



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Sistema Domótico Inteligente  
Documentación Técnica**



Presentado por  
en Universidad de Burgos — 16 de diciembre  
de 2020

Tutor: Álgvar Arnaiz-González  
Tutor: Alejandro Merino Gómez



---

# Índice general

---

<b>Índice general</b>	<b>I</b>
<b>Índice de figuras</b>	<b>III</b>
<b>Índice de tablas</b>	<b>IV</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Estudio de viabilidad . . . . .	4
<b>Apéndice B Especificación de Requisitos</b>	<b>5</b>
B.1. Introducción . . . . .	5
B.2. Objetivos generales . . . . .	5
B.3. Funcionamiento autónomo . . . . .	5
B.4. Escalabilidad . . . . .	5
B.5. Acceso multiplataforma . . . . .	5
B.6. Ahorro energético . . . . .	6
B.7. Objetivos técnicos . . . . .	6
B.8. Catálogo de requisitos . . . . .	6
B.9. Especificación de requisitos . . . . .	6
<b>Apéndice C Especificación de diseño</b>	<b>7</b>
C.1. Introducción . . . . .	7
C.2. Diseño de datos . . . . .	7
C.3. Diseño procedimental . . . . .	7
C.4. Diseño arquitectónico . . . . .	7

<b>Apéndice D Documentación técnica de programación</b>	<b>9</b>
D.1. Introducción . . . . .	9
D.2. Estructura de directorios . . . . .	9
D.3. Manual del programador . . . . .	9
D.4. Compilación, instalación y ejecución del proyecto . . . . .	9
D.5. Pruebas del sistema . . . . .	9
<b>Apéndice E Documentación de usuario</b>	<b>11</b>
E.1. Introducción . . . . .	11
E.2. Requisitos de usuarios . . . . .	11
E.3. Instalación . . . . .	11
E.4. Manual del usuario . . . . .	11
<b>Bibliografía</b>	<b>13</b>

---

## Índice de figuras

---

---

## Índice de tablas

---

## *Apéndice A*

---

# Plan de Proyecto Software

---

### A.1. Introducción

En este apéndice del anexo se plasmará toda la información de seguimiento (sprints de Scrum), el análisis de costes y viabilidad del proyecto.

### A.2. Planificación temporal

Al comenzar el proyecto se determinó que se utilizaría una metodología ágil para hacer el desarrollo del proyecto pero esta decisión ha terminado siendo un handicap producido por la falta de documentación y la dudosa credibilidad de la mayoría de páginas webs y documentación, se traducido en un gran coste temporal. Por otro lado, el redactar el proyecto en  $\text{\LaTeX}$  desde la plataforma de Overleaf no refleja el tiempo real invertido puesto que no se realizan cambios en el repositorio de forma automática, hasta que compilo, descargo y actualizo el repositorio manualmente.

Ante todo, se ha intentado seguir un mínimo de pautas **Scrum** pese a no existir un grupo de trabajo real con unas tareas diarias y roles definidos:

- Las tareas fueron siempre semanales en forma de «sprints».
- Al finalizar el sprint se hace la entrega del trabajo elaborado en la semana y se determinan las próximas tareas.
- Tras determinar las tareas se definen los «milestones» y los «issues».
- Para hacer el seguimiento de las tareas se ha utilizado en tablero **Kanban** de **ZenHub**.

- Tras finalizar el sprint se puede comprobar el trabajo mediante los *gráficos burndown*.

Las reuniones de trabajo han sido consensuadas y planificadas los jueves de cada semana, con el siguiente resultado:

### **Sprint 00 - 01/10/2020 - 07/10/2020**

En la primera reunión nos reunimos Álar Arnaz González y yo. En ella hice la propuesta de temática del TFG y, además expuse diferentes ideas y enfoques sobre el proyecto en la que se determinó a grandes rasgos la posible viabilidad del proyecto. Los objetivos de este sprint fueron la búsqueda de repositorios y otros proyectos para tomar ideas o si era posible hacer algún fork. También, el estudio la estructura del proyecto y herramientas software e interfaces a utilizar.

«“IMAGENES BURNDOWN”»

### **Sprint 01 - 08/10/2020 - 14/10/2020**

Tras determinar el enfoque final de proyecto, se determinó que los objetivos de este sprint fueron la búsqueda de información relevante, otros proyectos y recursos que puedan servir de utilidad como pueden ser APIS. Se propusieron algunas APIS online para realizar pruebas. Las tareas se descompusieron en unidades pequeñas de trabajo para poder afrontarlas semanalmente mediante sprints.

### **Sprint 02 - 15/10/2020 - 21/10/2020**

Los objetivos de este sprint fueron el buscar repositorios, APIS, servicios y tecnologías con las que darle forma al proyecto. Se decidió generar algún tipo de aplicación desde la que poder controlar la instalación haciéndola más amigable y con mejor capacidad de interacción. Se investigan opciones.

### **Sprint 03 - 22/10/2020 - 28/10/2020**

Los objetivos de este sprint fueron determinar los componentes hardware necesarios para completar el proyecto y se realiza la compra de material. Se presenta la opción de hacer una aplicación mediante el Bot de Telegram.



**Sprint 04 - 29/10/2020 - 05/10/2020**

Se incorpora D.Alejandro Merino Gómez al que se le presenta el proyecto y se le concede acceso a los repositorios para que pueda co-tutorizar el proyecto. Los objetivos de este sprint fueron la instalación física de la instalación y la tirada de cable inicial.

**Sprint 05 - 06/11/2020 - 12/11/2020**

Se repasan los cambios propuestos y se incluyen nuevos. Se hace el seguimiento de la instalación y se comentan las fotos y el estado de la instalación. Los objetivos de este sprint son todos aquellos que sean necesarios para la adecuación del software al proyecto. Esto incluye la actualización del sistema operativo e instalación de software adicional. Además se deben valorar opciones para controlar los GPIO.

**Sprint 06 - 13/11/2020 - 19/11/2020**

Se compone un tablero de pruebas de forma que se puede controlar el encendido de una bombilla desde nuestra Raspberry Pi. Se graba un vídeo explicativo y se diseñan unos diagramas explicativos.

**Sprint 07 - 20/11/2020 - 26/11/2020**

Se realiza una investigación sobre como se puede implantar el bot en nuestro proyecto, se realiza el primer código de pruebas y se presentan las primeras pruebas con el bot. Además, se proponen correcciones en la redacción.

**Sprint 08 - 27/11/2020 - 03/12/2020**

Se genera la primera automatización de la parte automática del proyecto utilizando CRON. Se implanta los scripts finales de geolocalización y cálculo de temperaturas que utilizaremos para generar el control de nuestro sistema domótico. Se presentan las primeras funcionalidades del bot, se proponen cambios y mejoras.

**Sprint 09 - 04/12/2020 - 10/12/2020**

Se proponen revisan las funcionalidades y se proponen cambios y mejoras en estas.

**Sprint 10 - 11/12/2020 - 17/12/2020**

Se comprueban las funcionalidades proponiendo cambios funcionales y de estilos de forma que se pueda interactuar contra el bot y se muestren los mensajes en un formato más atractivo. También se propone unificar las rutas de trabajo de los diferentes scripts.

**Sprint 11 - 18/12/2020 - 24/12/2020****Sprint 12 - 30/12/2020 - 06/01/2021****Sprint 13 - 07/01/2020 - 13/01/2020****Sprint 14 - 14/01/2020 - 20/01/2020****A.3. Estudio de viabilidad**

**Viabilidad económica**

**Viabilidad legal**

## *Apéndice B*

---

# **Especificación de Requisitos**

---

### **B.1. Introducción**

### **B.2. Objetivos generales**

### **B.3. Funcionamiento autónomo**

Se pretende que tras pocas configuraciones el sistema sea capaz de funcionar extrayendo información de Internet y decidiendo que acción realizar en cada caso a partir de ese momento.

### **B.4. Escalabilidad**

Como cualquier proyecto de calidad debe ser escalable, esto es, debe existir la posibilidad de que sea ampliado fácilmente.

### **B.5. Acceso multiplataforma**

En este caso dispondremos de un acceso GUI (del inglés «Graphical User Interface» o «Interfaz Gráfica de Usuario») o CLI<sup>1</sup> (Línea de comandos) a nuestro equipo.

---

<sup>1</sup>Traducción del inglés: 'Command Line Interface'

## B.6. Ahorro energético

El ahorro energético se produce al permitir que la temperatura exterior incida, o no, en la vivienda para conseguir las condiciones deseadas optimizando el consumo de recursos.

COMPROBAR UBICACIÓN

## B.7. Objetivos técnicos

- Debe disponer de un Sistema Operativo GNU como puede ser una distribución de Linux. En nuestro caso utilizaremos Raspbian.
- Posibilidad de interacción multiplataforma para aumentar la flexibilidad a la hora de interactuar con los equipos.
- También, debe ser fácilmente escalable para permitir la posibilidad de continuar con el desarrollo.
- Utilizar Git y Github como sistema de control de versiones, de Zenhub como herramienta de gestión de proyectos y Atom para trabajar con el control de versiones.
- Utilización de Frizking para realizar los diseños electrónicos de la instalación.
- Utilización de [www.draw.io](http://www.draw.io) para realizar los diseños de la instalación.
- Tratamiento de datos utilizando lenguaje Python.
- Utilización de diferentes APIs para obtener datos.
- Correcto conectorizado electrónico de todos los elementos según RETB.

## B.8. Catálogo de requisitos

## B.9. Especificación de requisitos

## *Apéndice C*

---

# **Especificación de diseño**

---

- C.1. Introducción
- C.2. Diseño de datos
- C.3. Diseño procedimental
- C.4. Diseño arquitectónico



## *Apéndice D*

---

# **Documentación técnica de programación**

---

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución del proyecto
- D.5. Pruebas del sistema





## *Apéndice E*

---

# **Documentación de usuario**

---

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario



---

## **Bibliografía**

---