de parecido al que se tenía en mente para la estancia. Esta tiene que estar compuesta de ciertos elementos, una alfombra grande, una única cama y un baño. Como podemos observar en la Figura 59, el modelo encontrado es similar al que se busca.

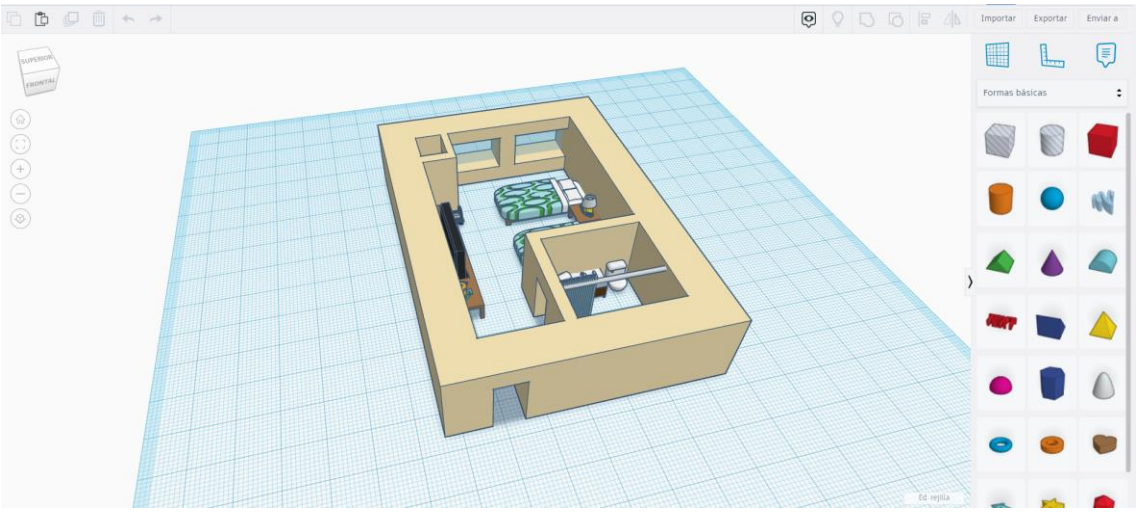


Figura 59 Diseño de referencia de Tinkercad

A partir de este momento, se comenzó a realizar distintos cambios al modelo hasta obtener el deseado.

Se plantea este primer rediseño, eliminando los elementos sobrantes y añadiendo los detalles necesarios.

El siguiente paso fue reducir todos los elementos sobrantes para obtener únicamente la maqueta de la habitación. Además, se ahuecaron los muros para así poder introducir sensores y cables en las paredes como se ve en la Figura 60.

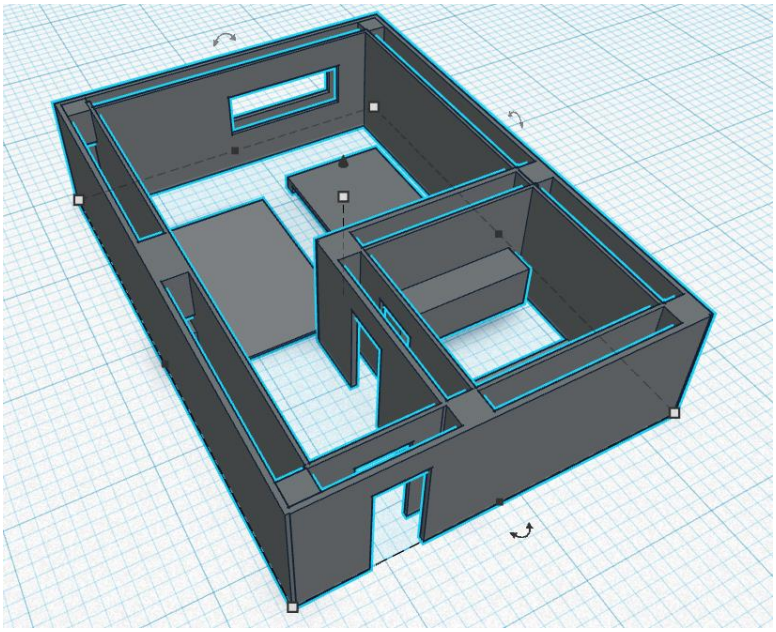


Figura 60 Segundo diseño del modelo

Por último, se ajustó a escala y se crearon los orificios para poder instalar los sensores. Quedando el modelo de la Figura 61:

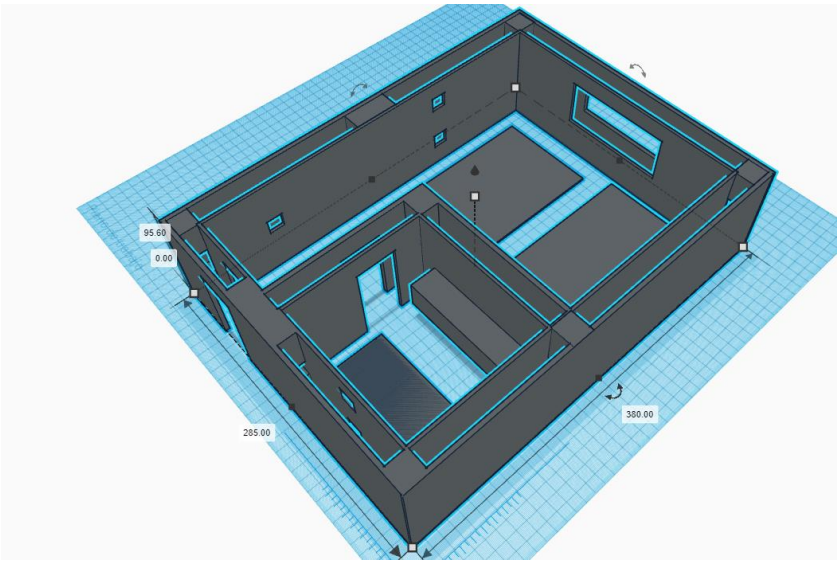


Figura 61 Diseño final de la estancia

A continuación, se imprimió usando el programa Meshmixer que permite cortar y modificar diseños en 3D. Para que entre dentro de la zona de impresión de la impresora 3D, dado que el modelo actual no es posible imprimirlo de una vez. Se decide cortar en cuatro fragmentos de similar tamaño. Obteniendo cuatro secciones como se puede apreciar en la Figura 62.

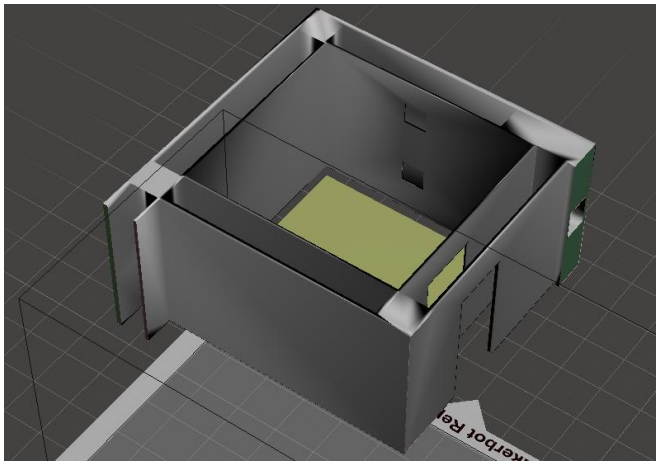


Figura 62 Detalle del cuarto de baño

Para comprobar que todas las partes son imprimibles según los cortes realizados usamos el programa PrusaSlicer. Este programa es el propio de las impresoras disponibles en FabLab e introduciendo el modelo de impresora podemos comprobar si los diseños se ajustan.

Este proceso requirió de varios intentos debido al tamaño de los modelos. Tras realizar las modificaciones podemos comprobar que las partes se pueden imprimir como se puede ver en la Figura 63.

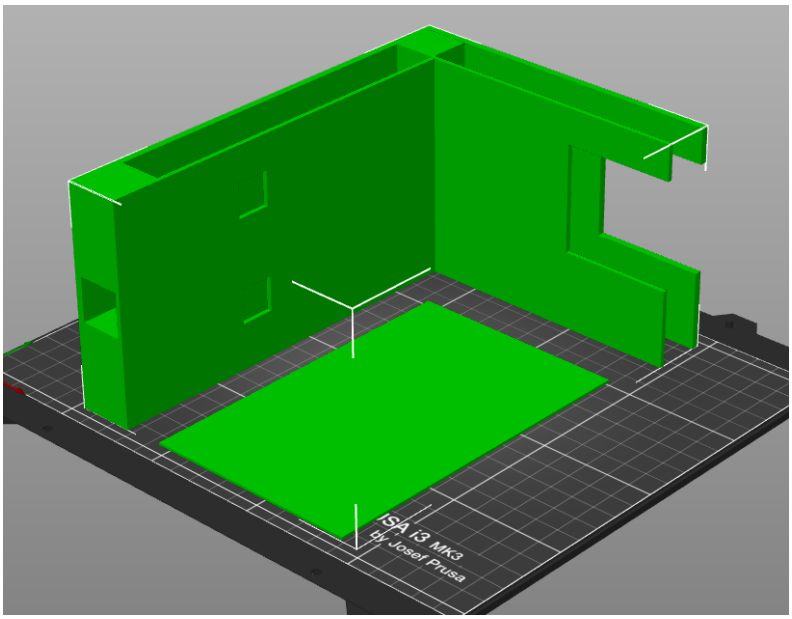


Figura 63 Detalle de impresión de la alfombra

El resultado final de la maqueta se puede apreciar en la Figura 64 y la Figura 65.

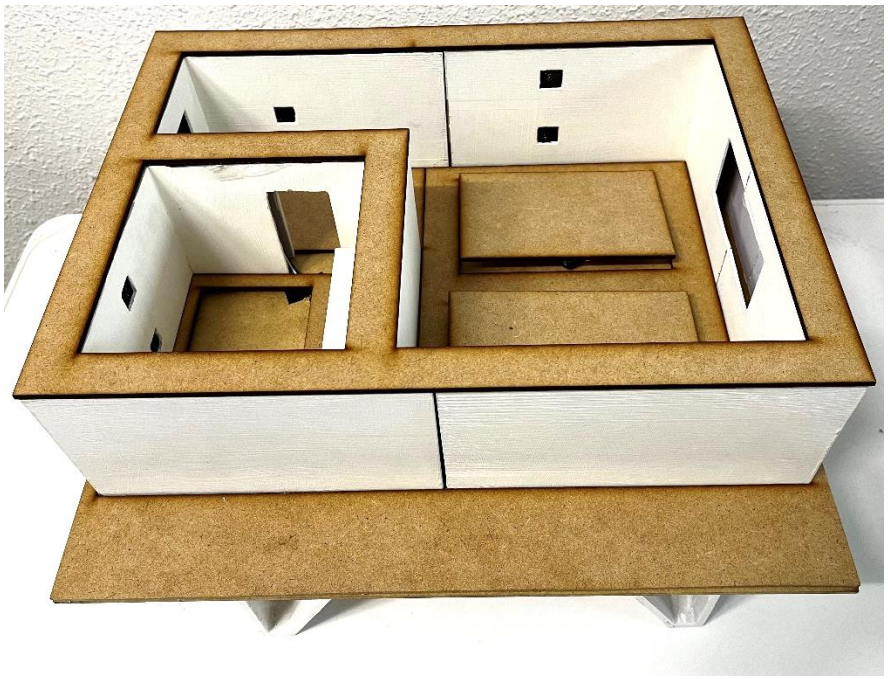


Figura 64 Vista lateral del resultado de la maqueta

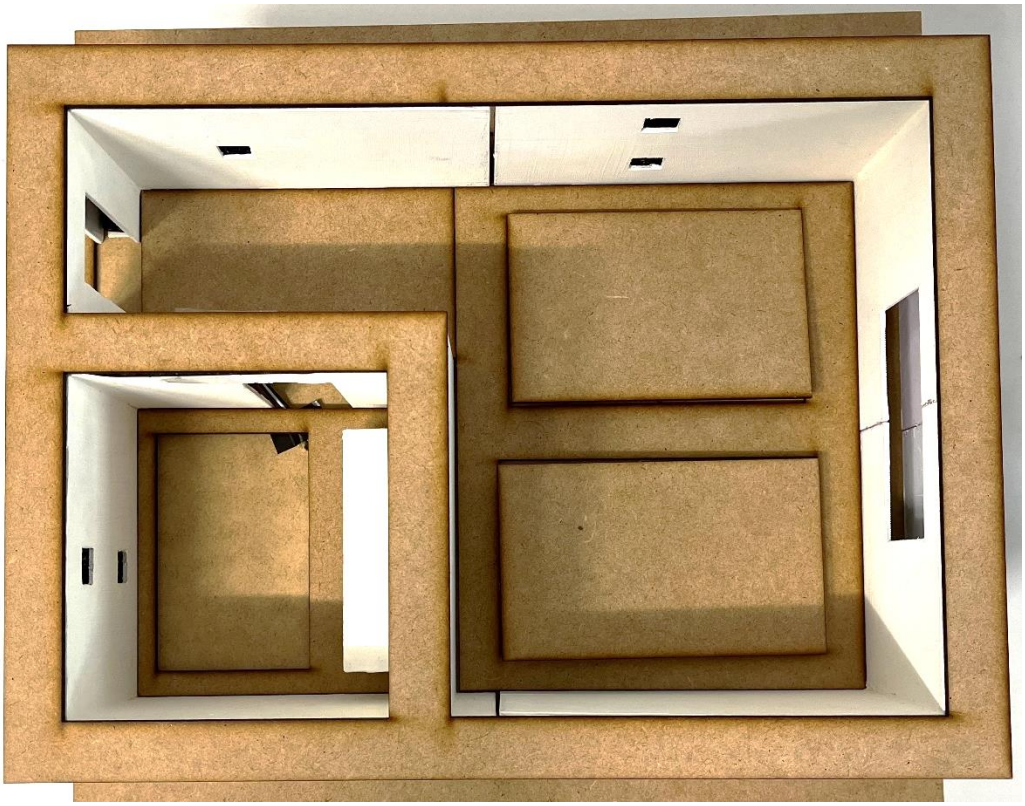


Figura 65 Vista aérea de la maqueta

En la Figura 66 se puede observar el laboratorio donde se ha realizado la maqueta.

Figura 66 Sala de FabLab donde se han desarrollado las maquetas

Diseño módulo inalámbrico

Para el desarrollo de esta maqueta se ha empleado un diseño de Thingiverse (Figura 67). Este diseño se trata de un archivo ".scad" lo que permite la modificación de las dimensiones mediante código.

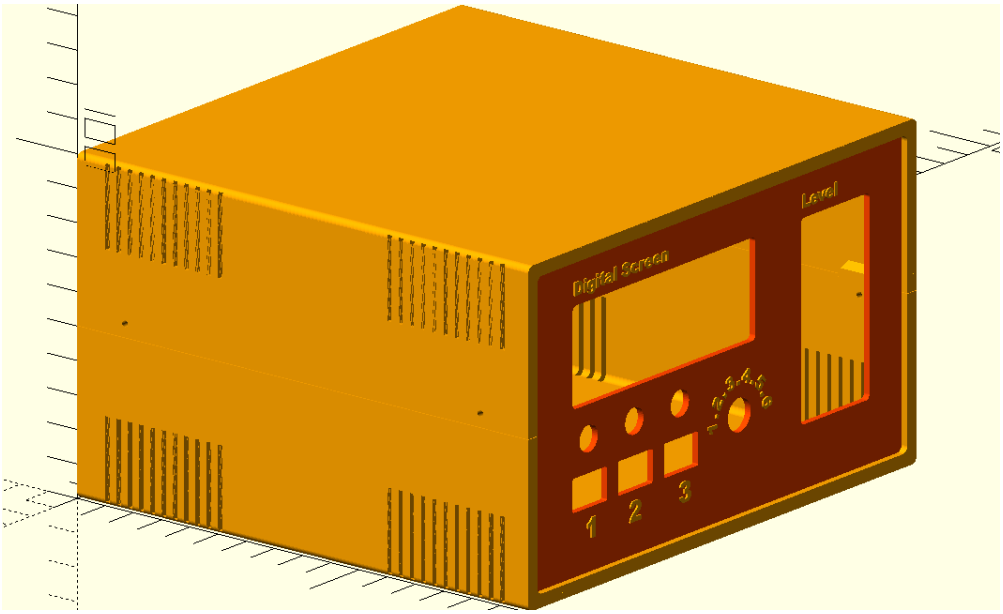


Figura 67 Diseño caja Thingiverse

Tras modificar las dimensiones y forma de la estructura se obtiene el modelo de la Figura 68.

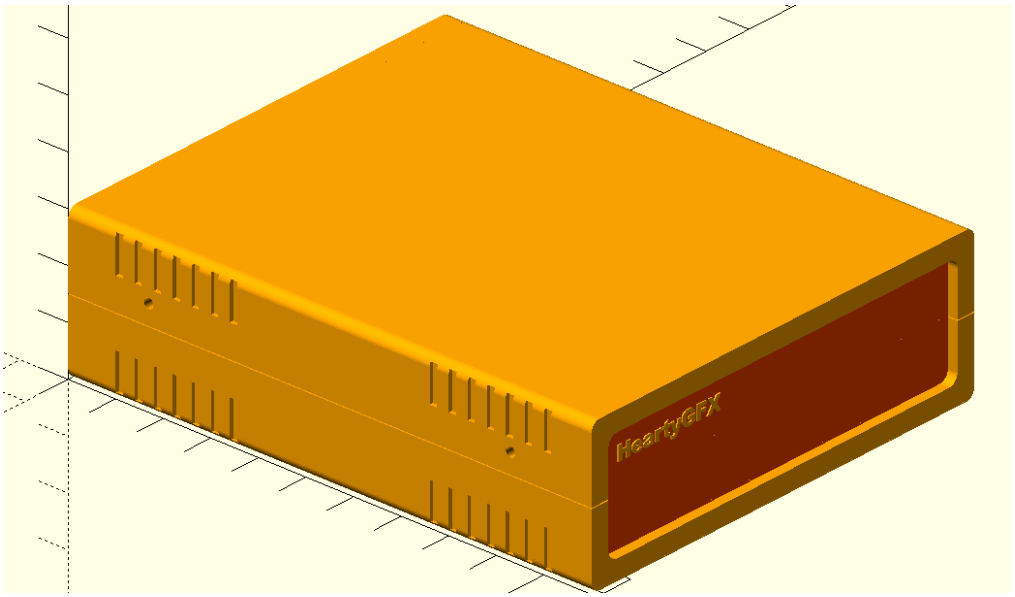


Figura 68 Diseño final del módulo inalámbrico

Tras imprimirlo se obtiene como resultado la maqueta de la Figura 69.

Figura 69 Detalle módulo inalámbrico

Diseño soporte

Al igual que como se realizó en el módulo inalámbrico se ha partido de un diseño de Thingiverse que puede verse en la Figura 70.

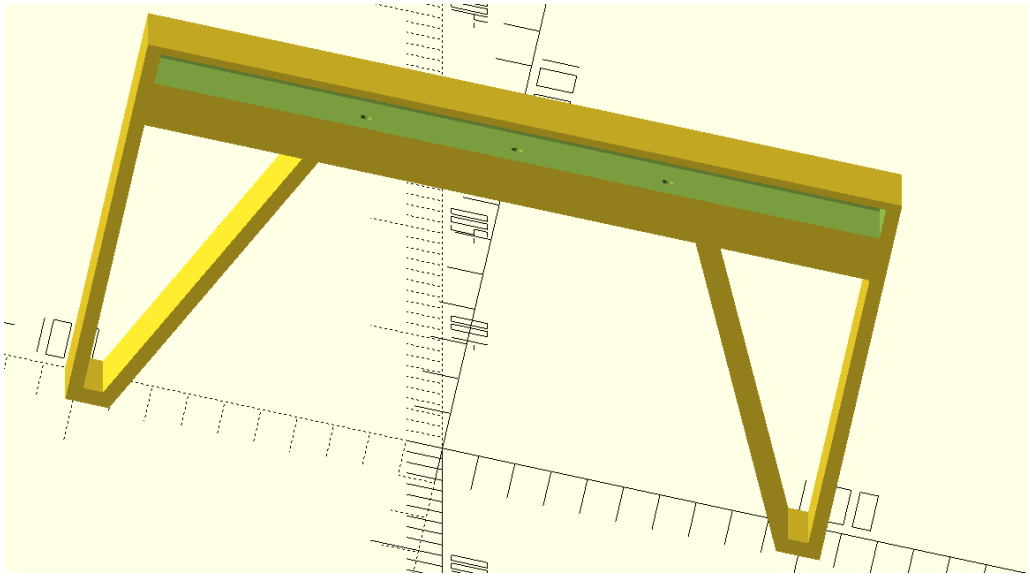


Figura 70 Diseño soporte

Posteriormente se realizaron las modificaciones necesarias para ajustarlas al proyecto. Obteniendo el modelo que se aprecia en la Figura 71:

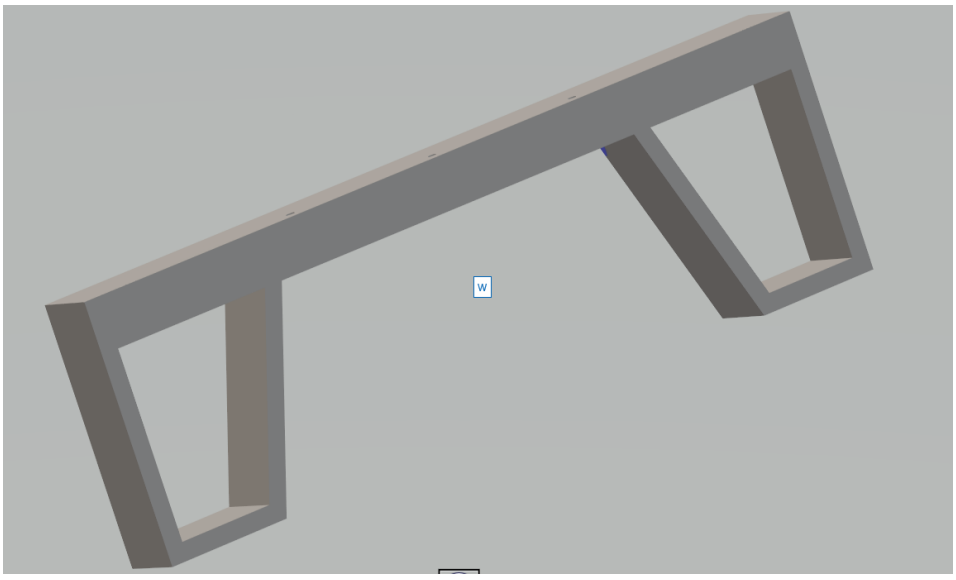


Figura 71 Diseño final del soporte

Y tras imprimirlo obtenemos el resultado que podemos ver en la Figura 72.

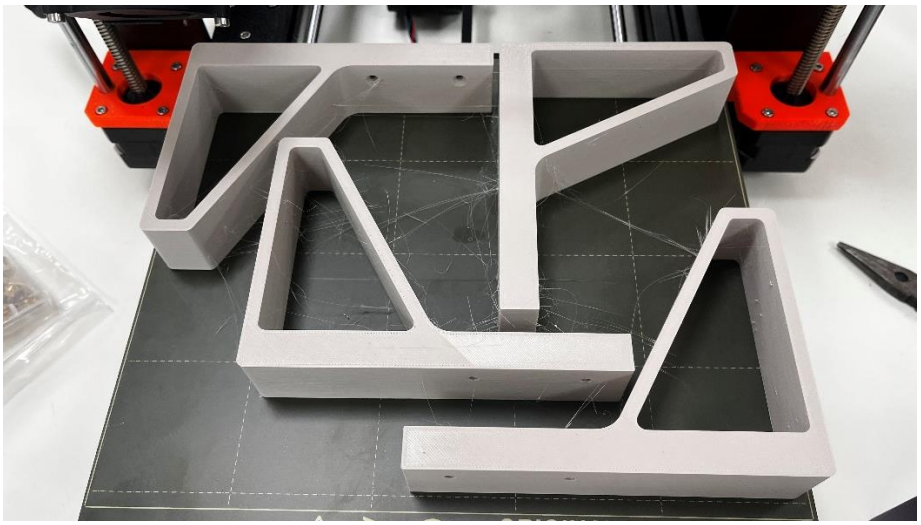


Figura 72 Impresión del soporte

6. Descripción funcional

En este apartado se va a mostrar el funcionamiento del proyecto. Para ello se simularán distintas situaciones. Usaremos el modelo 3D con el fin de demostrar gráficamente la situación, pero las situaciones se realizarán en la maqueta real. Se mostrarán las distintas pantallas de resultados en el panel de control de la habitación tras realizar las distintas acciones. Se irán enumerando las diferentes situaciones e indicando las condiciones indicadas para que se generen y el resultado que tendremos.

Situación 1: Usuario entra en la habitación

Para que se de esta situación el usuario se debe encontrar fuera de la estancia como se ve en la Figura 73.

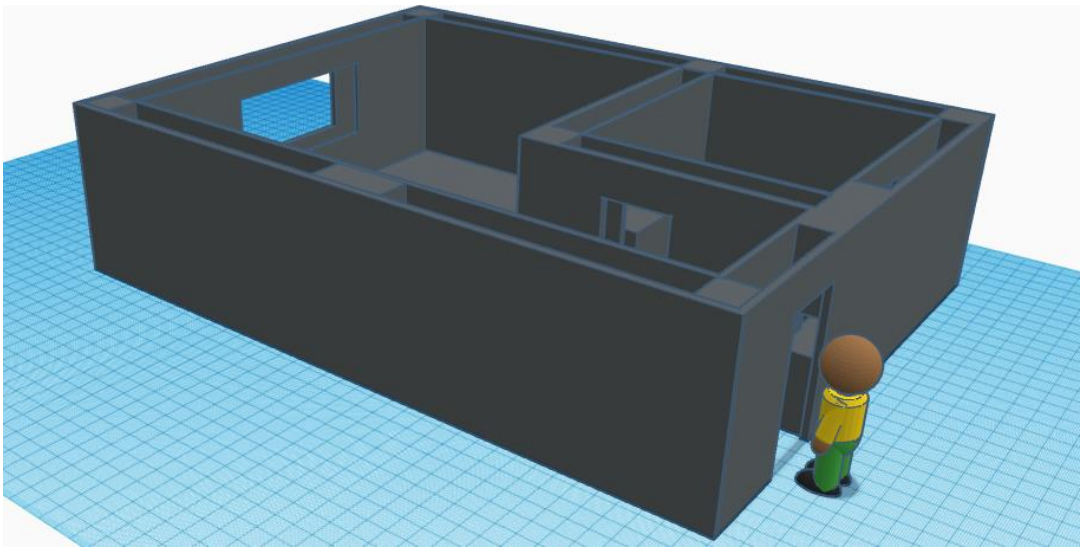


Figura 73 Situación persona fuera de la habitación

Pasando por la puerta activa el primer sensor IR, si sigue caminando activa el segundo sensor IR (Figura 74).

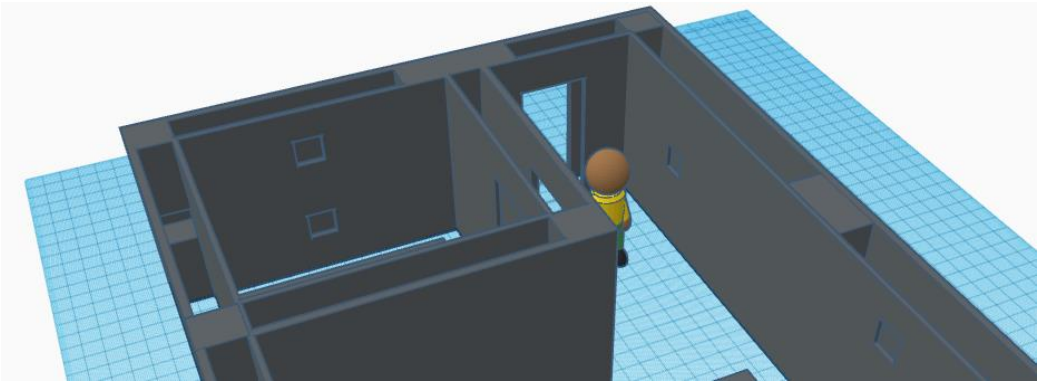


Figura 74 Situación persona entrando en la habitación

Esto desencadena que se envíe el evento “Usuario ha entrado en la habitación”. Con el siguiente resultado por pantalla (Figura 75).

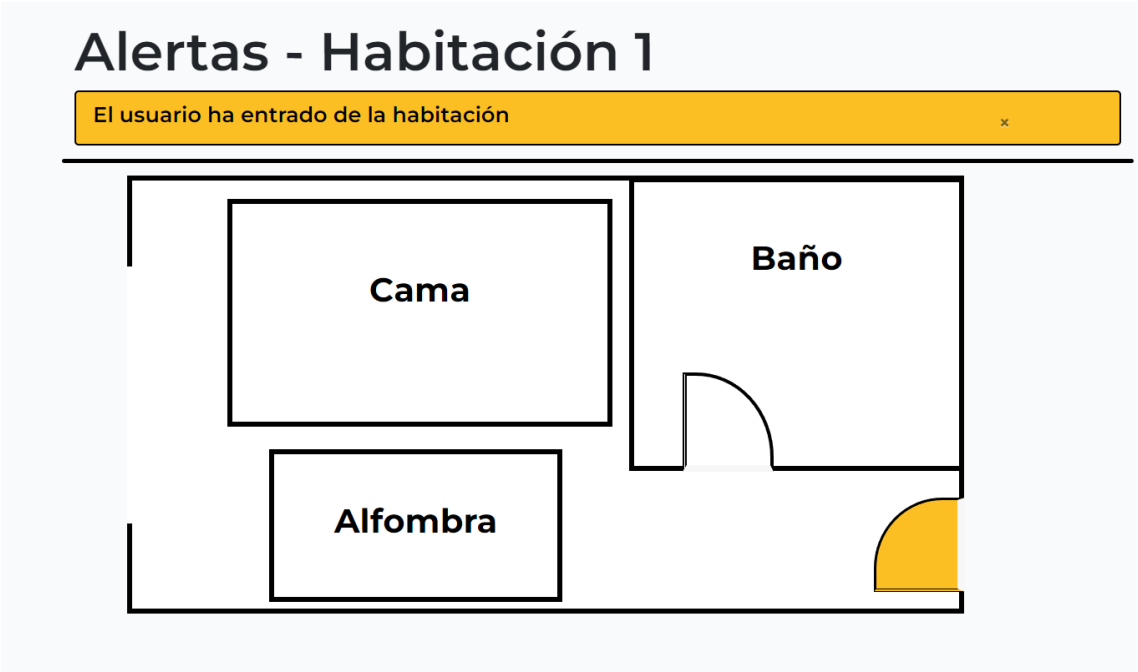


Figura 75 Resultado de la anterior acción

Situación 2: Usuario enciende la luz de la habitación

Para que se produzca este evento, la iluminación debe encontrarse apagada con anterioridad. El usuario, al encenderla (Figura 76), producirá con el consecuente mensaje (Figura 77).

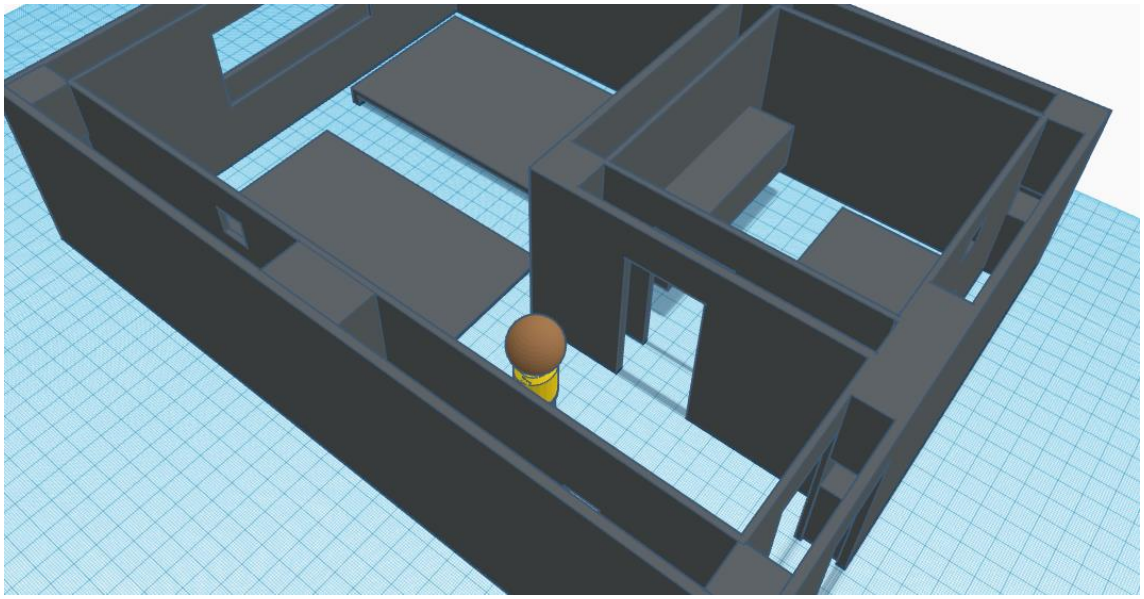


Figura 76 Usuario enciende la luz

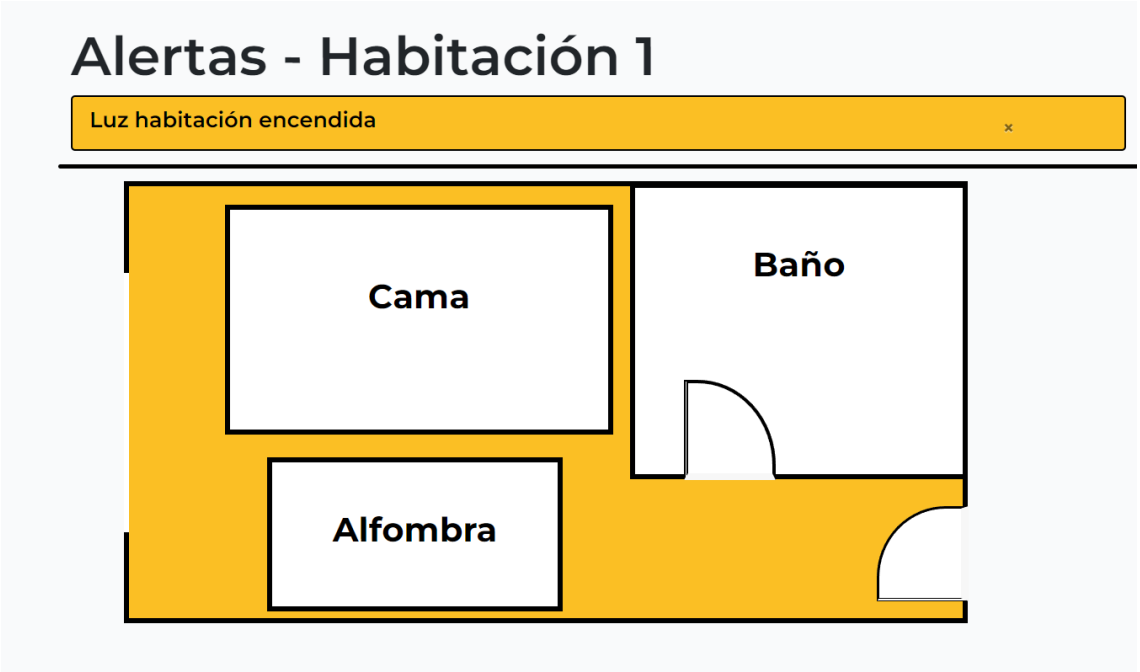


Figura 77 Resultado de encender la luz

Situación 3: Usuario apaga la luz de la habitación

Situación contraria al caso anterior, para que se produzca este evento la luz debe encontrarse encendida con anterioridad. Una vez que el usuario realiza esta acción se mostrará el siguiente resultado (Figura 78).

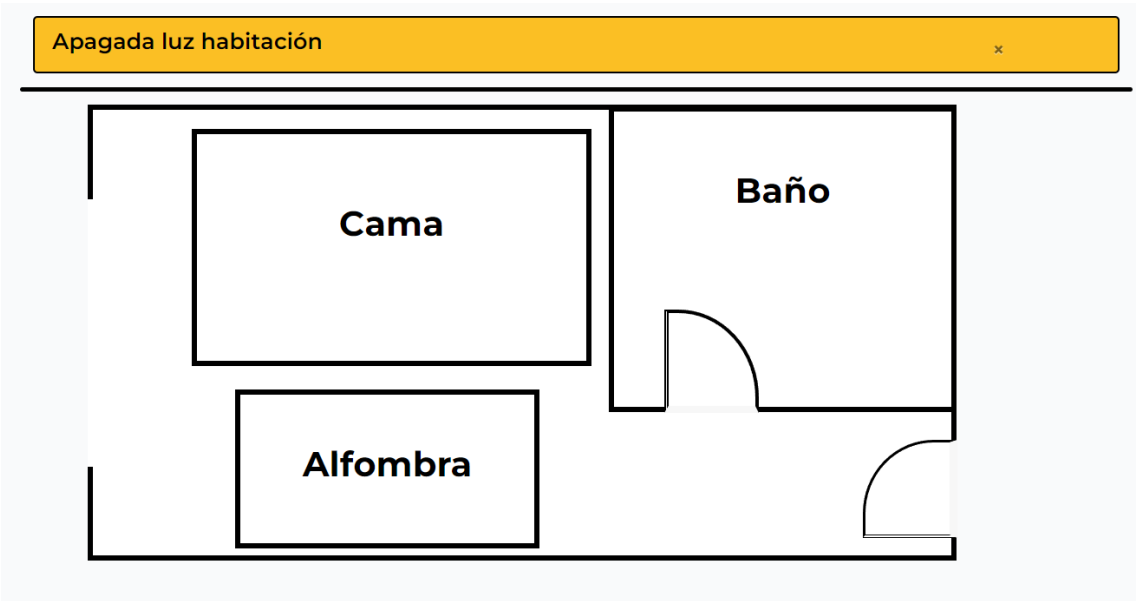


Figura 78 Resultado de apagar la luz

Situación 4: Usuario utiliza la cama

Una de las situaciones más comunes es la que se muestra a continuación, el usuario usa la cama (Figura 79 y Figura 80).

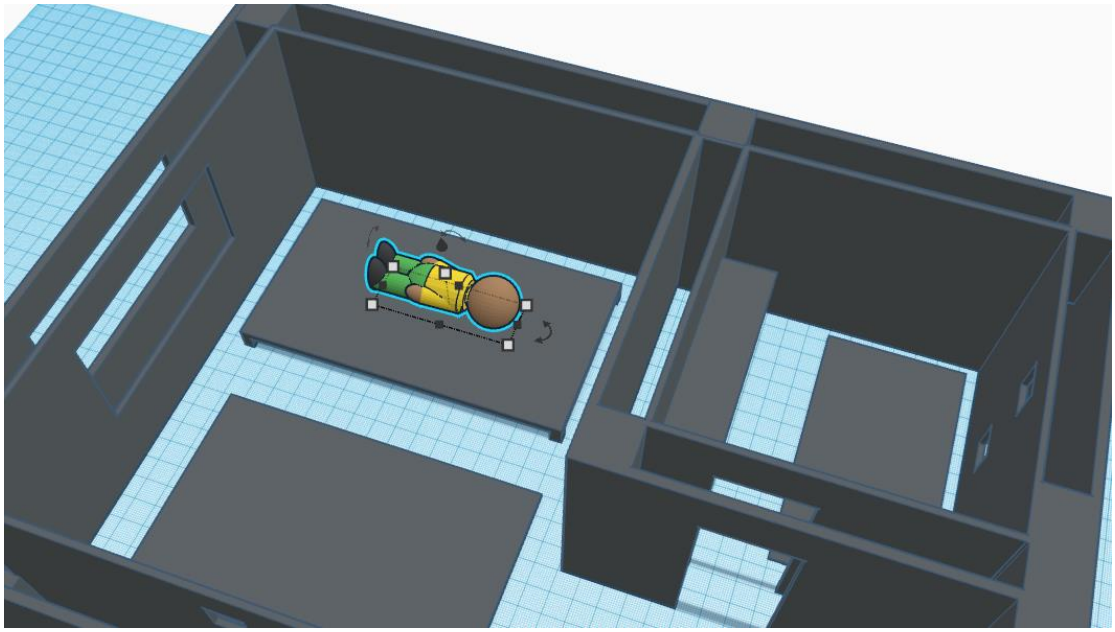


Figura 79 Usuario utilizando la cama

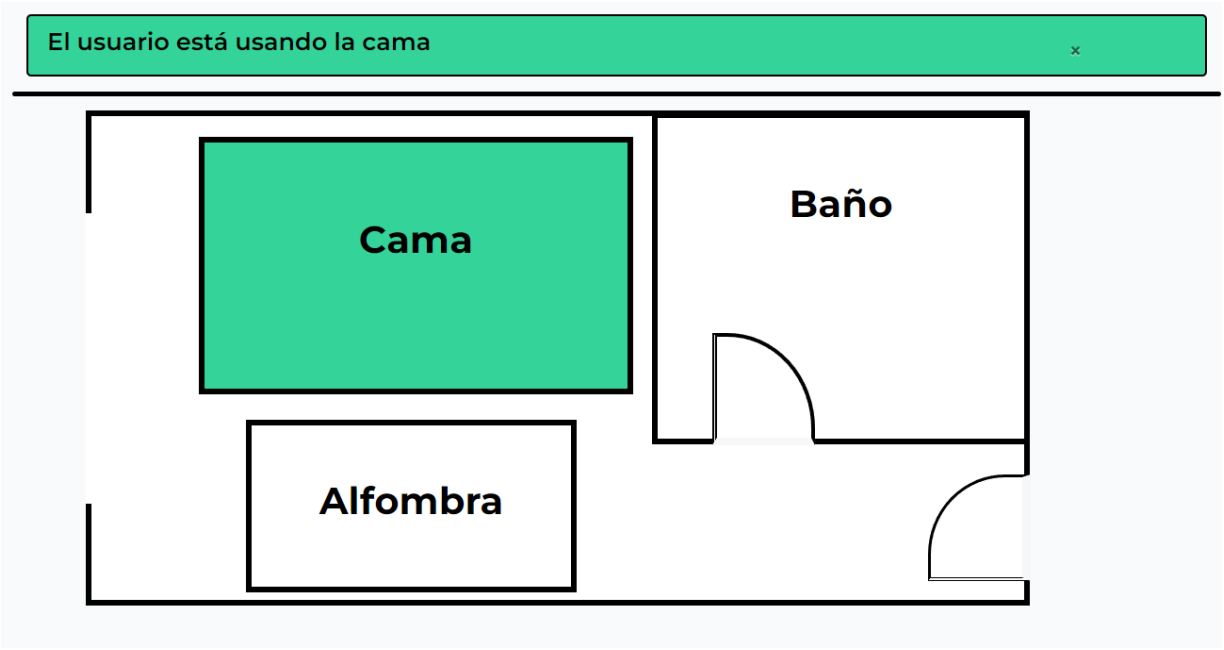


Figura 80 Resultado de usar la cama

Situación 5: Usuario deja de utilizar la cama

Situación contraria a la anterior. El usuario se levanta de la cama (Figura 81). La alerta (Figura 82) que se produzca puede cambiarse dependiendo del horario.

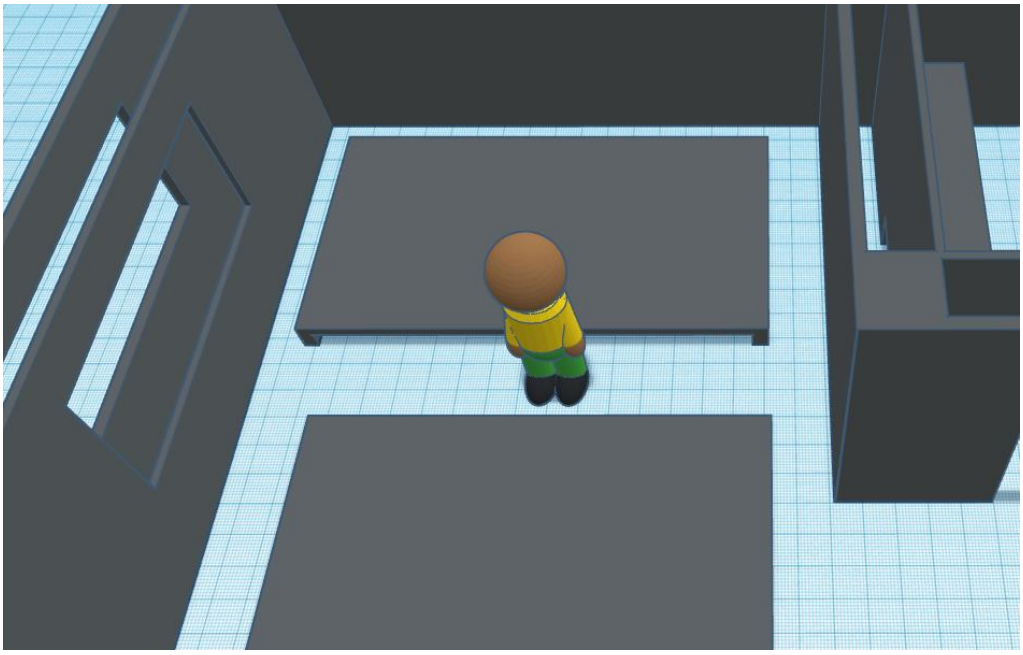


Figura 81 Usuario se levanta de la cama

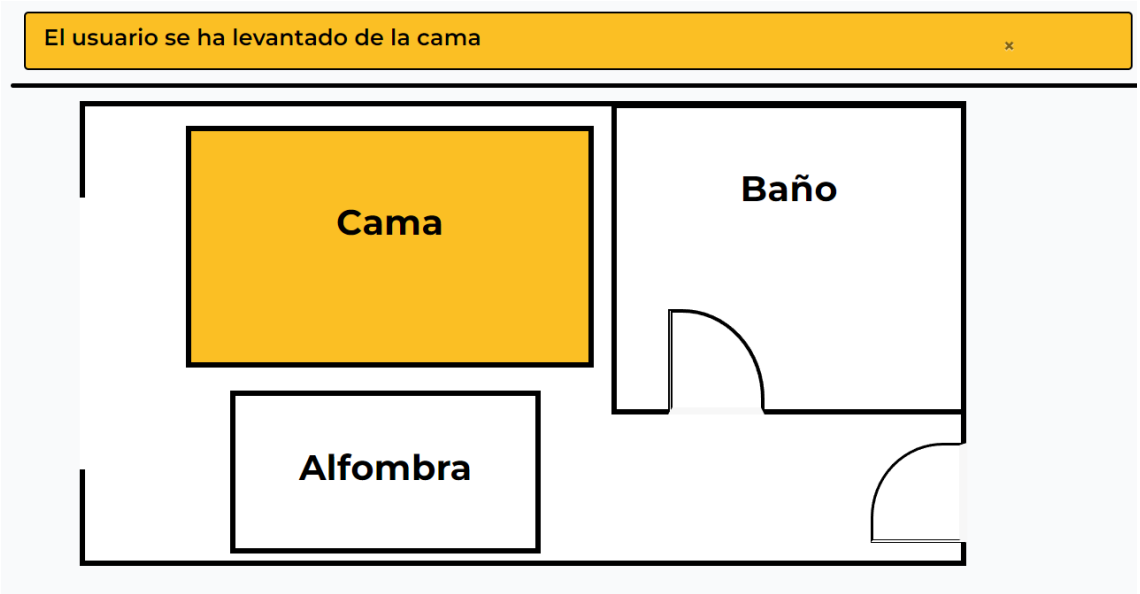


Figura 82 Resultado de dejar la cama

Situación 6: Usuario sale de la habitación

Para que se produzca esta situación, el usuario debe encontrarse dentro de la habitación (Figura 83) y haber accionado al pasar el sensor de presencia exterior (Figura 84).

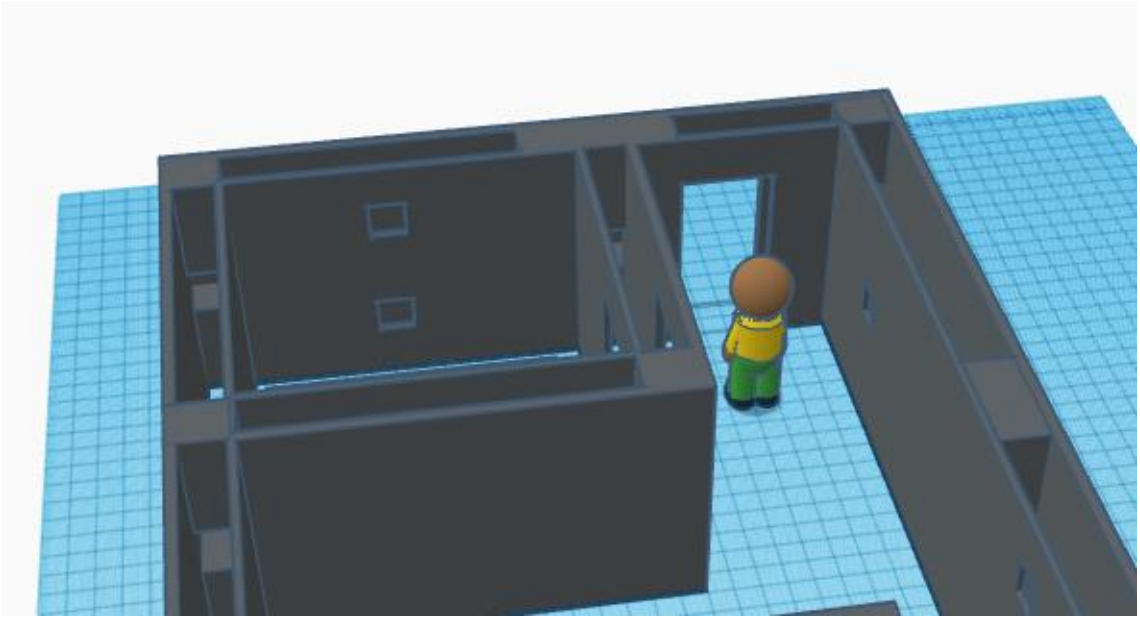


Figura 83 Accionamiento del sensor

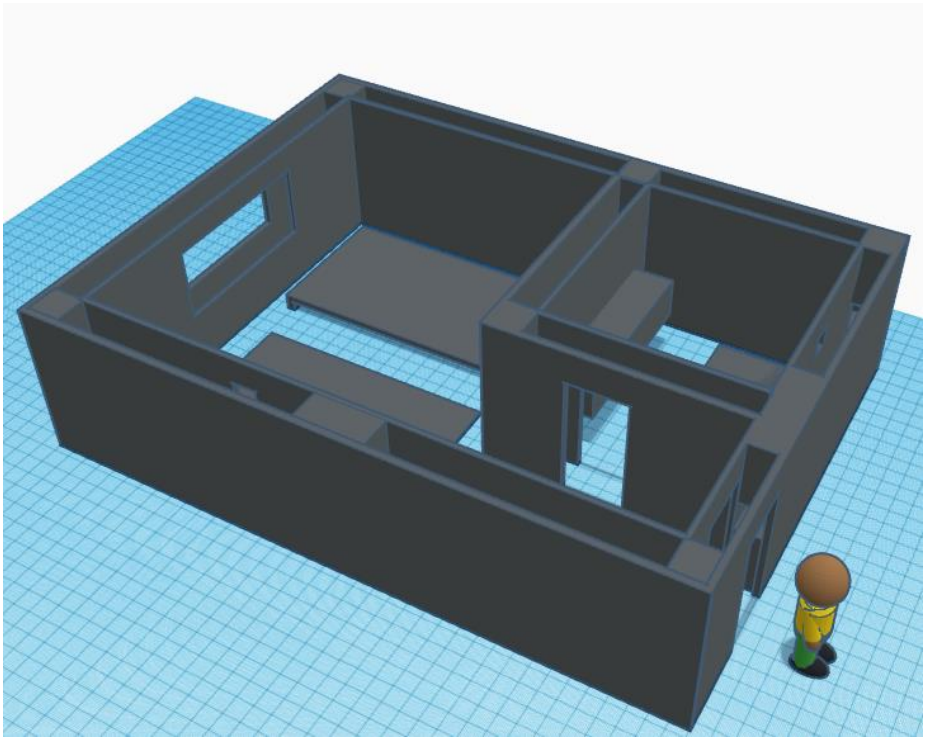


Figura 84 Salida de la habitación

Cabe destacar que la alerta (Figura 85) producida deberá ser atendida por el responsable y no desaparecerá hasta que se solucione.

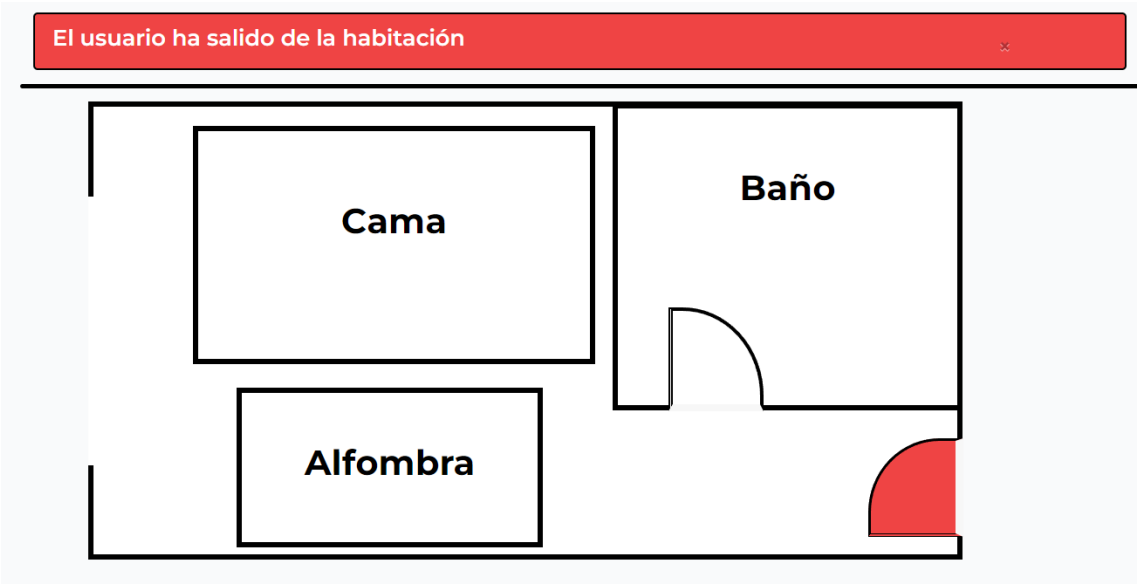


Figura 85 Resultado de salir de la habitación a hora no permitida

Cuando se ha solucionado la situación el panel queda de la siguiente forma (Figura 86).

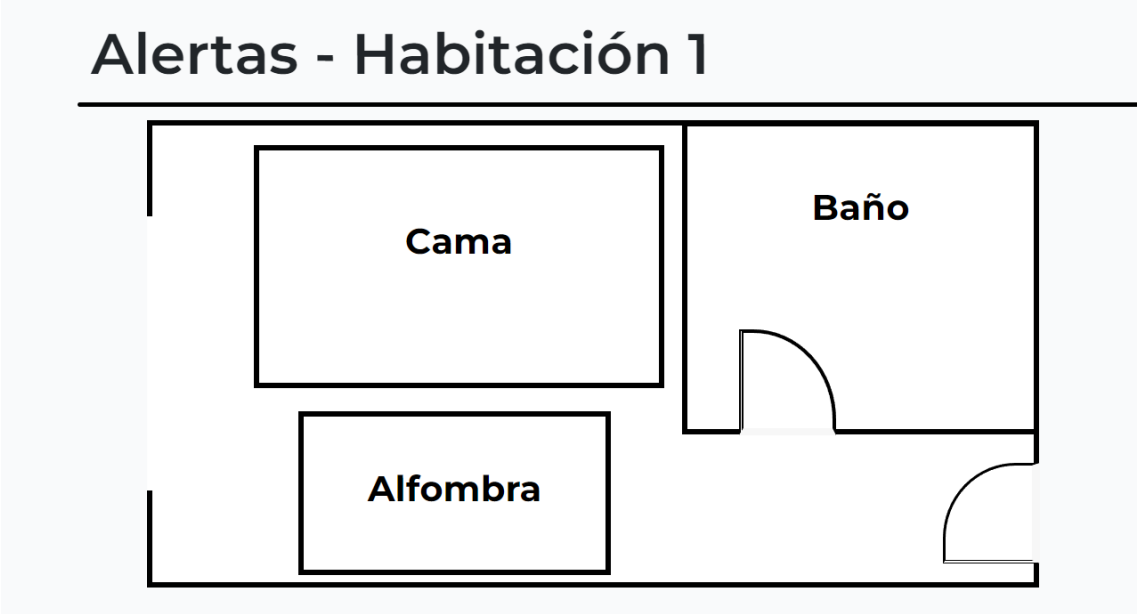


Figura 86 Resultado de contener la situación

Situación 7: Usuario entra en el baño

Para que se produzca esta situación el usuario debe encontrarse dentro de la estancia. El siguiente paso es accionar, en orden, tanto el sensor de la puerta como el del interior del baño. Este proceso se puede apreciar en la Figura 87 y Figura 88, con el consecuente resultado de la Figura 89.

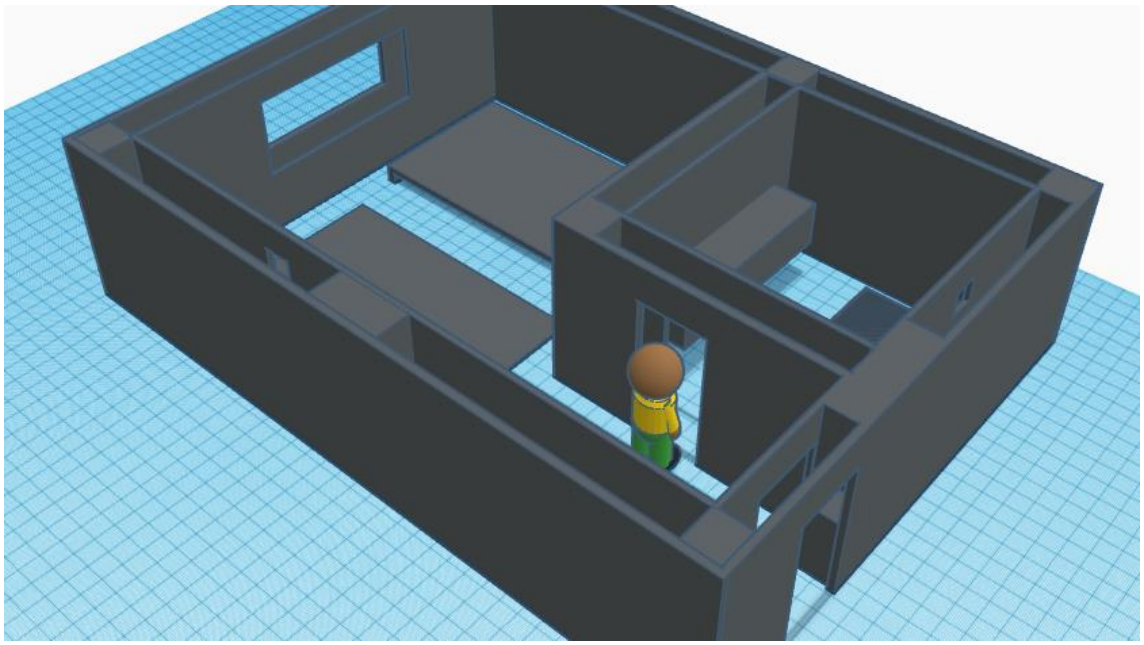


Figura 87 Usuario antes de entrar en el baño

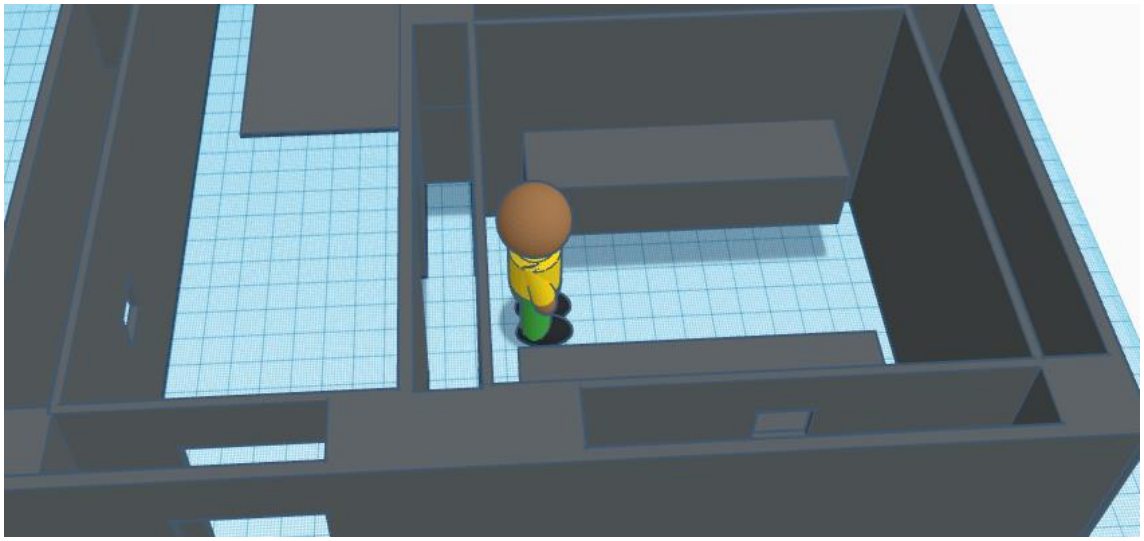


Figura 88 Usuario tras entrar en el baño

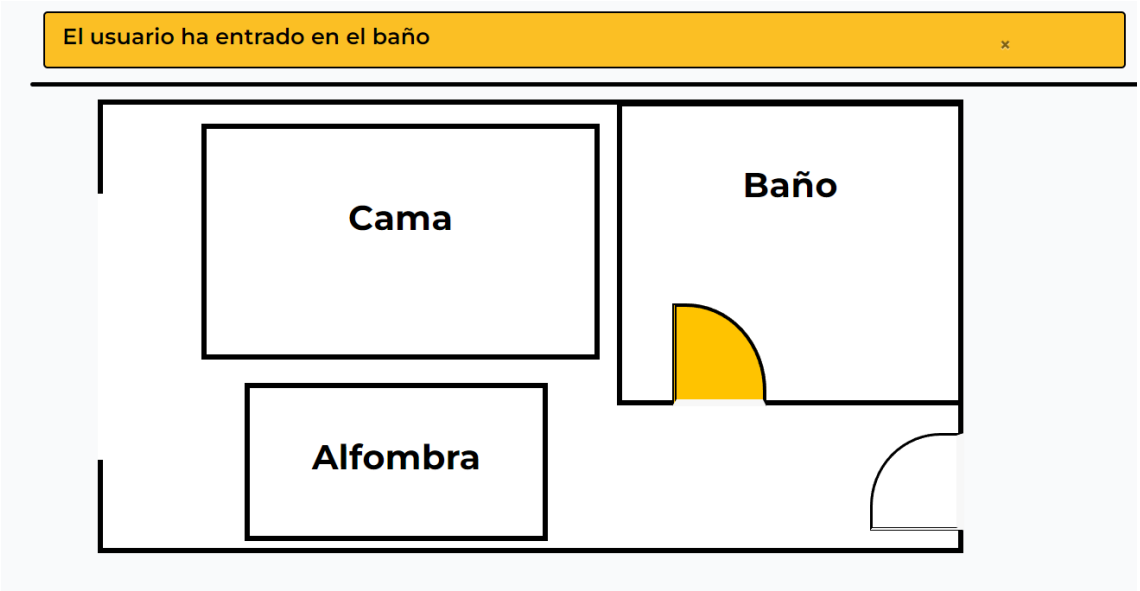


Figura 89 Resultado de la anterior acción

Situación 8: Usuario sale del baño

Estado contrario al anterior, por tanto, deben darse las condiciones contrarias. El usuario tiene que estar dentro del baño (Figura 90) y salir hacia la habitación (Figura 91). Durante el trayecto accionará primero el sensor interior y posteriormente el exterior, con el resultado acorde (Figura 92).

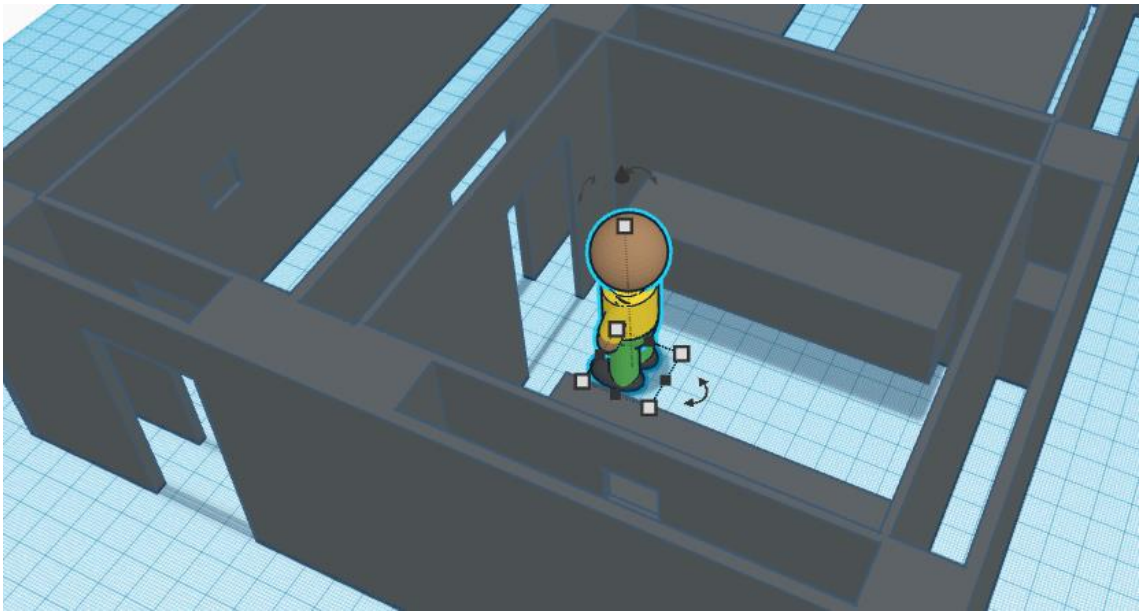


Figura 90 Usuario dentro del baño

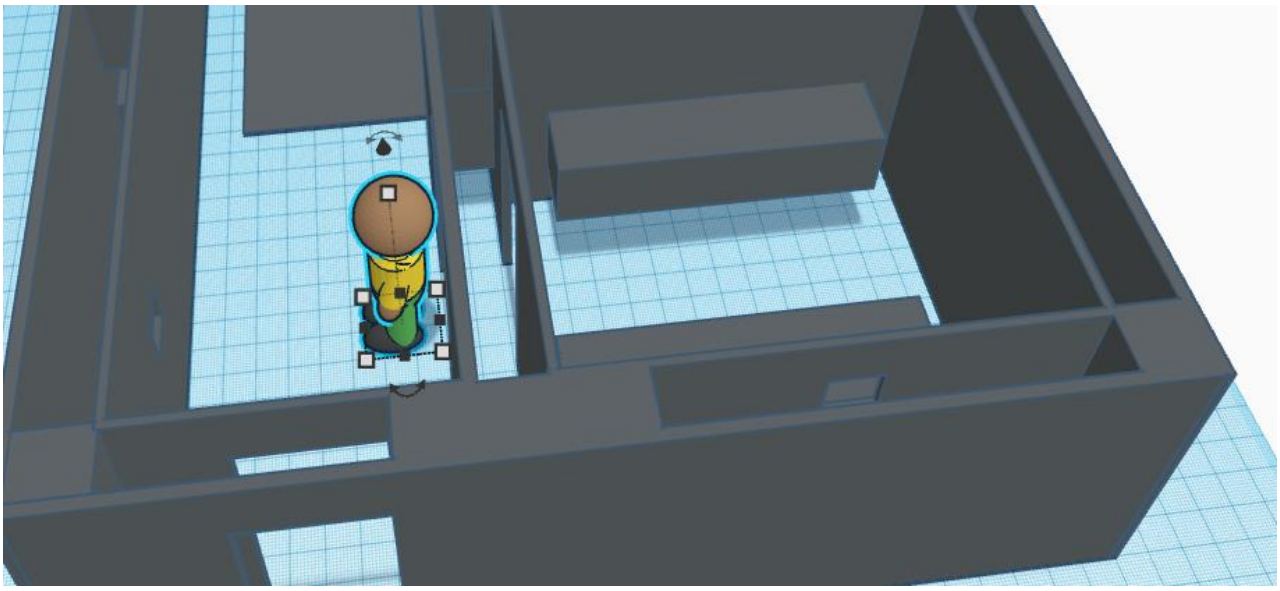


Figura 91 Usuario fuera del baño

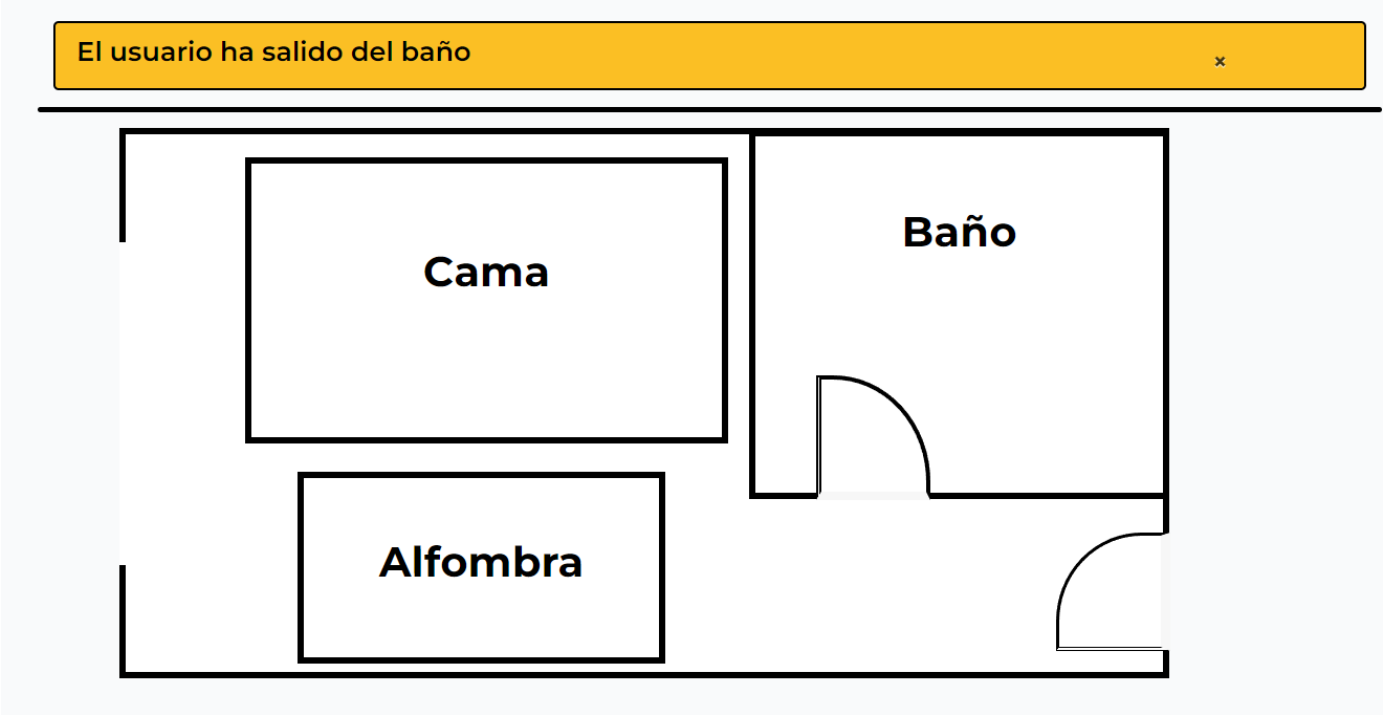


Figura 92 Resultado de salir del baño

Situación 9: Usuario sufre caída en el baño

En esta situación, el usuario no lleva encima el dispositivo inalámbrico. Ha sufrido una caída y es detectado tanto por el sensor de presión del suelo como por el de presencia inferior (Figura 93 y Figura 94).

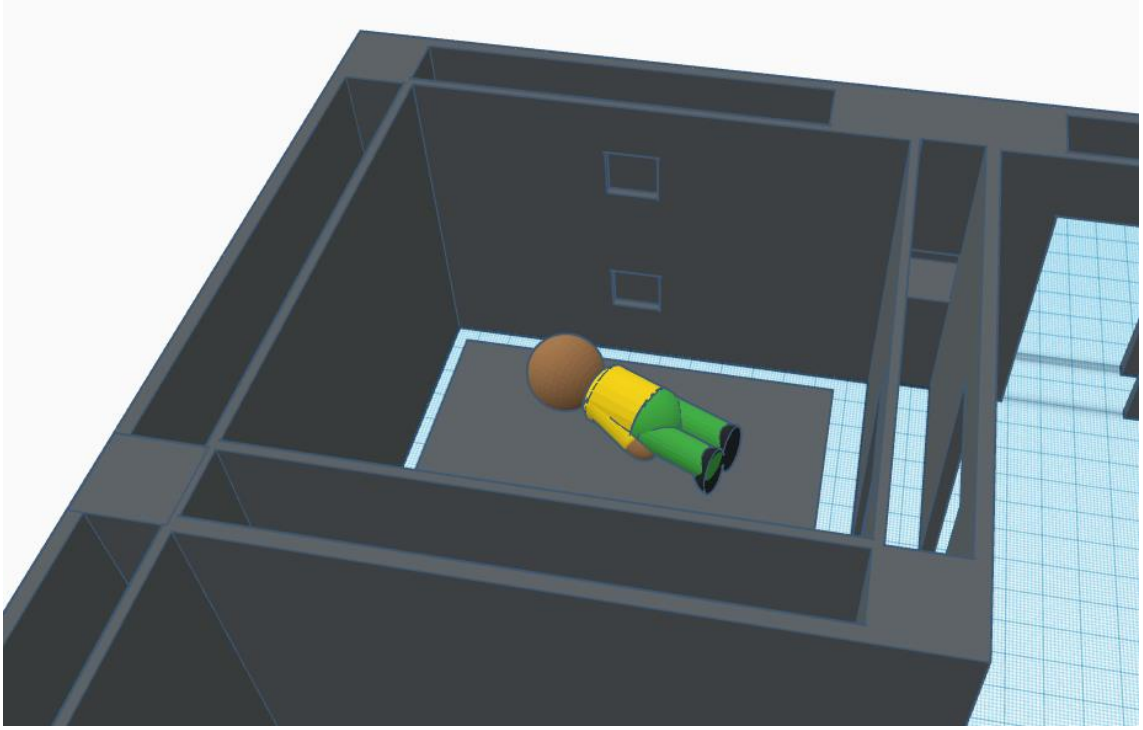


Figura 93 Caída en el baño

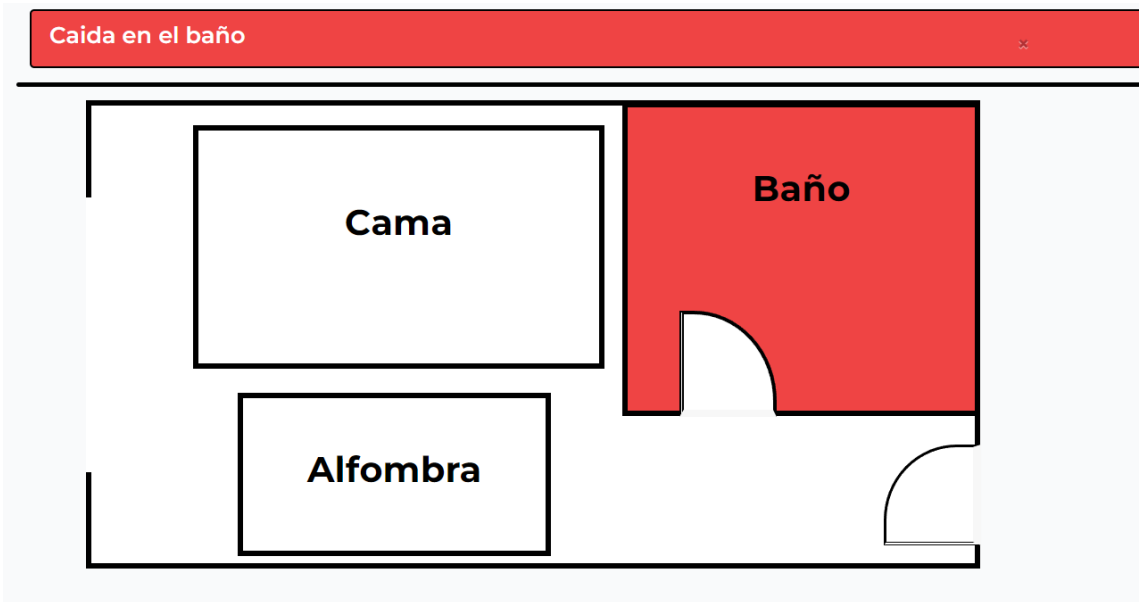


Figura 94 Resultado de caída en el baño

Situación 10: Usuario sufre caída en la alfombra

Situación similar al caso anterior, solo que producida en la habitación (Figura 95 y Figura 96).

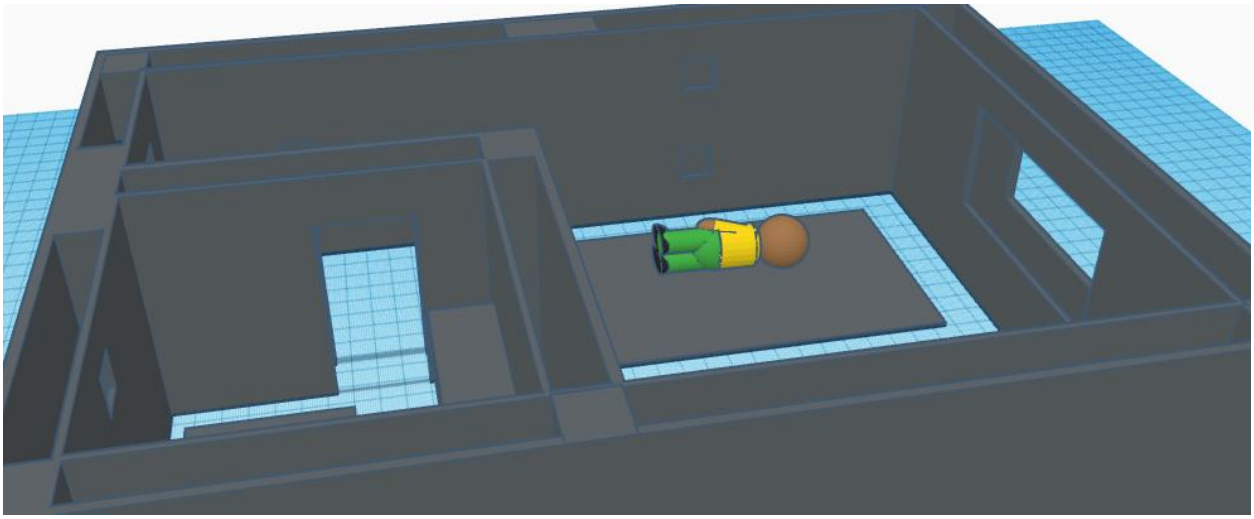


Figura 95 Caída encima de la alfombra

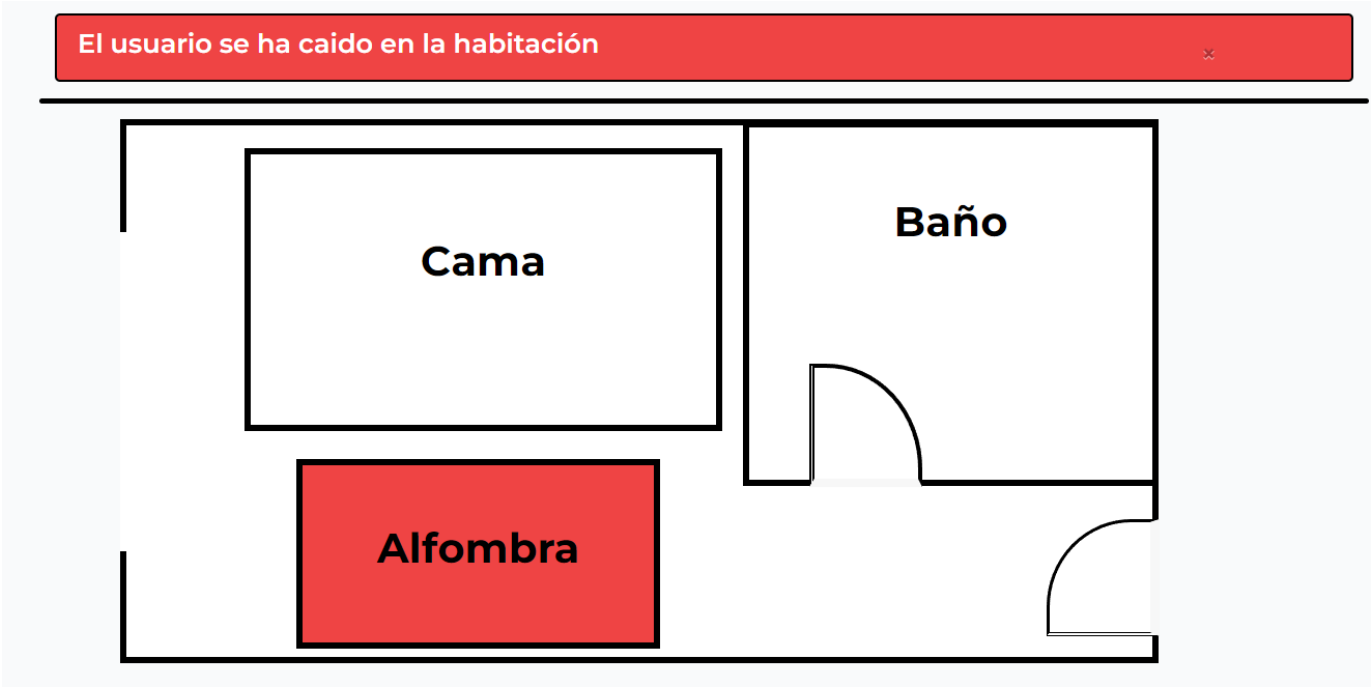


Figura 96 Consecuencia de la caída

Situación 11: Usuario sufre caída en la habitación o utiliza botón del pánico

En esta situación, a diferencia de los casos anteriores, no es detectada mediante sensores de presión o de presencia. Para detectar esta caída se utiliza el módulo inalámbrico (Figura 97). Se produce una situación similar si se pulsa el botón del pánico (Figura 98).

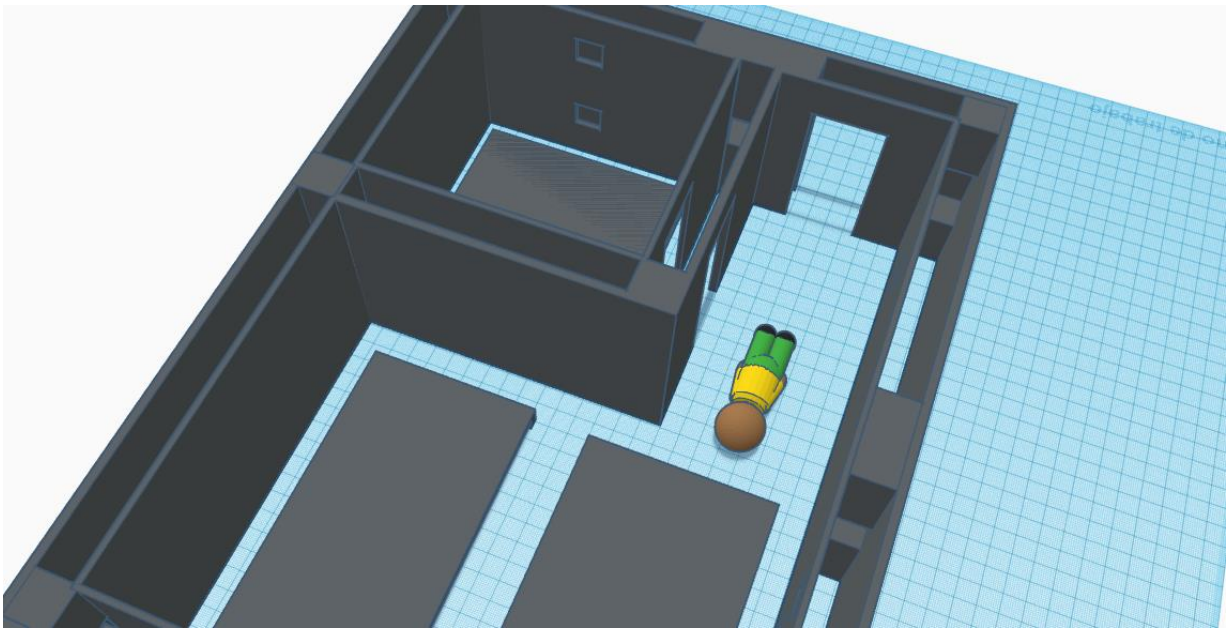


Figura 97 Caída utilizando el módulo inalámbrico

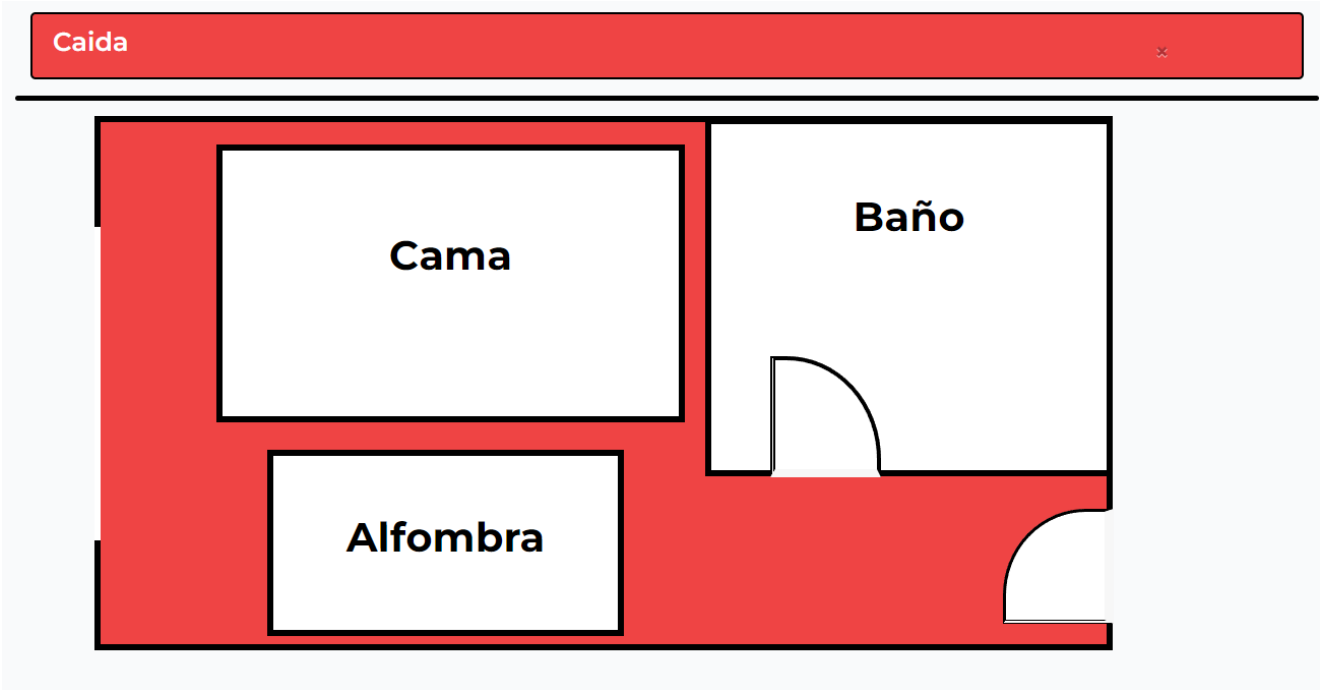


Figura 98 Resultado de la caída

7. Conclusiones

Tras llevar a cabo el desarrollo del trabajo a lo largo de más de un año desde el nacimiento de la idea se ha concluido el trabajo de forma satisfactoria. Además, se han analizado una serie de mejoras que pueden ser empleadas en el futuro.

Del desarrollo de proyecto cabe destacar que se ha adquirido nuevos conocimientos, ya sea sobre el funcionamiento de los sensores, de las tarjetas Arduino y de la plataforma de Google Cloud, y se ha conseguido llevar a cabo un proyecto software desde el inicio hasta su culminación. Este trabajo comprende tanto el diseño y desarrollo de hardware, circuitos y maquetas, como el de software, con la creación de una plataforma web y uso de distintos servicios.

Repasaremos ahora brevemente los objetivos técnicos que se fijaron al inicio del proyecto y valoraremos su grado de realización:

- Ayuda: El objetivo primario del proyecto era ayudar tanto a los usuarios como a los responsables. Este objetivo ha sido satisfecho, ya que tanto la plataforma como los sensores permiten controlar las situaciones peligrosas presentando una herramienta útil.
- No sustitución de personal: Como se indicó al inicio del desarrollo este proyecto no planeaba substituir personal, únicamente ser una herramienta de apoyo. Esto se ha conseguido ya que sin la presencia de una persona que controle la pantalla el sistema no funciona.
- Coste reducido: Uno de los pilares del desarrollo se ha alcanzado gracias al empleo de sensores de reducido precio, así como la utilización de Google Cloud. Esto hace que para instalaciones individuales el único coste sea el necesario para la instalación y compra de dispositivos. El coste por el uso de la plataforma es nulo o casi nulo. En cuanto a grandes instalaciones, el coste de compra se amortiza rápidamente, y su mantenimiento representa un gasto minúsculo.
- Velocidad de respuesta: Como se ha probado en las pruebas realizadas durante el desarrollo del proyecto, el envío de información en instantáneo. En cuanto a la recepción de los datos está en el rango del medio segundo a segundo desde que se envía. Por lo que podemos concluir que también se ha conseguido.
- Integridad y gestión de los datos: Dado que se emplea FireStore, los datos están seguramente almacenados y son accesibles desde cualquier lugar con conexión a internet.
- Escalabilidad: Como se ha comentado con anterioridad Google Cloud permite una escalabilidad casi infinita de dispositivos, estancias e instalaciones.

- Facilidad de uso: Se ha desarrollado una plataforma que permite conocer el estado de las estancias con un rápido vistazo y que muestra la información clara y prácticamente al instante. Por lo que podemos confirmar que este objetivo también ha sido cumplido.
- Robustez ante distintas situaciones: Como último objetivo que cumplimentar nos queda el de robustez y reporte adecuado. Este objetivo se ha conseguido, pero puede ser mejorado con el uso de más y mejores sensores.

Tras finalizar el proyecto se llega a la conclusión de que los objetivos propuestos, tanto funcionales como personales, se han cumplido. En cuanto a los personales el alumno ha aprendido a emplear nuevas tecnologías, herramientas y programas para realizar el proyecto.

A parte de los conocimientos adquiridos y del cumplimiento de una serie de objetivos este trabajo ha permitido al autor explorar ramas de la informática que hasta entonces le eran desconocidas. Cabe destacar es el campo del trabajo descentralizado en la nube o el uso de impresoras 3D o cortadoras láser. Este último aspecto se debe gracias a la colaboración con FabLab, que ha permitido que el proyecto sea fácilmente entendible y demostrable como prueba de concepto, haciendo uso de una representación en 3D de una estancia.

Como nota final hay que destacar que este proyecto ha sido una experiencia muy enriquecedora, que ha permitido al autor conocer cómo se desarrolla un sistema real, desde la idea inicial hasta la finalización, pasando por todas las fases del desarrollo.

El proyecto se planteó inicialmente hace más de un año. Durante este periodo se ha realizado diversas tareas que han supuesto un desafío para el alumno, que le han permitido conocer y mejorar sus cualidades, así como aprender de distintos campos e investigar hasta obtener el resultado deseado.

8. Líneas de trabajo futuras

Una vez finalizado el proyecto final de carrera se plantean una serie de mejoras que se pueden seguir en un futuro para mejorar el proyecto existente:

- Emplear equipamiento más preciso y adecuado en las instalaciones a los utilizados en el desarrollo de la maqueta. Cómo se ha comentado en apartados anteriores, tanto durante el desarrollo del proyecto como en el de la maqueta se han empleado sensores pensados para este caso de uso en concreto. Queda claro entonces que para una posible instalación en el futuro es necesario emplear mejores tarjetas (que permitan un mayor número de conexiones) y sensores más precisos.
- Realizar una aplicación móvil. Para poder llevar el control de las estancias también desde dispositivos móviles. Pese a que la instalación actual presenta una aplicación web desde la cuál tener una vista global del sistema, una de las mejoras que se pueden plantear es realizar una aplicación para dispositivos móviles nativa.
- Mejora del panel de control. Se plantea desarrollar una serie de mejoras sobre el panel de control actual. Entre estas mejoras se proyectan realizar diseños de distintas estancias, tanto en complejidad como en estructura. Incluso se podría realizar en tres dimensiones, para poder obtener una mejor lectura de los datos y también conocer mejor cómo y dónde se encuentran instalados los sensores.
- Crear un sistema que sustituya a Google Cloud. Como se ha indicado con anterioridad, el sistema que permite que el proyecto funcione como lo hace es la nube de Google. Pese a que tiene una gran cantidad de buenas cualidades (escalabilidad, precio reducido, rapidez) presentan una serie de desventajas. Entre estas encontramos principalmente que es un servicio propietario de una compañía, y se depende de ella para su viabilidad a largo plazo y funcionamiento del sistema. Así como la potencial vulnerabilidad de la privacidad, ya que los mensajes se envían a los servidores del gigante tecnológico y son procesados. Esto hace que, aunque los mensajes no contengan información comprometida o importante, el sistema no sea completamente opaco. Por ello se plantea desarrollar un sistema que realice las mismas funciones, pero sin tener que depender de una tercera empresa. Queda claro que este apartado es el más difícil de llevar a cabo, debido principalmente al coste y complejidad que conlleva desarrollar un sistema de este tipo. Este problema, unido a lo conveniente que resulta el sistema de Google, hace que sea extremadamente complejo migrar a un sistema personalizado.

- Mejorar el control de las personas que entran en la habitación. Como se comentó en capítulos anteriores se plantea mejorar el control de la habitación mediante el uso de tarjetas RFID y así distinguir a la persona que entran en la habitación. Con ello se consigue evitar problemas de duplicidad de mensajes.
- Adaptación del sistema para otros requerimientos. Otro de las potenciales mejoras es adaptar el sistema para controlar otras estancias y realizarlo en otros campos, diferentes a residencias e instalaciones individuales, como pueden ser hospitales o instalaciones penitenciarias.
- Documentación y publicación del proyecto. Se plantea realizar una documentación exhaustiva del apartado de conexión y comunicación de dispositivos Arduino y Google Cloud. Pese a la documentación existente para otras tarjetas y dispositivos, es escasa e incompleta para los elegidos en el proyecto. Por ello se buscará documentarlo adecuadamente para posteriores usos. También se documentará el apartado de conexión entre Cloud Functions y FireStore. Por último, se planea hacer público el proyecto y las conclusiones obtenidas del mismo para que pueda servir como punto de partida para otros trabajos y sistemas.
- También se plantea para el módulo inalámbrico la creación de una tarjeta que contenga toda la funcionalidad necesaria, así se reduciría el tamaño y aumentaría la eficiencia. Para ello habría que desarrollar y producir la tarjeta de circuito integrado.

Glosario

Google Cloud IoT Core

Cloud IoT Core es un servicio de IoT que extrae, conecta y gestiona datos de diferentes dispositivos que se integran en un único sistema que puede integrarse con Google Analytics. El sistema IoT Google es compatible con protocolos industriales como MQTT y HTTP, por lo que los dispositivos pueden utilizarse sin necesidad de programación ni grandes cambios.

Arduino

Arduino es una compañía de desarrollo de software y hardware libres, así como una comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales e interactivos que puedan detectar y controlar objetos del mundo real.

WIFI

El WIFI es una tecnología que permite la interconexión inalámbrica de dispositivos electrónicos. Los dispositivos habilitados con WIFI pueden conectarse entre sí o a Internet a través de un punto de acceso de red inalámbrica.

HTML

HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web.

CSS

Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML. CSS describe como debe ser mostrado el elemento en la pantalla, en papel, en el habla o en otros medios.

JavaScript

JavaScript (JS) es un lenguaje de programación ligero, interpretado o compilado justo-a-tiempo con funciones de primera clase. Es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa.

FireStore

Cloud Firestore es una base de datos NOSQL flexible y escalable para el desarrollo en servidores, dispositivos móviles y la Web desde Firebase y Google Cloud. Mantiene los datos sincronizados entre apps cliente a través de objetos de escucha en tiempo real y ofrece soporte sin conexión para dispositivos móviles y la Web.

IoT

Internet de las cosas (IoT) describe la red de objetos físicos ("cosas") que llevan incorporados sensores, software y otras tecnologías con el fin de conectarse e intercambiar datos con otros dispositivos y sistemas a través de Internet.

Google Cloud Platform

Es un conjunto de servicios de computación en la nube que se ejecuta en la misma infraestructura que Google utiliza internamente para sus productos de usuario final, como Google Search, Gmail, Google Drive y YouTube. Junto con un conjunto de herramientas de gestión, proporciona una serie de servicios modulares en la nube que incluyen computación, almacenamiento de datos, análisis de datos y aprendizaje automático.

Microsoft Azure

Microsoft Azure es un servicio de computación en la nube operado por Microsoft para la gestión de aplicaciones a través de centros de datos gestionados por Microsoft. Ofrece software como servicio (SaaS), plataforma como servicio (PaaS) e infraestructura como servicio (IaaS) y es compatible con muchos lenguajes de programación, herramientas y marcos diferentes, incluyendo software y sistemas específicos de Microsoft y de terceros.

Amazon Web Services

Es una filial de Amazon que proporciona plataformas de computación en la nube y APIs bajo demanda a particulares, empresas y gobiernos, en régimen de pago por uso. Estos servicios web de computación en la nube proporcionan capacidad de procesamiento informático distribuido y herramientas de software a través de las granjas de servidores de AWS.

MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de red ligero, de publicación-suscripción, de máquina a máquina. Está diseñado para conexiones con ubicaciones remotas que tienen dispositivos con restricciones de recursos o un ancho de banda de red limitado. Debe ejecutarse

sobre un protocolo de transporte que proporcione conexiones ordenadas, sin pérdidas y bidireccionales, normalmente TCP/IP.

Push

Es un estilo de comunicación basado en Internet en el que la solicitud de una determinada operación es iniciada por el editor o servidor central. Los servicios push suelen basarse en las preferencias de información expresadas de antemano. Se trata de un modelo de publicación/suscripción.

Cloud Functions

Las funciones en la nube eliminan el trabajo de gestión de los servidores, la configuración del software, la actualización de los marcos de trabajo y la aplicación de parches en los sistemas operativos. El software y la infraestructura están totalmente gestionados por Google, de modo que sólo se tiene que añadir código.

Firebase

Es una plataforma ubicada en la nube, integrada con Google Cloud Platform, que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad, haciendo posible el crecimiento del número de usuarios y dando resultado también a la obtención de una mayor monetización.

BigQuery

Almacén de datos totalmente gestionado y sin servidor que permite el análisis escalable de petabytes de datos. Se trata de una plataforma como servicio (PaaS) que admite consultas mediante ANSI SQL. También cuenta con capacidades de aprendizaje automático integradas.

NoSQL

Una base de datos NoSQL (no relacional) proporciona un mecanismo para el almacenamiento y la recuperación de datos que se modela en medios distintos a las relaciones tabulares utilizadas en las bases de datos relacionales. Las bases de datos NoSQL se utilizan cada vez más en aplicaciones web de big data y en tiempo real.

Java

Es un lenguaje de programación y una plataforma informática que fue comercializada por primera vez en 1995 por Sun Microsystems.

Unity

Unity es un motor de juegos multiplataforma desarrollado por Unity Technologies. Es especialmente popular para el desarrollo de juegos para móviles iOS y Android y se considera fácil de usar para los desarrolladores principiantes.

REST

La transferencia de estado representacional (REST) es un estilo arquitectónico de software que se creó para guiar el diseño y el desarrollo de la arquitectura de la World Wide Web. REST define un conjunto de restricciones sobre cómo debe comportarse la arquitectura de un sistema hipermedia distribuido a escala de Internet, como la Web.

IDE

Un entorno de desarrollo integrado (IDE) es una aplicación de software que ofrece amplias facilidades a los programadores informáticos para el desarrollo de software. Un IDE suele constar, como mínimo, de un editor de código fuente, herramientas de automatización de la construcción y un depurador.

SDK

El kit de desarrollo de software (SDK) es una colección de herramientas de desarrollo de software en un paquete instalable. Facilitan la creación de aplicaciones al contar con un compilador, un depurador y, en ocasiones, un marco de trabajo de software.

PCB

Una placa de circuito impreso (PCB) es una estructura sándwich laminada de capas conductoras y aislantes.

CNC

El control numérico por ordenador es el control automatizado de herramientas de mecanizado por medio de un ordenador. Una máquina CNC procesa una pieza de material para cumplir las especificaciones siguiendo instrucciones programadas codificadas y sin que un operador manual controle directamente la operación de mecanizado.

HTTP

El Protocolo de Transferencia de Hipertexto (HTTP) es un protocolo de capa de aplicación en el modelo del conjunto de protocolos de Internet para sistemas de información distribuidos, colaborativos e hipermedia.

HTTP GET

El método GET solicita que el recurso de destino transfiera una representación de su estado. Las peticiones GET sólo deben recuperar datos y no deben tener ningún otro efecto.

HTTP POST

El método POST solicita que el recurso de destino procese la representación incluida en la solicitud de acuerdo con la semántica del recurso de destino.

URL

Localizador uniforme de recursos es una referencia a un recurso web que especifica su ubicación en una red informática y un mecanismo para recuperarlo.

ASCII

ASCII, abreviatura de American Standard Code for Information Interchange, es un estándar de codificación de caracteres para la comunicación electrónica.

MAC

Dirección de control de acceso al medio, es un identificador único asignado a un controlador de interfaz de red para su uso como dirección de red en las comunicaciones dentro de un segmento de red.

IP

El Protocolo de Internet es el protocolo de comunicaciones de la capa de red del conjunto de protocolos de Internet para la transmisión de datagramas a través de las fronteras de la red. Su función de enrutamiento permite el trabajo en red y, esencialmente, establece Internet.

PHP

PHP es un lenguaje de programación de propósito general orientado al desarrollo web.

VPN

Red privada virtual, extiende una red privada a través de una red pública y permite a los usuarios enviar y recibir datos a través de redes compartidas o públicas como si sus dispositivos informáticos estuvieran conectados directamente a la red privada.

GitHub

Es un proveedor de alojamiento en Internet para el desarrollo de software y el control de versiones mediante Git.

Swift

Swift es un lenguaje de programación compilado de propósito general y multiparadigma desarrollado por Apple.

Flutter

Flutter es un kit de desarrollo de software de interfaz de usuario de código abierto creado por Google. Se utiliza para desarrollar aplicaciones multiplataforma para Android, iOS, Linux, macOS, Windows, Google Fuchsia, y la web a partir de un único código base.

Broker

Es el servidor que acepta mensajes publicados por clientes y los difunde entre los clientes suscritos.

Byte

Byte es la unidad de información de base utilizada en computación y en telecomunicaciones, y está compuesta por un conjunto ordenado de ocho bits.

Bits

Bit es el acrónimo de dígito binario. Un bit es un dígito del sistema de numeración binario.

Plugin

Un plugin es aquella aplicación que, en un programa informático, añade una funcionalidad adicional o una nueva característica al software.

Tinkercad

Es un programa de modelado 3D en línea gratuito que se ejecuta en un navegador web. Permite el diseño de circuitos eléctricos y modelado de diseños tridimensionales.

Script

Son fragmentos de código que tienen como objetivo realizar o añadir funciones a una determinada aplicación o servicio web.

XAMPP

Paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X, Apache, MariaDB/MySQL, PHP, Perl.

RFID

Sistema de almacenamiento y recuperación de datos remotos mediante el empleo de dispositivos como etiquetas, tarjetas o transpondedores.

MySQL

Sistema de gestión de bases de datos relacional considerada como la base de datos de código abierto más popular del mundo.

Bibliografía

[1] INE, «Indicadores demográficos básicos. Últimos datos,» 9 12 2021. [En línea]. Available: https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736177003&menu=ultiDatos&idp=1254735573002. [Último acceso: 15 Febrero 2022].

[2] «España - Piramide de población,» Datosmacro, [En línea]. Available: https://datosmacro.expansion.com/demografia/estructura-poblacion/espana. [Último acceso: 15 Febrero 2022].

[3] «España - Esperanza de vida al nacer,» INE, Diciembre 2021. [En línea]. Available: https://datosmacro.expansion.com/demografia/esperanza-vida/espana. [Último acceso: 15 Febrero 2022].

[4] G. d. España, «http://envejecimiento.csic.es/,» CSIC, 21 Enero 2021. [En línea]. Available: http://envejecimiento.csic.es/estadisticas/indicadores/residencias/index.html. [Último acceso: 15 Febrero 2022].

[5] «¿Cuántas residencias de mayores hay en España?,» Pensium, 8 Septiembre 2021. [En línea]. Available: https://pensium.es/residencias-mayores-espana/. [Último acceso: 15 Febrero 2022].

[6] J. L. R., «ComoFunciona una fotorresistencia,» [En línea]. Available: https://como-funciona.co/una-fotorresistencia/. [Último acceso: 26 Enero 2022].

[7] S. B. R, «Working with Light Dependent Resistor,» 9 Mayo 2019. [En línea]. Available: https://create.arduino.cc/projecthub/SBR/working-with-light-dependent-resistor-ldr-1ded4f. [Último acceso: 26 Enero 2022].

[8] AspenCore, «Electronics-tutorials - Light sensor,» 1 Octubre 2014. [En línea]. Available: https://www.electronics-tutorials.ws/io/io_4.html. [Último acceso: 25 Enero 2022].

[9] J. Froehlich, «Physical Computing - Photoresistors,» 2 Abril 2021. [En línea]. Available: https://makeabilitylab.github.io/physcomp/sensors/photoresistors.html. [Último acceso: 25 Enero 2022].

[10] S. Peddireddy, «Gyroscope Sensor Working and Its Applications,» 23 Marzo 2022. [En línea]. Available: https://www.elprocus.com/gyroscope-sensor/. [Último acceso: 28 Marzo 2022].

[11] R. González Vega, R. Hernández Bretones y B. Jiménez del Olmo, «Desarrollo de un sistema de detección de caídas,» 13 Junio 2016. [En línea]. Available: https://eprints.ucm.es/id/eprint/38704/1/MemoriaTFG.pdf. [Último acceso: 27 Enero 2022].

[12] A. Lady y D. Nosonowitz, «Force Sensitive Resistor (FSR),» 5 Abril 2014. [En línea]. Available: https://learn.adafruit.com/force-sensitive-resistor-fsr. [Último acceso: 25 Enero 2022].

[13] T. Agarwal, «What is an IR Sensor : Circuit Diagram & Its Working,» 30 Enero 2015. [En línea]. Available: https://www.elprocus.com/infrared-ir-sensor-circuit-and-working/. [Último acceso: 28 Enero 2022].

[14] Google, «Google Cloud,» 9 Enero 2014. [En línea]. Available: https://console.cloud.google.com/getting-started?hl=es. [Último acceso: 28 Diciembre 2021].

[15] R. Santos y S. Santos, «Randomnerdtutorial - ESP8266 Client-Server Wi-Fi Communication Between Two Boards (NodeMCU),» 6 Agosto 2019. [En línea]. Available: https://randomnerdtutorials.com/esp8266-nodemcu-client-server-wi-fi/. [Último acceso: 21 Noviembre 2021].

[16] R. Santos y S. Santos, «Randomnerdtutorials - ESP-NOW with ESP8266: Receive Data from Multiple Boards (many-to-one),» 15 Junio 2020. [En línea]. Available: https://randomnerdtutorials.com/esp-now-many-to-one-esp8266-nodemcu/. [Último acceso: 28 Noviembre 2021].

[17] J. Shi, «Github ESP-NOW Arduino library,» 3 Enero 2018. [En línea]. Available: https://github.com/yoursunny/WifiEspNow. [Último acceso: 29 Noviembre 2021].

[18] R. Santos y S. Santos, «Randomnerdtutorials - ESP32/ESP8266 Insert Data into MySQL Database using PHP and Arduino IDE,» 21 Enero 2020. [En línea]. Available: https://randomnerdtutorials.com/esp32-esp8266-mysql-database-php/. [Último acceso: 3 Diciembre 2021].

[19] G. Class, «GitHub - Google Cloud IoT JWT,» Google, 6 Abril 2018. [En línea]. Available: https://github.com/GoogleCloudPlatform/google-cloud-iot-arduino. [Último acceso: 7 Diciembre 2021].

[20] «Google - Cloud Firestore,» Google, 18 Mayo 2016. [En línea]. Available: https://firebase.google.com/docs. [Último acceso: 15 Diciembre 2021].

[21] R. DIY, «Hackster.io,» 5 Mayo 2020. [En línea]. Available: https://www.hackster.io/RoboticaDIY/send-data-from-arduino-to-nodemcu-and-nodemcu-to-arduino-17d47a. [Último acceso: 16 Diciembre 2021].