



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



TFG del Grado en Ingeniería  
Informática

**Sistema Domótico Inteligente**



Presentado por David Colmenero Guerra  
en Universidad de Burgos — 16 de diciembre  
de 2020

Tutor: Álgvar Arnaiz González  
Tutor: Alejandro Merino Gómez







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D.Álvar Arnaiz-González, profesor del departamento de Ingeniería Informática del área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. David Colmenero Guerra, con DNI 02287122W, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 16 de diciembre de 2020

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. Arnaiz-González, Álvaro

D. Merino Gómez, Alejandro





## Resumen

El proyecto pretende integrar diversas tecnologías para confeccionar una solución domótica generalista y de bajo costo. Es decir, se pretende publicar un producto final que pueda ser utilizado por quien lo desee para confeccionar su propio sistema domótico básico dentro de su domicilio y, para ello, se ha desarrollado un software así como vídeos explicativos para su correcta comprensión.

El sistema utiliza una placa Raspberry Pi con un Sistema Operativo propio con licencia GNU, para obtener información de diferentes APIS y páginas web, como pueden ser la hora a la que amanece, anochece, la temperatura por horas, velocidad del viento, etc; permitiendo al sistema tomar decisiones conforme a estos datos y controlar diferentes dispositivos eléctricos mediante la activación de relés. Para añadir usabilidad y facilitar la capacidad de transmisión de información del sistema, se desarrollará un Bot en Telegram con el que poder interactuar.

En el proyecto se incluye la documentación que se debe consultar antes de realizar cualquier tipo de instalación de telecomunicaciones o eléctrica en el ámbito doméstico pese a que, finalmente, el proyecto se basa en el REBT<sup>1</sup>. Esta documentación es la necesaria para implementar este proyecto de forma profesional por una empresa.

## Descriptores

Android, Raspberry Pi, GPIO, REBT, ICT, Autónomo, Sistema Domótico, Linux, Bot, Telegram, energético, Python, relé, Bash scripting...

---

<sup>1</sup>Reglamento Electrotécnico de Baja Tensión

## Abstract

The project aims to integrate various technologies to make a generalist and low-cost home automation solution. That is to say, it is intended to publish a final product that can be used by whoever wishes to make their own basic home automation system within their home and, for this, a fully commented code has been developed as well as explanatory videos for their correct understanding.

The system uses a Raspberry Pi board with its own operating system with a GNU license, to obtain information from different APIs and web pages, such as the time of sunrise, sunset, hourly temperature, wind speed, etc; allowing the system to make decisions based on this data and control different electrical devices by activating relays. To add usability and facilitate the information transmission capacity of the system, a Telegram Bot will be developed with which to interact.

The project includes the documentation that must be consulted before carrying out any type of telecommunications or electrical installation in the domestic sphere, despite the fact that, finally, the project is based on the REBT <sup>2</sup>. This documentation is necessary to implement this project professionally by a company.

## Descriptors

Android, Raspberry Pi, GPIO, REBT, ICT, Standalone, Domotic System, Linux, Bot, Telegram, energetic, Python, relay, Bash scripting  
...

---

<sup>2</sup>Low Voltage Electrotechnical Regulation



---

# Índice general

---

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
Objetivos del proyecto	3
2.1. Objetivos generales . . . . .	3
2.2. Objetivos personales . . . . .	4
Conceptos teóricos	5
3.1. Domótica . . . . .	5
3.2. GPIO . . . . .	6
3.3. API . . . . .	6
3.4. Json . . . . .	6
3.5. RETB . . . . .	7
3.6. Normativa de ICT . . . . .	7
3.7. Cableado estructurado . . . . .	7
3.8. WiFi . . . . .	8
3.9. UTP . . . . .	8
Técnicas, herramientas y componentes	11
4.1. Entorno Software . . . . .	11
4.2. Control de datos . . . . .	12
4.3. Técnicas manuales . . . . .	13

4.4. Metodologías . . . . .	14
4.5. Entorno de desarrollo del Proyecto . . . . .	14
4.6. Entorno físico . . . . .	16
4.7. Plataforma de interacción . . . . .	19
<b>Aspectos relevantes del desarrollo del proyecto</b>	<b>21</b>
5.1. Motivación del proyecto . . . . .	21
5.2. Formación necesaria . . . . .	21
<b>Trabajos relacionados</b>	<b>23</b>
<b>Trabajos relacionados</b>	<b>25</b>
6.1. Comparativa con otros proyectos . . . . .	25
6.2. Presentación de los proyectos . . . . .	25
6.3. Fortalezas y debilidades del proyecto . . . . .	26
<b>Conclusiones y Líneas de trabajo futuras</b>	<b>29</b>
<b>Bibliografía</b>	<b>31</b>

---

## Índice de figuras

---

3.1. Bobina UTP . . . . .	9
3.2. Diagrama de muestra de cable UTP . . . . .	9
3.3. Diagrama de muestra de cable FTP . . . . .	10
3.4. Diagrama de muestra de cable STP . . . . .	10
4.5. Especificaciones de Raspberry Pi 2B . . . . .	17
4.6. Estructura interna de un relé . . . . .	18
4.7. Imagen de una placa «protoboard». . . . .	18

---

## Índice de tablas

---

6.1. Comparativa de las características de los proyectos. . . . .	28
6.2. Comparativa de las características de los proyectos. . . . .	28
6.3. Comparativa de las características de los proyectos. . . . .	28

---

# Introducción

---

El concepto de domótica se acuñó para poder denominar a aquellos sistemas que disponen de la capacidad de automatizar elementos de una vivienda aportando confort, seguridad, mejoras energéticas, etc.

El término procede de la unión de dos palabras:

- Domo, procedente del griego «domus», que significa casa, vivienda.
- Por otra parte, «tica» procede de automática, cuyo significado es que dispone de la capacidad para realizar tareas por sí solo.

Formando una palabra cuyo significado aúna los términos de casa y automático.

Pese a conformarse el término de domótica en el año 1984, ésta aún es una gran desconocida, aunque se van introduciendo pequeños elementos automatizables como pueden ser las famosas bombillas que podemos encender o apagar desde diferentes plataformas.

Al carecer de movilidad desde la llegada de la pandemia de la COVID19 nos vemos en la necesidad de que nuestras viviendas dispongan de algún elemento de seguridad a un precio razonable, como puede ser un sistema que simule nuestra presencia en la vivienda para intentar evitar posibles percances, aumentando la sensación de confort.

Hay quien opta por opciones tradicionales de seguridad como el blindaje del domicilio para impedir el acceso o contratar a una empresa externa para que monitorice el domicilio. Los sistemas domóticos que desarrollaremos pretenden ser elementos complementarios.

Nuestro simulador de presencia funcionará de forma autónoma interactuando con persianas y luces desde una máquina Raspberry Pi mediante relés. De esta forma la vivienda parece estar ocupada de forma que ahuyentamos a potenciales delincuentes. También dispondremos de un estudio diario de la temperatura con la que podremos programar la calefacción. Además, este sistema domótico es fácilmente escalable con sistemas de acceso a la vivienda, telefonía IP, música, calefacción, telefonillo IP, etc.

Por otro lado, el que las persianas estén automatizadas generará un evidente ahorro energético, tanto en invierno como en verano, al hacer de pantalla térmica exterior.

En resumen, el proyecto se sitúa en un campo que cubre un conjunto de necesidades generales dentro de los domicilios, a un bajo coste y con relativa sencillez a la hora de implantarlo, lo cual hace que pueda llegar a un gran número de hogares.

---

# Objetivos del proyecto

---

Con este proyecto se pretende crear un sistema domótico automatizado que nos permita aumentar la seguridad y la sensación de confort y bienestar dentro de nuestros domicilios.

Para ello debemos alcanzar algunos objetivos funcionales mínimos.

## 2.1. Objetivos generales

- El sistema domótico funcionará de forma autónoma para que no interfiera en la vida diaria del inquilino y consiga facilitarle el día a día.
- El sistema domótico debe ser capaz de extraer información de Internet ya sea vía API o web scrapping.
- El sistema domótico debe poder conectarse a distintas instalaciones para actuar sobre éstas de una forma parametrizada.
- El usuario podrá interactuar con la máquina cuando lo desee.
- La instalación se podrá realizar con materiales de fácil acceso.
- Poder controlar elementos eléctricos desde la interfaz GPIO (General Purpose Input/Output, que significa Entrada/Salida de Propósito General).
- Debe ser un proyecto de bajo coste y asequible para que pueda llegar al mayor número de viviendas posible aumentando el beneficio social.

- Se pretende conseguir también un notable ahorro energético real que repercuta en el bolsillo de quien instale el sistema, además de ayudar a combatir el cambio climático consumiendo de una manera autónoma y responsable conforme a los parámetros del domicilio haciendo de éste un entorno más eficiente. Por ello, podremos controlar el encendido de la calefacción.

## **2.2. Objetivos personales**

- Poder aportar un dispositivo útil a la sociedad.
- Comprender la composición de un sistema domótico y aplicarlo.
- Desarrollar un sistema domótico con cierta complejidad y autonomía más allá de subir o bajar persianas o encender y apagar luces de forma programada o manual.
- Obtener conocimientos sobre trabajo con json desde Python.
- Aprender a controlar elementos eléctricos desde una Raspberry Pi.
- Aprender a utilizar una Raspberry Pi para fines domóticos utilizando GPIO.
- Poner en práctica conocimientos de cableado estructurado y REBT.
- Profundizar mis conocimientos sobre Linux.
- Aprender a utilizar el procesador de textos  $\text{\LaTeX}$ .



---

# Conceptos teóricos

---

Este punto nace ante la necesidad de enmarcar el proyecto dentro de las tecnologías y elementos que utilizaremos durante todo el proyecto y que no tienen por qué conocerse.

El término ‘domótica’ es el pilar principal del proyecto y, por ello, comenzaré explicando lo qué es y cómo lo enfocaremos:

## 3.1. Domótica

La domótica podemos definirla como aquel conjunto de elementos capaces de automatizar una vivienda aportando un beneficio. En nuestro caso, nuestro sistema domótico deberá controlar luces, persianas y calefacción permitiendo un aumento del confort y la seguridad, además de permitir un consumo eficiente de recursos a la hora de climatizar la vivienda.

*The home automation system can be applied to many areas including home security, lighting control, flame detection, smart heating, motion sensor and door control to provides its homeowner's comfort, security, energy efficiency (low operating costs) and convenience at all times. The Internet of Things (IoT) is anticipated to enable a variety of smart home services in which each service provides a set of home automation solutions. This proposed study consists of developing an automated home monitoring using Raspberry Pi that provides a customizable and cost-efficient platform for a smart home. [16]*

## 3.2. GPIO

Éstos son unos puertos de entrada y salida conformados en forma de pines que están albergados en las placas Raspberry Pi [12]. Con ellos enviaremos órdenes a otros dispositivos externos para que realicen las tareas que les designemos. En nuestro caso, servirán para controlar unos relés para conseguir una acción final.

## 3.3. API

Es el acrónimo de ‘Application Programming Interfaces’ que, traducido al castellano significa ‘Interfaz de Programación de Aplicaciones’. Estas interfaces nos sirven información que podremos utilizar en un desarrollo. Por ejemplo, para obtener la ubicación de la máquina Raspberry Pi accedemos a una API pública que nos devolverá información que podremos procesar a nuestro gusto. Podemos acceder a una URL como esta <http://ip-api.com/json/?fields=country,regionName,city,lat,lon,isp,query> y obtendremos los valores de país, región, ciudad, latitud, longitud, ISP y dirección IP. Estos valores, podremos recogerlos con BeautifulSoup<sup>3</sup> y procesarlos desde Python como json.

## 3.4. Json

Json [11], es el acrónimo en inglés de ‘JavaScript Object Notation’ y sirve para almacenar información de forma estructurada mediante etiquetas «key» y etiquetas «value» del siguiente modo:

```
1 {  
2     "country": "Spain",  
3     "regionName": "Madrid",  
4     "city": "Getafe"  
5 }
```

En ella, podemos ver que las etiquetas «key» son las que están a la izquierda de los dos puntos y las etiquetas de la derecha son las «value».

---

<sup>3</sup>Beautifulsoup es una librería de Python para hacer web scraping.

## 3.5. RETB

Es el acrónimo de Reglamento electrotécnico para baja tensión y en él se recoge la normativa eléctrica aplicable en domicilios. Esta norma acaba de ser actualizada y podemos disponer de la información en páginas oficiales como puede ser el BOE [6]. Del documento ICT-BT-21 [3] podemos extraer información para realizar las instalaciones eléctricas de nuestro sistema domótico como el número máximo de cables a introducir por un tubo eléctrico.

## 3.6. Normativa de ICT

Como figura en el *BOE 143, de 16 de junio de 2011, El Reglamento regulador de las infraestructuras comunes de telecomunicaciones para el acceso a los servicios de telecomunicación en el interior de las edificaciones, aprobado por el Real Decreto 346/2011, de 11 de marzo*: Debemos regirnos por esta normativa a la hora de hacer cualquier instalación de comunicaciones nueva dentro de domicilios. Podemos informarnos y ampliar información en la publicación en la sección de ICT en el BOE [4].

Por otro lado, disponemos de guías para instaladores con dibujos y tablas que facilitan la comprensión, como puede ser la documentación que publica Televés [7].

De este punto, obtendremos la norma para introducir cableado ICT [4] conforme a norma.

Tras hacer un estudio en mi domicilio, no necesitaré utilizar la normativa de ICT [4] porque toda la instalación se realizará mediante canales eléctricos, pero está bien conocer la norma para, en caso de necesitarla, poder hacer uso de ella correctamente.

## 3.7. Cableado estructurado

El establecimiento de un sistema de cableado estructurado consiste en la organización de los cables en un recinto conforme a una norma y constituye el nivel básico de cualquier red de comunicaciones. Al contar y cumplir con este estándar nos damos cuenta de que tendremos instalaciones limpias, uniformes, seguras y escalables, facilitando la supervisión, el mantenimiento y posibles migraciones de tecnologías. Un sistema de cableado genérico

dispone de tres subsistemas, Troncal, de Edificio y Horizontal. En nuestro proyecto únicamente trataremos con el subsistema horizontal.

En este proyecto no contaremos con un gran número de cables, pero no está de más realizar una instalación lo más correctamente posible con unas normas de referencia

### 3.8. WiFi

Es una tecnología de comunicaciones de forma inalámbrica o “Wireless”. WiFi<sup>4</sup> es el acrónimo traducido de “Fidelidad Inalámbrica”. Estas tecnologías inalámbricas se rigen por la norma IEEE 802.11 [14]. En la web oficial del organismo podremos comprobar qué estándares dentro del 802.11 están vigentes y cuáles no.

### 3.9. UTP

Es un tipo de cableado de datos que se compone de 4 pares de cables sin apantallar que están albergados dentro de una camisa de PVC, ver imagen 3.1.

Existen diferentes tipos de cables de datos: UTP, STP, FTP:

- Los cables **UTP** (del inglés «Unshielded Twisted Pair» o «Par trenzado no apantallado») no disponen de protección ante interferencias electromagnéticas. Ver imagen 3.2.
- Los cables **FTP** (del inglés «Foiled Twisted Pair» o «Par trenzado con pantalla global») disponen de una pantalla global contra interferencias electromagnéticas dentro de la camisa de PVC que recoge los 4 pares destinados a transmisión de datos. Ver imagen 3.3.
- Los cables **STP** (del inglés «Shielded Twisted Pair» o «Par trenzado apantallado») disponen de una pantalla contra interferencias electromagnéticas por cada par de cables pero, además, también cuentan con una malla metálica exterior. Ver imagen 3.4.

En nuestro caso utilizaremos UTP puesto que no necesitamos un apantallamiento ya que no transmitiremos datos y tampoco tendremos un alto grado de interferencias.

---

<sup>4</sup>Traducción del inglés: 'Wireless Fidelity'



Figura 3.1: Bobina UTP con muestra del mismo con corte de camisa exterior.  
Imagen de <https://solarmat.es> [9]. [https://creativecommons.org/licenses/by-sa/3.0/deed.es\\_ES](https://creativecommons.org/licenses/by-sa/3.0/deed.es_ES)

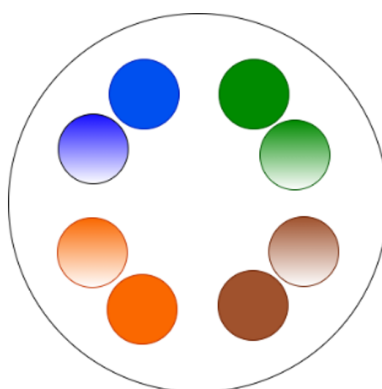


Figura 3.2: Diagrama de muestra de cable UTP

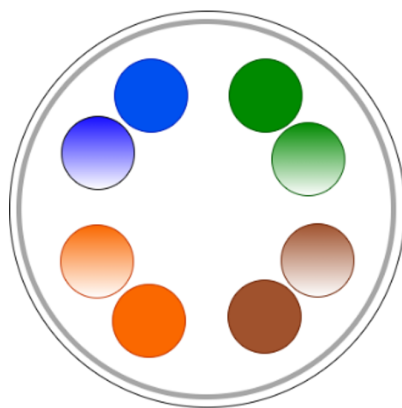


Figura 3.3: Diagrama de muestra de cable FTP. Podemos observar la camisa exterior del cable

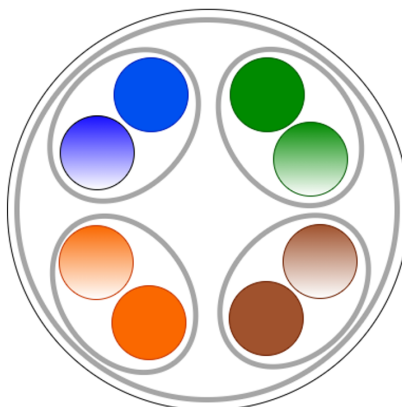


Figura 3.4: Diagrama de muestra de cable STP. Vemos camisa exterior y por pares

---

# Técnicas, herramientas y componentes

---

Durante el proyecto se han utilizado diferentes tecnologías, herramientas y componentes que son imprescindibles y necesitan conocerse antes de continuar con el proyecto. Para optar por éstos y no por otros, se ha realizado una valoración que queda plasmada en este apartado a modo de justificación.

## 4.1. Entorno Software

### Raspbian (Distribución Linux)

Como he comentado anteriormente, pretendo correr una distribución Linux en nuestro microPC. Las placas Raspberry Pi disponen de unas distribuciones de Linux desarrolladas expresamente para su hardware desde la Raspberry Pi Foundation. De esta manera conseguimos que el entorno esté diseñado para el hardware donde será ejecutado incluyendo, además, utilidades preinstaladas para explotarlas más fácil y eficientemente. Una de estas distribuciones optimizadas y orientadas a estas placas es Raspbian [12] o Raspberry Pi OS, que incluye software orientado a la educación, programación y otras de uso general. Algunas de estas aplicaciones son Python [5] (Lenguaje de programación que pretende que se desarrolle código de una forma sencilla, rápida, poco costosa y legible), Scratch [10] (Simulador amigable para aprender programación.) o Java[17] (Lenguaje de programación multiplataforma que utiliza una máquina virtual transparente para el usuario para ejecutarse), entre otros.

## Entorno de desarrollo Bash

- Herramientas valoradas: [Vi](#), [Vim](#), [Nano](#).
- Herramienta elegida: [Nano](#).

Nuestro Sistema Operativo Raspbian[12], al ser una distribución de Linux [15], dispone de líneas de comandos, procesadores de texto plano y editores de texto integrados. Éste es un editor de textos básico que facilita la interacción con él pese a que Vi o Vim son mucho más potentes.

## Entorno de desarrollo Python

- Herramientas valoradas: [Jetbrains PyCharm](#) y [Jupyter Notebook](#).
- Herramienta elegida: [Jupyter Notebook](#).

Jupyter Notebook es un entorno de desarrollo interactivo y open source, basado en cuadernos que estructuran el código pudiendo ejecutarlo todo o parte. Dispone de una interfaz limpia, ligera e interactiva que nos permite programar en 40 lenguajes, incluyendo Python [5].

## 4.2. Control de datos

### Web Scraping

Es una técnica utilizada para extraer información de una página web utilizando las etiquetas de que dispone el propio lenguaje interpretado de HTML (del inglés «HyperText Markup Language» o lenguaje de marcas de hipertexto) para organizar elementos dentro de una página web, de forma que se introduce dentro de una etiqueta y subetiquetas hasta llegar al contenido del elemento requerido. Podemos entenderlo como si fueran contenedores lógicos configurables. En nuestro caso, podremos utilizarlo desde Python [5] sirviéndonos de la librería «beautifulsoup» siempre que necesitemos obtener información de una página web.

### APIS

- API situación geográfica



En primer lugar estuve haciendo pruebas con la API de [www.ifconfig.me/ip](http://www.ifconfig.me/ip) que devuelve la provincia en la que se encuentra tu IP pública pero quería una información más precisa ya que no tendremos la misma temperatura en El Escorial que en Aranjuez. Por ello, opté por <http://ip-api.com> que sí obtiene correctamente la ciudad desde la que nos conectamos.

#### ■ API Tiempo

Al principio probé la API de [www.weatherapi.com](http://www.weatherapi.com) pero me entregaba únicamente la hora de salida y puesta del sol, lo cual es correcto para el control básico de las persianas pero quería llevar el proyecto más allá obteniendo además, una previsión de las temperaturas para el día siguiente pudiendo trabajar con ésta haciendo gráficos y poder decidir si encenderemos la calefacción o no. Y, por ello, opté por probar con la API de [www.climacell.co](http://www.climacell.co) que además, aseguran que es un 60 % más fiable que otras APIs ya que obtiene información de teléfonos móviles, cámaras y otros servicios online.

### json

La librería json [11] para Python [5] y Python v3 nos permite, entre otros, parsear el código json [11] de archivos mediante la estructura de *key:value*. En nuestro caso, tras obtener información de las APIs trataremos dicha información como json [11] gracias a su librería para Python [5].

## 4.3. Técnicas manuales

### Tirada de cable con guía pasacables

El procedimiento a seguir es el siguiente:

1. Se abren las tapas de dos cajas de derivación próximas.
2. Se introduce una guía pasacables (herramienta plástica con la forma de cuerda para introducir cables por canalizaciones) por el extremo de uno de los tubos dentro de la caja hasta llegar al otro extremo.
3. Se asegura el cable a uno de los extremos de la guía pasacables.
4. Se tira del otro extremo de la guía pasacables hasta conseguir sacar el cable por éste.

## 4.4. Metodologías

### Scrum

Scrum[20] es un marco de trabajo para el desarrollo de software mediante la metodología ágil en el que se busca realizar un trabajo colaborativo de desarrollo incremental. Se aplica una metodología basada en milestones y sprints de forma iterativa.

### Modularidad

Utilizaremos la modularidad para subdividir la aplicación en pequeños subprogramas que tienen una pequeña funcionalidad. De esta manera es más fácil detectar errores y escalar el proyecto. Además, teniendo clara la entrada y salida de cada uno de los módulos se pueden lanzar pruebas a cada uno de los módulos para comprobar su correcto funcionamiento mejorando, también, el mantenimiento.

## 4.5. Entorno de desarrollo del Proyecto

### Control de versiones o CVS, Concurrent Versioning System

- Herramientas valoradas: [Git](#), [SVN](#).
- Herramienta elegida: [Git](#).

Git es un software destinado al control de versiones software en el que se registran los cambios producidos en el mismo, facilitando la integración de código por parte de cualquiera de los integrantes del proyecto. La diferencia más notable entre ellos es que Git es distribuido y SVN es un sistema centralizado. Significa que Git nos permite disponer de una copia en cada uno de los equipos desde los que se trabaje haciendo un clonado del repositorio, mientras que en SVN se trabaja en la nube.

### Hosting del Repositorio

- Herramientas valoradas: [Github](#) y [Bitbucket](#).
- Herramienta elegida: [Github](#).

Ambas opciones de hosting de repositorios funcionan de forma similar aunque GitHub incorpora opciones como la revisión de código, Kanban, Wikis o tableros entre otros que me han hecho decantarme por esta opción. Ésta, es la plataforma principal de trabajo, que a su vez es una red social de código donde cualquiera puede contribuir en proyectos públicos y Open Source.

## Gestión del proyecto

- **Herramientas valoradas:** [MS.Teams](#), [ZenHub](#), [GitHub Projects](#), [Trello](#), [Jira](#), [Board](#), [Monday](#), [Zube](#), [Clubhouse](#).
- **Herramienta elegida:** [ZenHub](#).

Es la única solución de colaboración en equipo integrada en GitHub y nos permite planificar hojas de ruta, generar informes, gestionar con Kanban, gestión ágil del proyecto mostrando la situación del proyecto para conseguir aumentar la productividad del equipo.

## Editor del proyecto

- **Herramientas valoradas:** [Atom](#), [Visual Studio Code](#), [Sublime](#).
- **Herramienta elegida:** [Atom](#).

Es un editor de código y texto creado por GitHub integrando las funciones de Git y GitHub, lo que nos facilita trabajar en nuestro equipo y replicar los cambios en Git, GitHub y ZenHub. Además, es un software multiplataforma, licencia open source, completamente personalizable, con temas y múltiples plugins, autocompletado y fácil navegación. En este proyecto se utilizará para publicar las actualizaciones del proyecto.

## Dibujos, diagramas y planos

- **Herramientas valoradas:** [Fritzing](#), [Paint](#), [Paint3D](#), [Photoshop](#), y [Draw.io](#).
- **Herramienta elegida:** [Fritzing](#) y [Draw.io](#).

Realmente no pude decidirme por entre Fritzing y draw.io puesto que cada uno está orientado a un tipo de tarea:

Fritzing es un software con licencia Open Source[18] que nos permite realizar material electrónico fácilmente, aunque desde hace algún tiempo debemos hacer un pequeño desembolso por la descarga. En mi caso con el fin de entregar diagramas de calidad los haré con este software.

Por otro lado, Draw.io está pensado para hacer planos y dibujos que no están orientados a la electrónica, por lo que es muy útil para hacer otro tipo de diagramas.

## Procesador de textos $\text{\LaTeX}$

- **Herramientas valoradas:**  $\text{\LaTeX}$ , MS Word, Sublime, Overleaf.
- **Herramienta elegida:**  $\text{\LaTeX}$  y Overleaf.

$\text{\LaTeX}$  es un editor de textos open source[18] con alta calidad tipográfica que trabaja con etiquetas permitiéndonos separar nuestro contenido del estilo del mismo. Para escribir el proyecto en  $\text{\LaTeX}$  utilizaré Overleaf por su capacidad de compilado instantáneo de forma que se puede comprobar como modificas la redacción en tiempo real.

## 4.6. Entorno físico

### RaspberryPi

En nuestro proyecto tendremos el control de la instalación domótica desde una Raspberry Pi[12]. Para dar un enfoque muy general, podemos decir que las placas RaspberryPi [12] son microordenadores que disponen de poca potencia si las comparamos con equipos usuales, pero disponen de suficiente potencia para llevar a cabo este tipo de proyectos.

Se diseñaron en su origen por la RaspBerry Pi Foundation [12] en el Reino Unido para dotar de equipos informáticos a los centros de estudios a un bajo coste, pero el proyecto ha evolucionado para poder desarrollar, además, otras muchas tareas como puede ser nuestro caso, que la utilizaremos como ‘núcleo’ de toda nuestra instalación domótica y, será donde configuremos todo el entorno domótico de la vivienda. Estas placas pueden ejecutar con agilidad distribuciones Linux [15] y, desde sus distribuciones podemos interactuar con sus famosos «GPIO», ver imagen 4.5

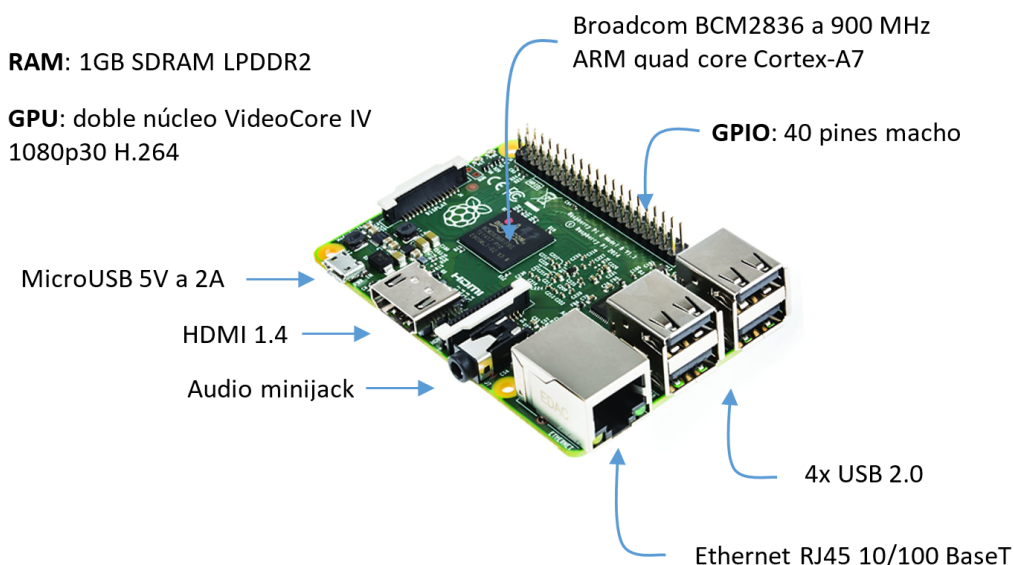


Figura 4.5: Especificaciones de Raspberry Pi 2B. Imagen de <https://rasberryparatorpes.net> modificada por el autor del TFG [9].

## Relé

Es un dispositivo electromagnético que desempeña la misma función de un interruptor, es decir, con nuestros relés, dejaremos pasar la energía eléctrica, o no, a nuestros dispositivos. Los relés se activan mediante impulsos eléctricos que abren o cierran el circuito según se predisponga. Podemos verlo en la imagen 4.6.

## Placa de Pruebas o ProtoBoard

Es un tablero electrónico para realizar pruebas. Protoboard es la agrupación de los términos ingleses “prototype board”. Esta protoboard la he instalado para poder hacer fácilmente el interconexión entre los cables que llegan de los relés y los que van a la Raspberry Pi, evitando posibles tirones y movimiento de cables a la hora de hacer alguna manipulación.

Éstas, disponen de tres zonas diferenciadas (Ver imagen 4.7):

- **Canal Central:** Está situada en el medio de la placa y es donde se colocan los circuitos.
- **Buses:** Se sitúan en los extremos de la placa y disponen de dos líneas:

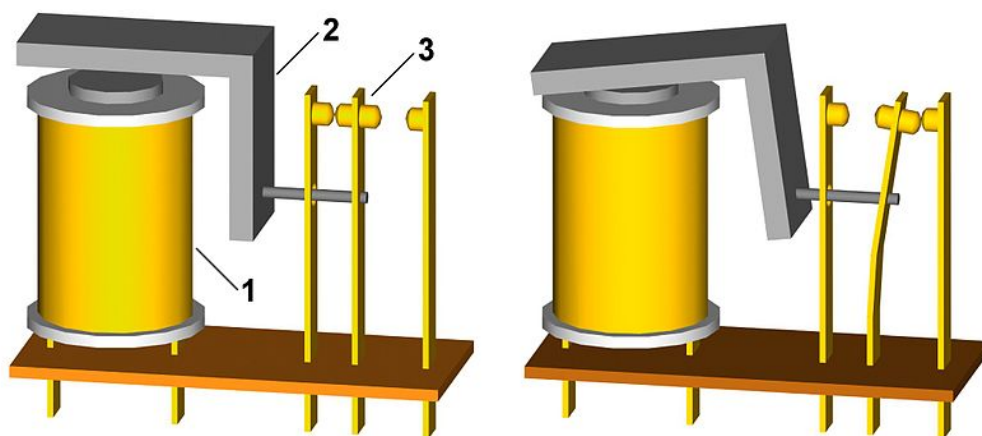


Figura 4.6: Estructura interna de un relé. Imagen de [https://commons.wikimedia.org/\[13\]](https://commons.wikimedia.org/[13])

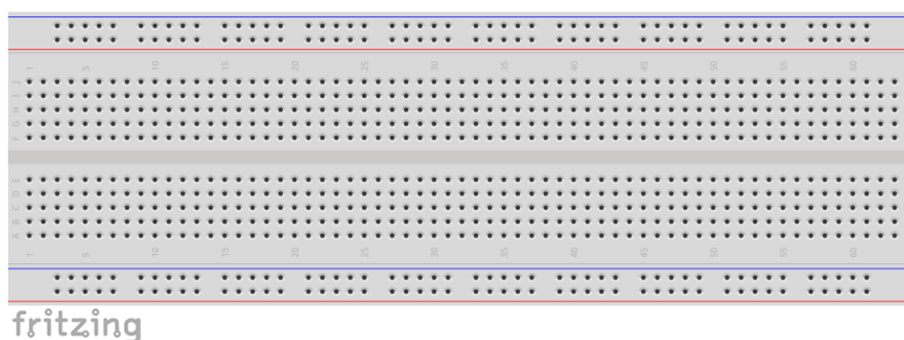


Figura 4.7: Imagen de una placa «protoboard».

**Línea roja:** Bus positivo o de voltaje.

**Línea azul:** Bus negativo o de tierra.

- **Pistas:** Se sitúan en la zona central de la placa y, conducen en sentido contrario de las líneas rojas y azul.

## Router

Es un dispositivo que nos permite interconectar diferentes redes de datos. En mi caso dispongo de un router con WiFi integrado para poder dotar a la Raspberry Pi de salida a Internet.

## 4.7. Plataforma de interacción

- **Herramientas valoradas:** [Telegram](#), [Flask](#).
- **Herramienta elegida:** [Telegram](#).

Para finalizar el proyecto con una interfaz de interacción se valoraron las opciones de desarrollar una página web ligera o una aplicación de mensajería. En primer lugar se propuso hacer una aplicación en Flask pero pareció más novedoso y potente hacer un Bot de Telegram de forma que podamos interactuar con él y enviarnos información bajo demanda. Además, se eliminó el buscar hosting y dominio para albergar los servicios propios del sistema domótico abaratando más los costes, además de incrementar la seguridad al utilizar mensajería cifrada.





---

# Aspectos relevantes del desarrollo del proyecto

---

En este punto se recogerán los aspectos más relevantes del desarrollo comentando en cada caso las decisiones tomadas para llegar a nuestros objetivos haciendo un resumen de la experiencia práctica del proyecto, de cómo se solucionaron los problemas encontrados en cada caso y la relevancia que tuvieron en el alcance total del proyecto.

## 5.1. Motivación del proyecto

El proyecto se me ocurrió hace años cuando me emancipé a una vivienda con las persianas motorizadas que me parecieron que podrían hacer una función mayor si dispusieran de cierto equipamiento, y cogió fuerza con la llegada de la pandemia de la COVID19. Este año al salir en conversaciones que había quien no podía acercarse a sus segundas viviendas y estaban preocupados por una posible ocupación. Tras darle vueltas a esta situación surgió la idea de crear un sistema domótico para poder ayudar a quien lo precise con este proyecto.

## 5.2. Formación necesaria

El proyecto requirió muchas horas de búsqueda de ideas por la web ya que no existe información de como realizar una instalación de estas características sino que existen muchos pequeños proyectos amateur que se centran en cubrir una pequeña necesidad; siendo proyectos que no disponen de un respaldo documental detrás, abundan las soluciones de profesionales de

otros campos que quieren probar a hacer sus propias soluciones de carácter amateur.

Por ello, para poder desarrollar el proyecto me vi en la necesidad de visitar numerosas páginas web de distinta índole para hacerme a la idea de cómo podría enfocar el proyecto. Aunque, el mayor hándicap a la hora de realizar este proyecto es la desinformación, por lo que me apoyé sobre todo en el REBT y en mis conocimientos básicos de motores fruto de formación pasada.

La información de cómo funcionan los GPIO la obtuve tras realizar el curso de: “Control de GPIO con Python en Raspberry Pi” de Programo Ergo Sum [19] Aunque, en parte la información de la web de [bujarra.com](http://bujarra.com) [8] está desactualizada, esta publicación me ayudó a comprender cuál era el funcionamiento real de un relé y como conectarlo a los GPIO.

Otro punto importante a la hora de enfocar correctamente el proyecto fue el estudio del REBT [6], de su apartado BT-21 [3], del reglamento de ICT [4] y los estándares de comunicaciones como son el IEEE802.11 [14] y el TIA568 [2] [1]. Todos ellos necesarios para realizar un proyecto de nivel profesional y documentalmente respaldado.

---

## **Trabajos relacionados**

---



---

## Trabajos relacionados

---

Desde la aparición de elementos electrónicos accesibles y asequibles se ha intentado, con mayor o menor fortuna, generar sistemas que nos ayuden en nuestro día a día. A continuación se exponen proyectos con objetivos similares.

### 6.1. Comparativa con otros proyectos

En este apartado mostraré otras opciones de acometer un proyecto similar y compararé los principales puntos de éstos con el presente proyecto para poder tener una visión global entre estas opciones.

### 6.2. Presentación de los proyectos

#### **Interacción remota para elementos de un hogar mediante una red de sensores actuadores**

Se trata de un proyecto de control domótico que se centra en un conjunto de sensores instalados en el domicilio utilizando nodos intermedios para realizar el control de los dispositivos finales y la comunicación con los sensores.

En la tabla [6.1](#) podemos ver algunas características de este proyecto. En las tablas se representará con la primera palabra del nombre del proyecto: "Interacción".

## Diseño e implementación de un sistema domótico basado en Raspberry Pi

Se trata de un proyecto de control domótico que se centra en un conjunto de sensores instalados en el domicilio.

- Url del trabajo: [https://e-archivo.uc3m.es/bitstream/handle/10016/26313/TFG\\_Hector\\_Santos\\_Senra.pdf?sequence=1&isAllowed=y](https://e-archivo.uc3m.es/bitstream/handle/10016/26313/TFG_Hector_Santos_Senra.pdf?sequence=1&isAllowed=y)

En la tabla 6.2 podemos ver algunas características de este proyecto. En las tablas se representará con la primera palabra del nombre del proyecto: "Diseño".

### 6.3. Fortalezas y debilidades del proyecto

Las principales fortalezas del proyecto son:

- Éste es un proyecto que cualquiera puede implementar fácilmente en su domicilio haciendo una copia local del repositorio y haciendo unas sencillas configuraciones en el archivo de configuración.
- En este proyecto se ha tenido en cuenta la eficiencia de recursos y de consumo de la propia máquina Raspberry Pi[12], por lo que no se han incluido servicios extraordinarios como puede ser un servidor de bases de datos o web, como sucede en otros proyectos. Se ha optado por incluir un bot que 'escucha' las órdenes y actúa conforme a un pequeño archivo de configuración y actúa contra la máquina utilizando los lenguajes nativos de la máquina. De esta manera, el gasto en cómputo es menor y también el utilizado en mantener los servicios en memoria, quedando recursos suficientes para continuar haciendo un uso normal de la máquina para cualquier otra tarea. Además, al no tener servidores levantados y con puertos de comunicaciones abiertos, aumentamos la seguridad de nuestra instalación.
- Se ha procurado utilizar los lenguajes integrados de forma nativa en el Sistema Operativo para evitar posibles fallos de integración entre ellos y aumentando la fiabilidad del sistema. Estos son Python [5] y Bash [15].
- El sistema automatizado de nuestro proyecto se nutre de la información contrastable obtenida de API's reconocidas.

- Se puede interactuar con el sistema domótico de varias formas. La más cómoda es mediante el bot de Telegram creado en exclusiva para controlar las funcionalidades principales de nuestro sistema domótico, aunque también se puede contactar de forma remota desde otra máquina mediante VNC <sup>5</sup> siempre que se esté en la misma red. Se puede interactuar con el bot mediante aplicaciones Android o IOs, o desde navegadores de uso extendido como Chrome, Firefox o Edge, entre otros.
- No se necesitan elementos intermedios de ningún tipo, únicamente tenemos nuestra Raspberry Pi [12] y los actuadores o relés 4.6. De esta manera evitamos posibles fallos de los elementos intermedios.
- Se ha preferido cablear la instalación para evitar que los elementos receptores puedan quedar en un estado no deseado así como evitar reconfigurar elementos ante cualquier posible cambio de alguno de los elementos como pueden ser el dongle WiFi 3.8 o el router. También, al dejar libres los canales WiFi, evitamos contribuir a la actual saturación de frecuencias en los espectros de 2.4GHz y 5GHz [14].

Podemos ver en la tabla 6.3 de comparativa de proyectos lo anteriormente expuesto.

Las principales debilidades del proyecto son:

- Hace falta cierta soltura a la hora de trabajar con el cableado UTP 4.3.
- Es necesaria una línea de datos para que se actualicen los parámetros de los scripts.

---

<sup>5</sup>VNC: "Virtual Network Computing", programa de control remoto de equipos informáticos basado en una estructura cliente-servidor

Características	Interacción
Proyecto libre	✓
No precisa montar servicios	×
No requiere lenguajes no nativos en el SO	×
Obtiene información externa contrastada	×
Interacción multiplataforma	✓
No necesita nodos intermedios	×
Cableado entre elementos	×
WiFi entre elementos	✓

Tabla 6.1: Comparativa de las características de los proyectos.

Características	Diseño
Proyecto libre	×
No precisa montar servicios	×
No requiere lenguajes no nativos en el SO	×
Obtiene información externa contrastada	×
Interacción multiplataforma	✓
No necesita nodos intermedios	✓
Cableado entre elementos	✓
WiFi entre elementos	×

Tabla 6.2: Comparativa de las características de los proyectos.

Características	SDI	Interacción	Diseño
Proyecto libre	✓	✓	×
No precisa montar servicios	✓	×	×
No requiere lenguajes no nativos en el SO	✓	×	×
Obtiene información externa contrastada	✓	×	×
Interacción multiplataforma	✓	✓	✓
No necesita nodos intermedios	✓	×	✓
Cableado entre elementos	✓	×	✓
WiFi entre elementos	×	✓	×

Tabla 6.3: Comparativa de las características de los proyectos.



---

## **Conclusiones y Líneas de trabajo futuras**

---

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.



---

## Bibliografía

---

- [1] ANSI/TIA/EIA. *ANSI/TIA/EIA568-Balanced Twisted-Pair Cabling Components*. [Página web Oficial, Noviembre de 2020].
- [2] ANSI/TIA/EIA568. *ANSI/TIA/EIA568-General Requirements*. [Página web Oficial, Noviembre de 2020].
- [3] BOE. *ICT-BT-21*. [Documentación técnica, Octubre de 2020].
- [4] BOE. *Normativa de ICT*. [Documentación técnica, Octubre de 2020].
- [5] BOE. Python software foundation. [Página web Oficial].
- [6] BOE. *Reglamento Electrotécnico de Baja Tensión*, 2020. [Documentación técnica, Octubre de 2020].
- [7] BOE+Televis. *Normativa de ICT Televis*. [Página web Televis, Octubre de 2020].
- [8] Bujarra.com. Rbp uso de relé y detector de movimiento. [Página web].
- [9] Creative Commons. Norma creative commons. [Página web Oficial, Noviembre de 2020].
- [10] Grupo Lifelong Kindergarten del MIT Media Lab. Scratch. [Página web Oficial].
- [11] ECMA. Javascript object notation. [Página web Oficial, Noviembre de 2020].
- [12] Raspberry Pi Foundation. Raspberry pi foundation. [Página web Oficial].

- [13] GNU. *Free Software Foundation, Inc.* [Página web Oficial, Noviembre de 2020].
- [14] IEEE. *Normativa IEEE802.11.* [Página web Oficial del estándar, Noviembre de 2020].
- [15] Linux. Linux. [Página web Oficial, Noviembre de 2020].
- [16] Arslan Musaddiq, Farhan Amin, Yousaf Zikria, Rojeena Bajracharya, and Sung Kim. Raspberry pi-based smart home automation system for internet of things. 05 2017.
- [17] Oracle Technology Network. Java. [Página web Oficial].
- [18] OpenSource. Opensource. [Página web Oficial, Noviembre de 2020].
- [19] programoergosum.com. Control de gpio con python en raspberry pi. [Página web].
- [20] Wikipedia. *Metodología Scrum.* [Página web, Noviembre de 2020].