



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería  
Informática

Mejora de interfaz para planta  
piloto y aplicación web (v2.0)  
Documentación Técnica



Presentado por Iván Cortés Aliende  
en Universidad de Burgos — 29 de junio  
de 2022

Tutores: Dr. Daniel Sarabia Ortiz - Dr.  
Alejandro Merino Gómez





---

# Índice general

---

|   |     |
|---|-----|
| Índice general                          | I   |
| Índice de figuras                       | III |
| Índice de tablas                        | V   |
| A Plan de Proyecto Software             | 1   |
| A.1. Introducción .....                 | 1   |
| A.2. Planificación temporal.....        | 1   |
| A.3. Estudio de viabilidad.....         | 18  |
| B Especificación de Requisitos          | 21  |
| B.1. Introducción .....                 | 21  |
| B.2. Objetivos generales.....           | 22  |
| B.3. Catálogo de requisitos.....        | 22  |
| B.4. Especificación de requisitos ..... | 24  |
| C Especificación de diseño              | 35  |
| C.1. Introducción .....                 | 35  |
| C.2. Diseño de datos .....              | 35  |
| C.3. Diseño procedimental .....         | 36  |
| C.4. Diseño arquitectónico .....        | 36  |
| D Documentación técnica de programación | 39  |

|      |  |    |
|------|--|----|
| D.1. | Introducción .....                                     | 39 |
| D.2. | Estructura de directorios.....                         | 39 |
| D.3. | Manual del programador .....                           | 41 |
| D.4. | Compilación, instalación y ejecución del proyecto..... | 42 |
| D.5. | Pruebas del sistema.....                               | 43 |
| E    | Documentación de usuario .....                         | 49 |
| E.1. | Introducción .....                                     | 49 |
| E.2. | Requisitos de usuarios .....                           | 49 |
| E.3. | Instalación .....                                      | 50 |
| E.4. | Manual del usuario .....                               | 50 |
|      | Bibliografía .....                                     | 57 |

---

## Índice de figuras

---

|   |    |
|---|----|
| Figura A.1: Tablero Kanban del proyecto. ....                         | 2  |
| Figura A.2: Tarea #1 en Trello y su descripción.....                  | 4  |
| Figura A.3: Tarea #2 en Trello y su descripción.....                  | 5  |
| Figura A.4: Tarea #3 en Trello y su descripción.....                  | 6  |
| Figura A.5: Tarea #4 en Trello y su descripción.....                  | 7  |
| Figura A.6: Tarea #5 en Trello y su descripción.....                  | 9  |
| Figura A.7: Tarea #6 en Trello y su descripción.....                  | 10 |
| Figura A.8: Tarea #7 en Trello y su descripción.....                  | 11 |
| Figura A.9: Tarea #8 en Trello y su descripción.....                  | 12 |
| Figura A.10: Tarea #9 en Trello y su descripción.....                 | 13 |
| Figura A.11: Tarea #10 en Trello y su descripción.....                | 14 |
| Figura A.12: Tarea #11 en Trello y su descripción.....                | 15 |
| Figura A.13: Tarea #12 en Trello y su descripción.....                | 16 |
| Figura A.14: Tarea #13 en Trello y su descripción.....                | 17 |
| Figura A.15: Tarea #14 en Trello y su descripción.....                | 18 |
| Figura B.16: Casos de uso de la aplicación local.....                 | 25 |
| Figura B.17: Casos de uso de la aplicación web.....                   | 25 |
| Figura C.18: Diagrama de secuencia de los distintos componentes de la |    |

|   |    |
|---|----|
| aplicación.....   | 36 |
| Figura C.19: Diagrama de clases UML de la Aplicación web. ....  | 38 |
| Figura D.20: Descarga del repositorio mediante el entorno web. ....   | 41 |
| Figura D.21: Proyecto abierto en el IDE Microsoft Visual Studio 2019.....   | 42 |
| Figura D.22: Menú contextual con las opciones relativas a la compilación y<br>debug señaladas. ....                 | 43 |
| Figura D.23: Microsoft Visual Studio 2019 depurando la aplicación local.<br>Controles de depuración señalados. .... | 43 |
| Figura D.24: Gráfico de actividad de la aplicación local en SonarQube.....  | 46 |
| Figura D.25: Gráfico de actividad de la aplicación web en SonarQube.....  | 47 |
| Figura E.26: Búsqueda de puerto en uso. ....  | 51 |
| Figura E.27: Formulario de despliegue de la aplicación web.....   | 52 |
| Figura E.28: Página principal de la aplicación web. ....  | 53 |
| Figura E.29: Página de información de la base de datos. ....  | 54 |
| Figura E.30: Información presente en la página Acerca De. ....  | 55 |

---

## Índice de tablas

---

|   |    |
|---|----|
| Tabla A.1: Listado con los materiales, su valor y su tiempo de amortización. .. | 19 |
| Tabla A.2: Listado con los materiales y coste amortizado. ....                  | 19 |
| Tabla A.3: Desglose del coste de personal. ....                                 | 19 |
| Tabla B.4: Aplicación local - Caso de uso 1. ....                               | 26 |
| Tabla B.5: Aplicación Local - Caso de uso 2. ....                               | 27 |
| Tabla B.6: Aplicación web – Caso de uso 1. ....                                 | 28 |
| Tabla B.7: Aplicación web – Caso de uso 2. ....                                 | 29 |
| Tabla B.8: Aplicación web – Caso de uso 3. ....                                 | 30 |
| Tabla B.9: Aplicación web – Caso de uso 4. ....                                 | 31 |
| Tabla B.10: Aplicación web – Caso de uso 5. ....                                | 32 |
| Tabla B.11: Aplicación web – Caso de uso 6. ....                                | 33 |





## *Apéndice A*

---

# Plan de Proyecto Software

---

### A.1. Introducción

Apartado destinado a explicar el proceso de planificación del desarrollo del proyecto. En concreto, me centraré en la planificación temporal, económica y legal del mismo.

### A.2. Planificación temporal

En este apartado se tratará el programa temporal que se diseñó y utilizó durante todo el desarrollo del proyecto.

Se descartó emplear la metodología ágil *Scrum* por el alto número de personas que necesita. En contramedida, se optó por emplear la metodología ágil *Kanban*.

*Kanban* se basa en dividir el trabajo en una serie de tarjetas representadas en un tablero que serán movidas entre diferentes listas, cada una de ellas representando una etapa distinta del desarrollo. Los principios básicos de esta metodología son redactados a continuación [1].

1. Visualizar el flujo de trabajo

2. Limitar el número de tareas en desarrollo
3. Medir y administrar el flujo de trabajo
4. Hacer explícitas las políticas de procesos
5. Utilizar modelos para identificar las oportunidades de mejora

En este caso se optó por utilizar la herramienta digital *Trello* para la gestión del tablero. Se puede observar el tablero empleado en la Figura A.1.

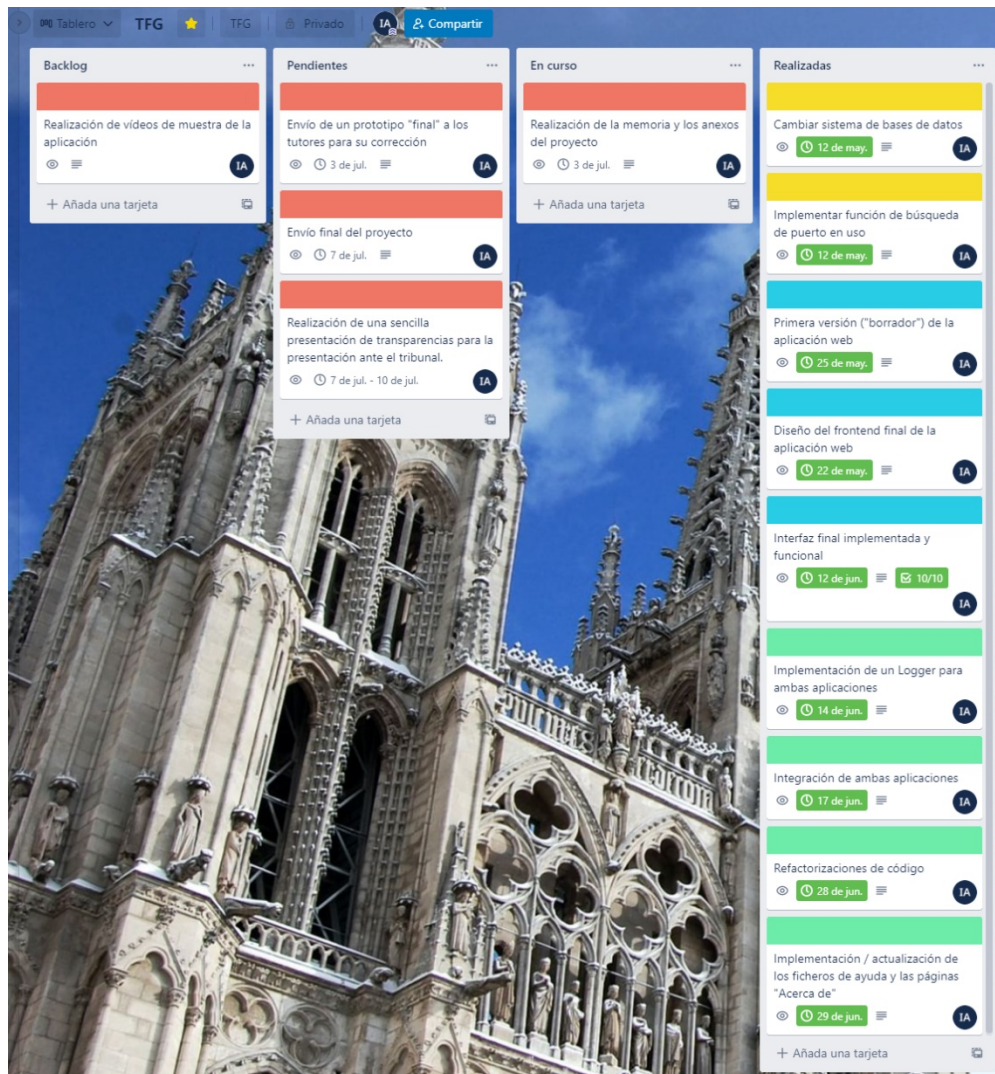


Figura A.1: Tablero Kanban del proyecto.

En este caso se optó por un sistema de tareas basado en la utilización de cuatro listados, cada uno de ellos representando un estado de tareas concreto.

- **Backlog.** En este listado figuran las tareas de prioridad baja o aquellas que todavía no están confirmadas a realizarse.

- **Pendientes.** Aquí se encuentran las tareas confirmadas pero que todavía no han empezado a realizarse.
- **En curso.** Tareas en desarrollo. Para este proyecto en concreto me puse un límite de dos tareas en curso de forma simultánea como máximo.
- **Realizadas.** Tareas ya finalizadas.

Además, asigné a cada tarea un color en función de su naturaleza.

- **Amarillo.** Tareas relativas a la aplicación local de entornos Windows.
- **Azul.** Tareas relativas a la aplicación web.
- **Verde.** Tareas relativas a ambas aplicaciones.
- **Rojo.** Tareas relativas a ninguna de las aplicaciones. Sobre todo, en este color se encontraban las tareas relativas a la documentación del proyecto.

En los siguientes subapartados se tratarán cada una de las tareas de forma individual.

## Tarea #1



|                     |   |
|---------------------|---|
| <b>Título</b>       | Cambiar sistema de bases de datos   |
| <b>Descripción</b>  | El proyecto inicialmente se basa en Microsoft SQL Server. Cambiar por un SGBD portable y liviano. |
| <b>Color</b>        | Amarillo  |
| <b>Miembros</b>     | Iván Cortés Aliende   |
| <b>Fecha límite</b> | 15 de junio de 2022   |
| <b>Apartados</b>    | Ninguno   |

Figura A.2: Tarea #1 en Trello y su descripción.

## Tarea #2



|                     |  |
|---------------------|--|
| <b>Título</b>       | Implementar función de búsqueda de puerto en uso   |
| <b>Descripción</b>  | Actualmente los alumnos tienen que buscar a mano cuál de todos los puertos de su equipo es el que está recibiendo la comunicación en serie para empezar a escuchar. La idea es simplificar esta labor mediante la implementación de un botón que escuche todos los puertos y detecte cual es el que se está empleando. |
| <b>Color</b>        | Amarillo   |
| <b>Miembros</b>     | Iván Cortés Aliende  |
| <b>Fecha límite</b> | 12 de mayo de 2022   |
| <b>Apartados</b>    | Ninguno  |

Figura A.3: Tarea #2 en Trello y su descripción.

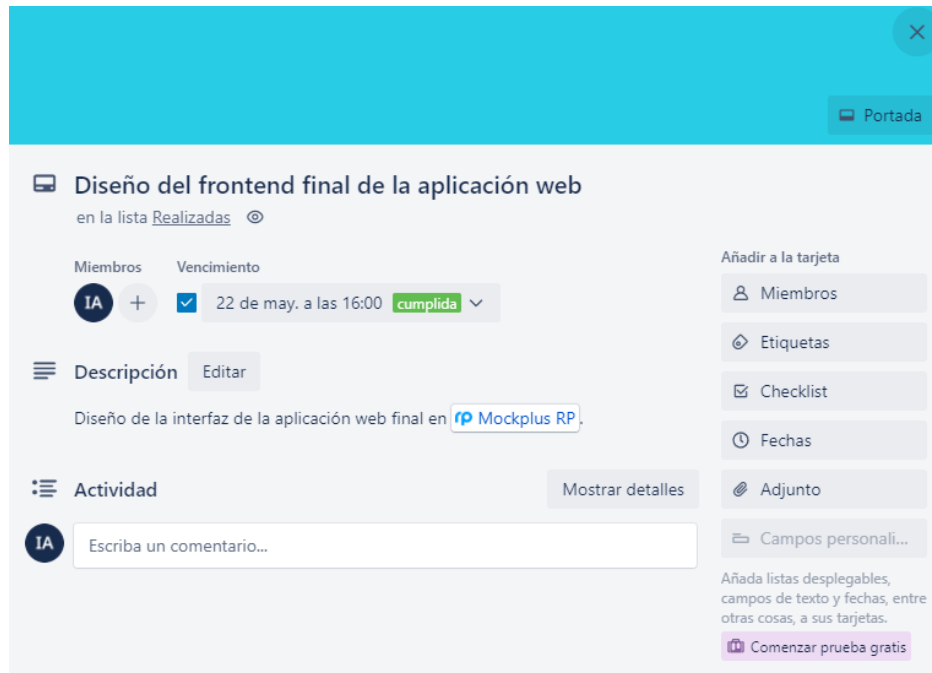
## Tarea #3



|                     |   |
|---------------------|---|
| <b>Título</b>       | Primera versión ("borrador") de la aplicación web                   |
| <b>Descripción</b>  | Implementar un prototipo funcional de la aplicación web en ASP.NET. |
| <b>Color</b>        | Azul  |
| <b>Miembros</b>     | Iván Cortés Aliende   |
| <b>Fecha límite</b> | 25 de mayo de 2022  |
| <b>Apartados</b>    | Ninguno   |

Figura A.4: Tarea #3 en Trello y su descripción.

## Tarea #4

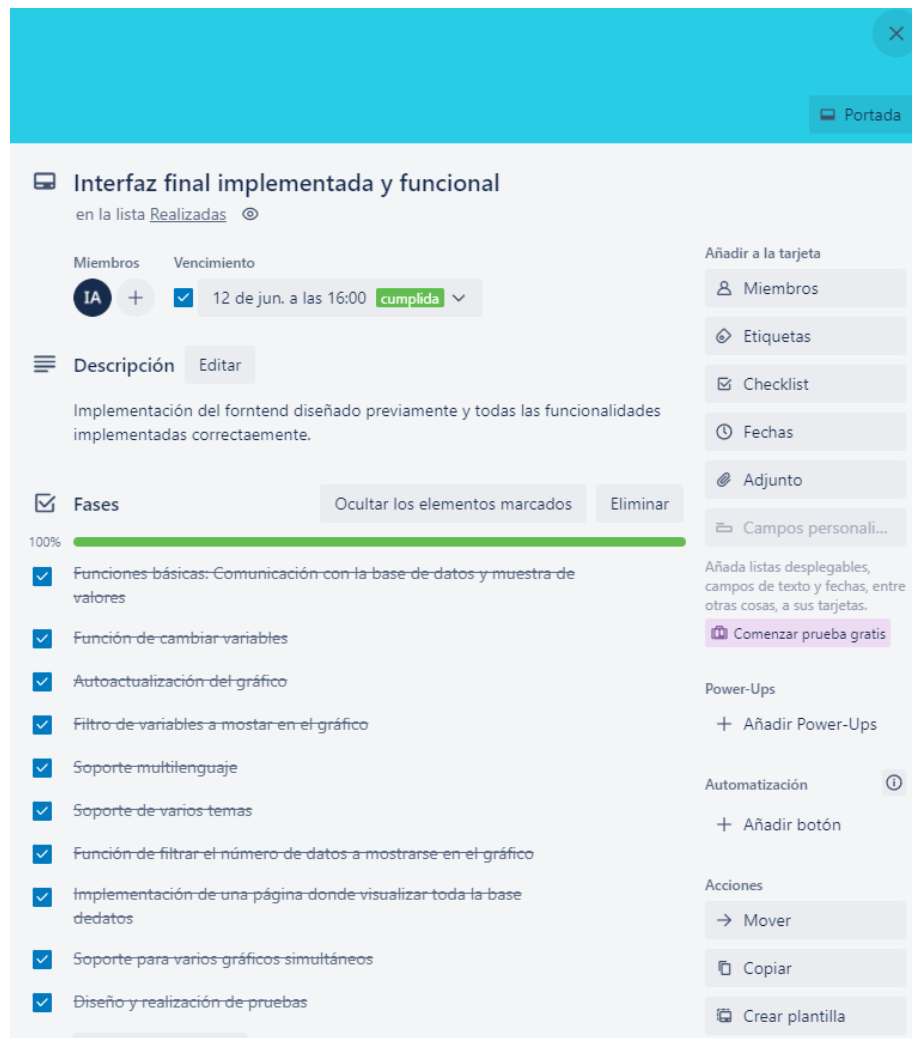


|                     |  |
|---------------------|--|
| <b>Título</b>       | Diseño del frontend final de la aplicación web                   |
| <b>Descripción</b>  | Diseño de la interfaz de la aplicación web final en Mockplus RP. |
| <b>Color</b>        | Azul   |
| <b>Miembros</b>     | Iván Cortés Aliende  |
| <b>Fecha límite</b> | 22 de mayo de 2022   |
| <b>Apartados</b>    | Ninguno  |

Figura A.5: Tarea #4 en Trello y su descripción.



## Tarea #5

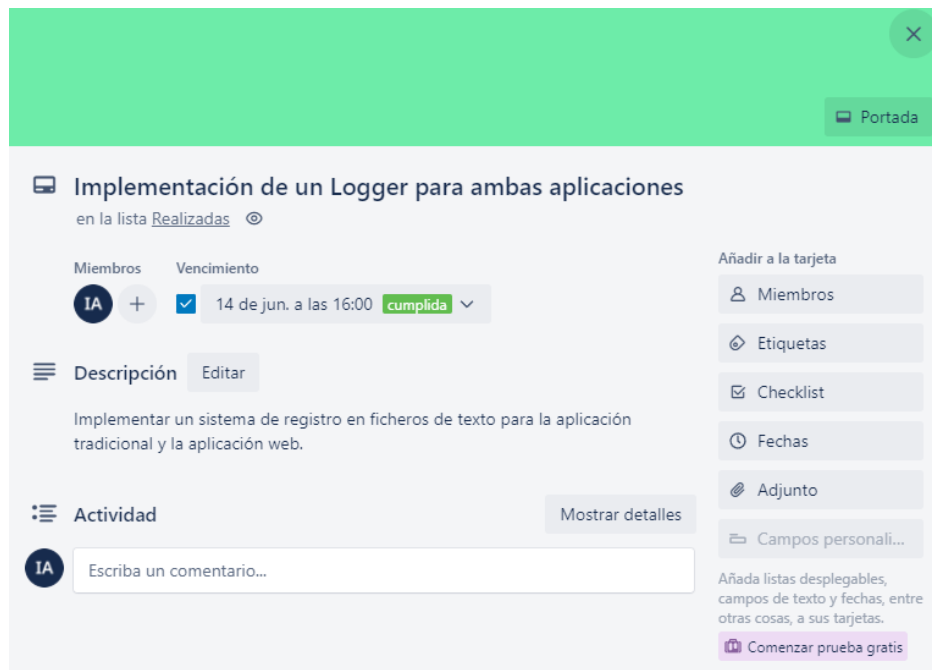


|                     |  |
|---------------------|--|
| <b>Título</b>       | Interfaz final implementada y funcional  |
| <b>Descripción</b>  | Implementación del forntend diseñado previamente y todas las funcionalidades implementadas correctamente.  |
| <b>Color</b>        | Azul   |
| <b>Miembros</b>     | Iván Cortés Aliende  |
| <b>Fecha límite</b> | 12 de junio de 2022  |
| <b>Apartados</b>    | <ul style="list-style-type: none"> <li>- Funciones básicas: Comunicación con la base de datos y muestra de valores</li> <li>- Función de cambiar variables</li> <li>- Automatización del gráfico</li> <li>- Filtro de variables a mostrar en el gráfico</li> <li>- Soporte multilenguaje</li> <li>- Soporte de varios temas</li> </ul> |

- 
- Función de filtrar el número de datos a mostrarse en el gráfico
  - Implementación de una página donde visualizar toda la base de datos
  - Soporte para varios gráficos simultáneamente
  - Diseño y realización de pruebas
- 

Figura A.6: Tarea #5 en Trello y su descripción.

## Tarea #6



|                     |   |
|---------------------|---|
| <b>Título</b>       | Implementación de un Logger para ambas aplicaciones   |
| <b>Descripción</b>  | Implementar un sistema de registro en ficheros de texto para la aplicación tradicional y la aplicación web. |
| <b>Color</b>        | Verde   |
| <b>Miembros</b>     | Iván Cortés Aliende   |
| <b>Fecha límite</b> | 14 de junio de 2022   |
| <b>Apartados</b>    | Ninguno   |

Figura A.7: Tarea #6 en Trello y su descripción.

## Tarea #7



|                     |   |
|---------------------|---|
| <b>Título</b>       | Integración de ambas aplicaciones   |
| <b>Descripción</b>  | Implementar, mediante un menú de lanzamiento, la aplicación web en la aplicación tradicional. |
| <b>Color</b>        | Verde   |
| <b>Miembros</b>     | Iván Cortés Aliende   |
| <b>Fecha límite</b> | 17 de junio de 2022   |
| <b>Apartados</b>    | Ninguno   |

Figura A.8: Tarea #7 en Trello y su descripción.

## Tarea #8



|                     |   |
|---------------------|---|
| <b>Título</b>       | Refactorizaciones de código   |
| <b>Descripción</b>  | Utilizando la herramienta SonarQube para la detección de malas praxis en el código, refactorizarlo. |
| <b>Color</b>        | Verde   |
| <b>Miembros</b>     | Iván Cortés Aliende   |
| <b>Fecha límite</b> | 28 de junio de 2022   |
| <b>Apartados</b>    | Ninguno   |

Figura A.9: Tarea #8 en Trello y su descripción.

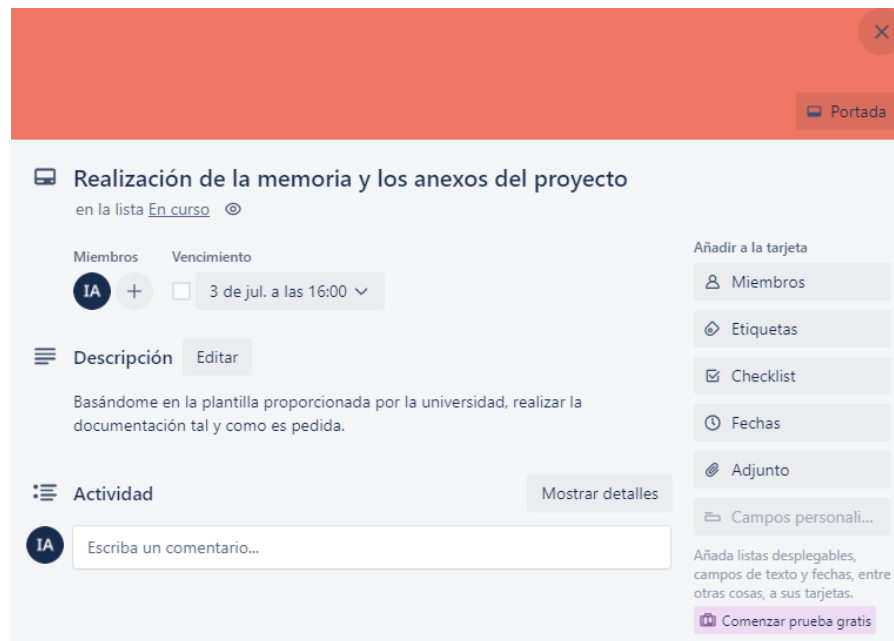
## Tarea #9



|                     |   |
|---------------------|---|
| <b>Título</b>       | Implementación / actualización de los ficheros de ayuda y las páginas "Acerca de"   |
| <b>Descripción</b>  | En el caso de la aplicación tradicional, actualización de los sistemas de ayuda y "Acerca de" y en el caso de la aplicación web implementación desde 0. |
| <b>Color</b>        | Verde   |
| <b>Miembros</b>     | Iván Cortés Aliende   |
| <b>Fecha límite</b> | 29 de junio de 2022   |
| <b>Apartados</b>    | Ninguno   |

Figura A.10: Tarea #9 en Trello y su descripción.

## Tarea #10



|                     |   |
|---------------------|---|
| <b>Título</b>       | Realización de la memoria y los anexos del proyecto   |
| <b>Descripción</b>  | Basándome en la plantilla proporcionada por la universidad, realizar la documentación tal y como es pedida. |
| <b>Color</b>        | Rojo  |
| <b>Miembros</b>     | Iván Cortés Aliende   |
| <b>Fecha límite</b> | 3 de julio de 2022  |
| <b>Apartados</b>    | Ninguno   |

Figura A.11: Tarea #10 en Trello y su descripción.

## Tarea #11

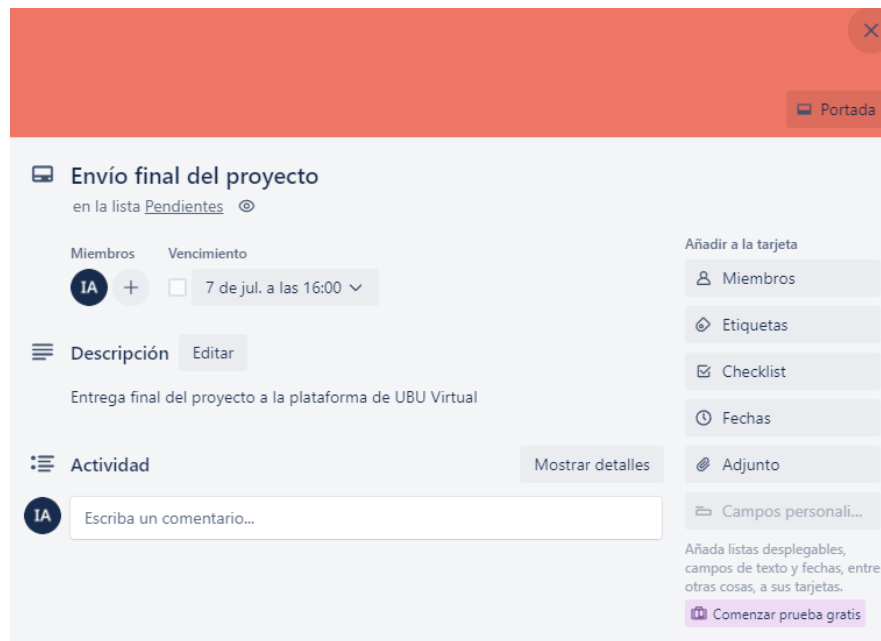


|                     |  |
|---------------------|--|
| <b>Título</b>       | Envío de un prototipo "final" a los tutores para su corrección         |
| <b>Descripción</b>  | Envío del código y la documentación a los tutores para su visto bueno. |
| <b>Color</b>        | Rojo   |
| <b>Miembros</b>     | Iván Cortés Aliende  |
| <b>Fecha límite</b> | 3 de julio de 2022   |
| <b>Apartados</b>    | Ninguno  |

Figura A.12: Tarea #11 en Trello y su descripción.



## Tarea #12



|                     |  |
|---------------------|--|
| <b>Título</b>       | Envío final del proyecto                                   |
| <b>Descripción</b>  | Entrega final del proyecto a la plataforma de UBU Virtual. |
| <b>Color</b>        | Rojo   |
| <b>Miembros</b>     | Iván Cortés Aliende  |
| <b>Fecha límite</b> | 7 de julio de 2022   |
| <b>Apartados</b>    | Ninguno  |

Figura A.13: Tarea #12 en Trello y su descripción.

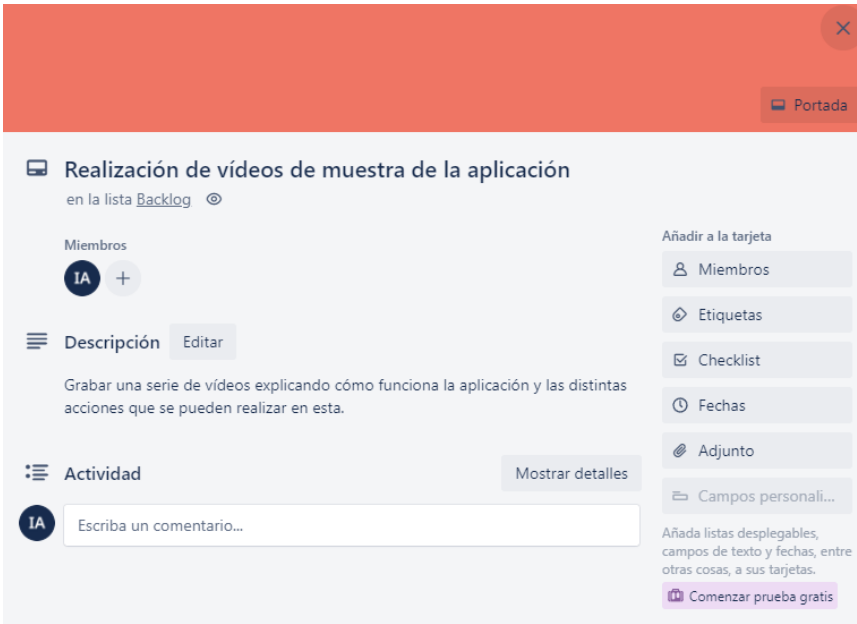
## Tarea #13



|                     |  |
|---------------------|--|
| <b>Título</b>       | Realización de una sencilla presentación de transparencias para la presentación ante el tribunal |
| <b>Descripción</b>  |  |
| <b>Color</b>        | Rojo   |
| <b>Miembros</b>     | Iván Cortés Aliende  |
| <b>Fecha límite</b> | 7 de julio de 2022   |
| <b>Apartados</b>    | Ninguno  |

Figura A.14: Tarea #13 en Trello y su descripción.

Tarea #14



|              |  |
|--------------|--|
| Título       | Realización de vídeos de muestra de la aplicación  |
| Descripción  | Grabar una serie de vídeos explicando cómo funciona la aplicación y las distintas acciones que se pueden realizar en esta. |
| Color        | Rojo   |
| Miembros     | Iván Cortés Aliende  |
| Fecha límite |  |
| Apartados    | Ninguno  |

Figura A.15: Tarea #14 en Trello y su descripción.

A.3. Estudio de viabilidad

Viabilidad económica

Apartado destinado a la comparación costes - beneficios del proyecto desarrollado, si este hubiera sido realizado en un ambiente empresarial, teniendo en cuenta los costes materiales y de personal.

Empezaremos con los costes materiales, tanto hardware como software (Tabla A.1). Se tendrán en cuenta los productos que se han empleado en el desarrollo, pero en su versión destinada al ámbito profesional. Además, se descartarán los materiales gratuitos empleados, para simplificación del estudio.

La lista de los materiales empleados para el desarrollo han sido los siguientes:

| Material   | Valor     | Amortizado en |
|--|-----------|---------------|
| Ordenador y periféricos                              | 1.000,00€ | 5 años        |
| Planta piloto  | 80,00€    | 1 año         |
| <a href="#">Microsoft Windows 11 Pro Edition</a>     | 259,00€   | 5 años        |
| <a href="#">Microsoft Visual Studio Professional</a> | 1.540,00€ | 3 años        |

Tabla A.1: Listado con los materiales, su valor y su tiempo de amortización.

En la Tabla A.2 podemos observar los costes que han supuesto estos materiales al proyecto, teniendo en cuenta que el periodo de desarrollo del proyecto ha sido de aproximadamente **cuatro meses** (marzo 2022 - junio 2022).

| Material                             | Coste amortizado |
|--------------------------------------|------------------|
| Ordenador y periféricos              | 66,67€           |
| Planta piloto                        | 26,67€           |
| Microsoft Windows 11 Pro Edition     | 17,27€           |
| Microsoft Visual Studio Professional | 171,11€          |
| <b>TOTAL</b>                         | <b>281,71€</b>   |

Tabla A.2: Listado con los materiales y coste amortizado.

Una vez vistos los costes materiales, trataremos los costes de personal. Para ello, se supondrá que este desarrollo lo ha llevado a cabo un Ingeniero Informático recién egresado, a tiempo parcial, con un salario bruto de 20.400,00€ anuales [2]. El desglose de estos costes para los cuatro meses de desarrollo quedaría de la siguiente forma (Tabla A.3) [3].

| Concepto                    | Coste            |
|-----------------------------|------------------|
| Salario anual en bruto      | 20.400,00€       |
| - IRPF (11,90%)             | 2.428,10€        |
| - Seguridad Social          | 1.295,40€        |
| - Salario anual neto        | 16.676,50 €      |
| <b>TOTAL (para 4 meses)</b> | <b>6.800,00€</b> |

Tabla A.3: Desglose del coste de personal.

Por lo tanto, si sumamos todos los costes, obtenemos que el coste total de la realización del proyecto asciende a **7.081,71€**.

En cuanto a los beneficios, no se dispone de beneficio alguno, pues la Universidad de Burgos proporcionará este software a sus alumnos de manera gratuita. Si seguimos con la simulación de un ambiente empresarial real y suponemos que la empresa desarrolladora vende este software a la Universidad de Burgos para que lo puedan distribuir entre sus alumnos, un buen precio sería de 9.500,00€, ya que proporcionaría a la empresa desarrolladora un margen de beneficio aproximado del 25,46% (2.418,70€).

## Viabilidad legal

Apartado dedicado al marco legal que envuelve el proceso de desarrollo del proyecto, su distribución, modificación y utilización. No es necesario atender al uso de licencias de terceros, pues todo el proyecto ha sido desarrollado con las licencias de Microsoft gratuitas para desarrolladores individuales o estudiantes y bajo una finalidad no comercial. En concreto, las licencias utilizadas más relevantes de Microsoft han sido:

- **Microsoft Visual Studio 2019 Community**. Al tratarse de un desarrollador individual, el uso de la licencia gratuita “Community” esta justificada. Si seguimos en el supuesto de una empresa, la aplicación se debería haber desarrollado bajo el estándar Open Source Initiative para seguir bajo esta licencia. De lo contrario (o si alguna otra condición no se cumple), se tendría que recurrir a **una licencia empresarial**, de pago.
- **Microsoft Office 365 Educational**. Distribuida de forma gratuita por la Universidad de Burgos a sus alumnos. En el supuesto de una empresa, se debería haber recurrido a uno de los **planes de pago**.

Cuando Francisco Crespo Diez [4] desarrolló la primera versión de esta aplicación en 2019, le asignó una licencia Creative Commons “Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional” [5]. Esto quiere decir que el usuario es libre de compartir, redistribuir, y adaptar la aplicación siempre que se le dé reconocimiento al autor, se siga una finalidad no comercial y se distribuyan las nuevas versiones bajo la misma licencia que el original. Esta licencia se ha seguido de forma estricta en todo momento y mantenido en esta nueva versión.

## *Apéndice B*

---

# Especificación de Requisitos

---

### B.1. Introducción

Apéndice destinado a hablar de los requisitos especificados del proyecto. Estos fueron definidos siguiendo el estándar IEEE 830-1998, el cual define que la Especificación de Requisitos Software (SRS) debe ser [6]:

- **Correcta.** Se considera correcta si todos los requisitos declarados se encuentran en el software.
- **Inequívoca.** Se considera inequívoca si todos los requisitos declarados tienen una única interpretación posible.
- **Completa.** Se considera completa si los requisitos están relacionados con la aplicación, se tienen en cuenta todos los posibles datos de entrada y todas las figuras y tablas adjuntas están referenciadas.
- **Consistente.** Se considera consistente si no contradice otras SRS de más alto nivel (sistema operativo, entorno de ejecución, etc).
- **Trazable en cuanto a importancia y/o estabilidad.**

- **Comprobable.** Se considera comprobable si todos los requisitos que contiene se pueden verificar.
- **Modificable.** Se considera modificable si se conservan su estructura y estilo pese a cualquier cambio en sus requisitos.
- **Identificable.** Se considera identificable si todos los requisitos declarados tienen un origen claro y si se facilitan referencias a los requisitos en futuros procesos de desarrollo.

## B.2. Objetivos generales

Como ya se comentó en la memoria, los objetivos generales de este proyecto han sido los siguientes:

1. Depuración de errores de código.
2. Implementación de nuevas funcionalidades sugeridas por usuarios de la aplicación.
3. Desarrollo de una nueva interfaz en forma de aplicación web mediante la cual los usuarios también puedan comunicarse con la placa.
4. Cambio del sistema gestor de bases de datos que utilizaba la aplicación original basado en Microsoft SQL Server 2017 Express por uno más ligero que se crea y destruye cada vez que se inicia la aplicación, dado que los datos que se almacenan en la base de datos no son demasiados ni presentan una estructura compleja.
5. Añadir a la sección de ayuda de la aplicación y al manual de usuario las nuevas funcionalidades implementadas.

## B.3. Catálogo de requisitos

### Aplicación local

En su día, Francisco Crespo Diez especificó los requisitos de la primera versión [4]. En este subpunto se tratarán únicamente los requisitos implementados en esta nueva versión.

Requisitos Funcionales:

- **RF1.** El usuario debe ser capaz de detectar el puerto en serie en uso.

- **RF2.** El usuario debe ser capaz de desplegar la interfaz web una vez establecida la conexión.

Requisitos no funcionales:

- **RNF1.** La base de datos debe encontrarse dentro de la propia aplicación y lanzarse a la vez que esta de forma transparente al usuario, sin requerimientos ni configuraciones previas.
- **RNF2.** Los archivos de la aplicación web deben encontrarse integrados dentro de los archivos de la aplicación local.
- **RNF3.** La aplicación debe ser capaz de lanzar mediante el servidor web IIS Express la interfaz web.
- **RNF4.** Se tomará registro de todas las acciones del usuario y sus consecuencias.
- **RNF5.** Los archivos de ayuda deberán ser actualizados con toda la información relativa a las funcionalidades implementadas.

## Aplicación web

En este subapartado se tratarán los requisitos funcionales de la nueva aplicación web desarrollada, integrada dentro de la aplicación local.

Requisitos funcionales:

- **RF1.** El usuario debe ser capaz de especificar el Connection String de la base de datos.
- **RF2.** El usuario debe ser capaz de visualizar los valores presentes en la base de datos.
  - **RF2.1.** El usuario debe se capaz de representar los valores de la base de datos en varias gráficas.
  - **RF2.2.** El usuario debe se capaz de filtrar qué valores son los representados en las gráficas
  - **RF2.3.** El usuario debe ser capaz de indicar el número de valores representados en las gráficas.
  - **RF2.4.** El usuario debe ser capaz de visualizar la totalidad de



valores almacenados en la base de datos.

- **RF3.** El usuario debe ser capaz de editar los valores de los parámetros de escritura del proyecto cargado.

Requisitos no funcionales:

- **RNF1.** La aplicación debe ser liviana y consumir pocos recursos.
- **RNF2.** La interfaz debe ser intuitiva.
- **RNF3.** La aplicación debe ser multilenguaje.
- **RNF4.** La aplicación debe mantener la ayuda accesible desde cualquier página que la conforme.
- **RNF5.** La aplicación deberá ser capaz de cargar el proyecto especificado desde la aplicación local.
- **RNF6.** La aplicación deberá ser capaz de establecer conexión con la base de datos automáticamente al lanzarse si el usuario ha especificado un Connection String en la aplicación local.
- **RNF7.** La aplicación deberá soportar varios temas visuales.
- **RNF8.** Se tomará registro de todas las acciones del usuario y sus consecuencias.

## B.4. Especificación de requisitos

A continuación, se relacionarán los requisitos definidos con los distintos casos de uso de la aplicación local (Figura B.16) y de la aplicación web (Figura B.17).



Figura B.16: Casos de uso de la aplicación local.

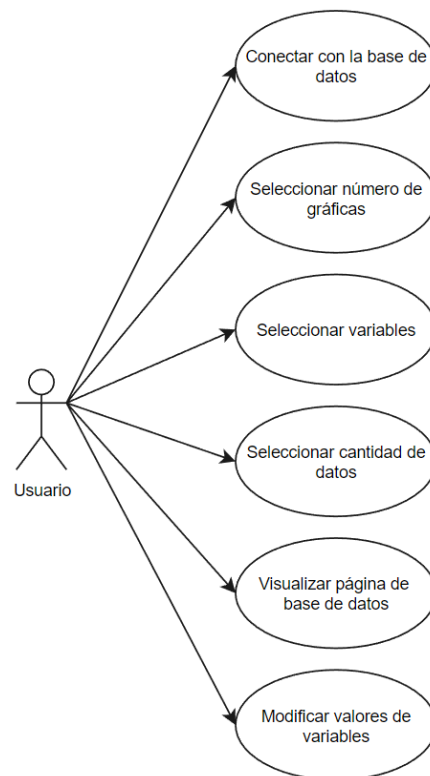


Figura B.17: Casos de uso de la aplicación web.

## Aplicación local

---

**Identificador:** CU1

**Requisito:** RF1

**Nombre:** Detectar puerto en uso

**Descripción:** Función que permite al usuario identificar cuál de todos los puertos serie del equipo es el que está recibiendo la comunicación con la planta.

**Autor:** Iván Cortés Aliende

**Importancia:** Media

---

**Responsabilidad:**

Leer la información del buffer de entrada de todos los puertos del sistema, con la intención de detectar el puerto al que está llegando la información enviada por la planta piloto y que los alumnos no pierdan el tiempo probando con todas las posibles opciones.

---

**Pre-condiciones:**

La aplicación debe estar en ejecución.

---

**Post-condiciones:**

Se muestra el puerto detectado por pantalla y se selecciona automáticamente en el desplegable.

Si no se encuentra ningún puerto en uso, después de 20 segundos para de buscar y notifica al usuario que no ha encontrado nada.

---

**Excepciones:** El sistema no tiene puertos en serie.

---

Tabla B.4: Aplicación local - Caso de uso 1.

---

**Identificador:** CU2**Requisito:** RF2**Nombre:** Desplegar aplicación web**Descripción:** El usuario puede lanzar desde la aplicación el servidor web IIS Express con la interfaz web del proyecto.**Autor:** Iván Cortés Aliende**Importancia:** Alta

---

**Responsabilidad:**

Modificar el fichero de configuración de IIS Express presente en el directorio de la aplicación web y crear un scrip .BAT que lance dicha aplicación con los criterios especificados por el usuario.

---

**Pre-condiciones:**

La aplicación debe estar en ejecución, se debe haber cargado un proyecto válido y se debe haber comenzado la comunicación con la planta piloto.

---

**Post-condiciones:**

Se lanza el servidor web con la aplicación web cargada, y se abre una pestaña del navegador web predeterminado del usuario en la dirección que él especificó.

La aplicación local se bloquea hasta que se termine la ejecución del servidor web.

---

**Excepciones:** IIS Express no encontrado en la ruta especificada.

---

Tabla B.5: Aplicación Local - Caso de uso 2.

## Aplicación Web

---

**Identificador:** CU1

**Requisito:** RF1

**Nombre:** Conectar con la base de datos

**Descripción:** El usuario podrá especificar la cadena de conexión con la base de datos, de manera que la aplicación web sepa dónde acudir a buscar los datos recibidos de la planta piloto y almacenados por la aplicación local

**Autor:** Iván Cortés Aliende

**Importancia:** Alta

---

**Responsabilidad:**

Utilizar la cadena especificada como ruta a la base de datos. Será utilizada por la clase DB\_services.cs para hacer las peticiones.

---

**Pre-condiciones:**

La aplicación debe estar en ejecución.

---

**Post-condiciones:**

Tras especificar la ruta de la base de datos, se prueba la conexión. Si es correcta, se informa al usuario mediante un indicador visual y se comienzan a mostrar los datos recibidos.

Si es incorrecta, se informa al usuario mediante un indicador visual.

---

**Excepciones:** Fallo al conectar con la base de datos mediante la dirección especificada.

---

Tabla B.6: Aplicación web – Caso de uso 1.

---

**Identificador:** CU2

**Requisito:** RF2.1

**Nombre:** Seleccionar número de gráficas

**Descripción:** El usuario podrá especificar el número de gráficas que se le mostrarán, pudiendo ser este parámetro un valor entre 1 y 5.

**Autor:** Iván Cortés Aliende

**Importancia:** Baja

---

**Responsabilidad:**

Si el valor es 1, solo mostrará la gráfica principal con los controles incrustados en la tabla de valores. Si es un valor mayor, se generarán y cargarán gráficas y controles suplementarios en la parte inferior de la página.

---

**Pre-condiciones:**

La aplicación debe estar en ejecución y la conexión con la base de datos establecida correctamente.

---

**Post-condiciones:**

Se mostrarán tantas gráficas y controles como se hayan indicado.

---

**Excepciones:** Fallo al cargar las gráficas extra.

---

Tabla B.7: Aplicación web – Caso de uso 2.

---

|   |
|---|
| <b>Identificador:</b> CU3   |
| <b>Requisito:</b> RF2.2   |
| <b>Nombre:</b> Seleccionar variables  |
| <b>Descripción:</b> El usuario podrá, mediante un sistema de check-boxes, indicar cuáles de todas las variables recibidas son las mostradas en las gráficas |
| <b>Autor:</b> Iván Cortés Aliende   |
| <b>Importancia:</b> Media   |

---

|   |
|---|
| <b>Responsabilidad:</b>   |
| Representar en la gráfica correspondiente al control de check-boxes únicamente las variables indicadas. |

---

|   |
|---|
| <b>Pre-condiciones:</b>   |
| La aplicación debe estar en ejecución y la conexión con la base de datos establecida correctamente. |

---

|   |
|---|
| <b>Post-condiciones:</b>  |
| Se cargará en el gráfico correspondiente únicamente los valores que coincidan con las variables marcadas. Si una variable ha sido desmarcada, la siguiente vez que se actualice el gráfico se borrarán los valores mostrados de esa variable y no se mostrarán más. |

---

|   |
|---|
| <b>Excepciones:</b> Ninguna variable seleccionada o variable no encontrada. |
|---|

---

Tabla B.8: Aplicación web – Caso de uso 3.

---

**Identificador:** CU4**Requisito:** RF2.3**Nombre:** Seleccionar cantidad de datos**Descripción:** El usuario podrá especificar cuántos datos de cada variable son mostrados en las gráficas.**Autor:** Iván Cortés Aliende**Importancia:** Media

---

**Responsabilidad:**

Representar en todas las gráficas la cantidad de datos que el usuario ha especificado, siendo el último valor representado el último valor presente en la base de datos.

---

**Pre-condiciones:**

La aplicación debe estar en ejecución y la conexión con la base de datos establecida correctamente.

---

**Post-condiciones:**

Se ajustarán todos los gráficos para mostrar únicamente el número de valores indicados por el usuario. Si el usuario especifica un valor no válido, se le notifica y no se realiza el cambio.

---

**Excepciones:** Fallo al conectar con la base de datos mediante la dirección especificada.

---

Tabla B.9: Aplicación web – Caso de uso 4.



---

**Identificador:** CU5**Requisito:** RF2.4**Nombre:** Visualizar páginas de base de datos**Descripción:** El usuario podrá visualizar la totalidad de valores almacenados en la base de datos relacionados con el proyecto cargado.**Autor:** Iván Cortés Aliende**Importancia:** Alta

---

**Responsabilidad:**

Redirigir al usuario a una página donde se aprecien todas las tablas de la base de datos que tengan relación con el proyecto.

---

**Pre-condiciones:**

La aplicación debe estar en ejecución y la conexión con la base de datos establecida correctamente.

---

**Post-condiciones:**

Se redirigirá al usuario a una página específica con las tablas de la base de datos.

---

**Excepciones:** Fallo al conectar con la base de datos mediante la dirección especificada.

---

Tabla B.10: Aplicación web – Caso de uso 5.

---

|   |
|---|
| <b>Identificador:</b> CU6   |
| <b>Requisito:</b> RF3   |
| <b>Nombre:</b> Modificar valores de variables   |
| <b>Descripción:</b> El usuario podrá editar los valores de las variables de escritura del proyecto cargado. |
| <b>Autor:</b> Iván Cortés Aliende   |
| <b>Importancia:</b> Alta  |

---

|   |
|---|
| <b>Responsabilidad:</b>   |
| Recoger la variable que el usuario quiere cambiar y el nuevo valor que esta va a tomar, para insertar esta información en la base de datos y que la aplicación local pueda leerla y mandar el mensaje por el puerto en serie. |

---

|   |
|---|
| <b>Pre-condiciones:</b>   |
| La aplicación debe estar en ejecución y la conexión con la base de datos establecida correctamente. |

---

|  |
|--|
| <b>Post-condiciones:</b>   |
| Se insertará la información en la base de datos y, posteriormente, se cambiará el valor de la variable. Si el valor introducido no es válido, se informará al usuario. |

---

|  |
|--|
| <b>Excepciones:</b> El cambio no tiene lugar en la planta. |
|--|

---

Tabla B.11: Aplicación web – Caso de uso 6.



## *Apéndice C*

---

# Especificación de diseño

---

### C.1. Introducción

En este apéndice se tratarán los aspectos relativos al diseño de tratamiento de datos, diseño de procesos y diseño arquitectónico seguido durante el desarrollo del proceso.

### C.2. Diseño de datos

En cuanto a la gestión de datos, estos son almacenados en una base de datos. Dicha base de datos tiene dos tablas por cada proyecto cargado en el sistema:

- La primera se encarga de almacenar los valores de todas las variables del proyecto desde que se inicia la comunicación en serie, junto con el indicador de tiempo y un campo de identificación.
- La segunda se encarga de almacenar las solicitudes enviadas desde la interfaz web de cambio de un valor de una determinada variable. Por tanto, cada entrada se compone del indicador de tiempo, el campo de identificación, el nombre de la variable a cambiar y el nuevo valor

solicitado.

Además, si la aplicación web es lanzada desde la aplicación local también se utiliza un fichero XML para el intercambio de cierta información entre ambas aplicaciones:

- El Connection String de la base de datos, para que el usuario no lo tenga que meter a mano.
- La localización de los ficheros de la aplicación web (sus propios ficheros), para la creación del directorio “Logs” y sus correspondientes ficheros.

### C.3. Diseño procedimental

Subapartado destinado a mostrar los procesos más relevantes de la aplicación. En la figura C.18 se puede observar el diagrama de secuencia con las interacciones de los distintos componentes del sistema para una sencilla tarea de almacenamiento y representación de los datos en serie recibidos de la placa FRDM-K64F.

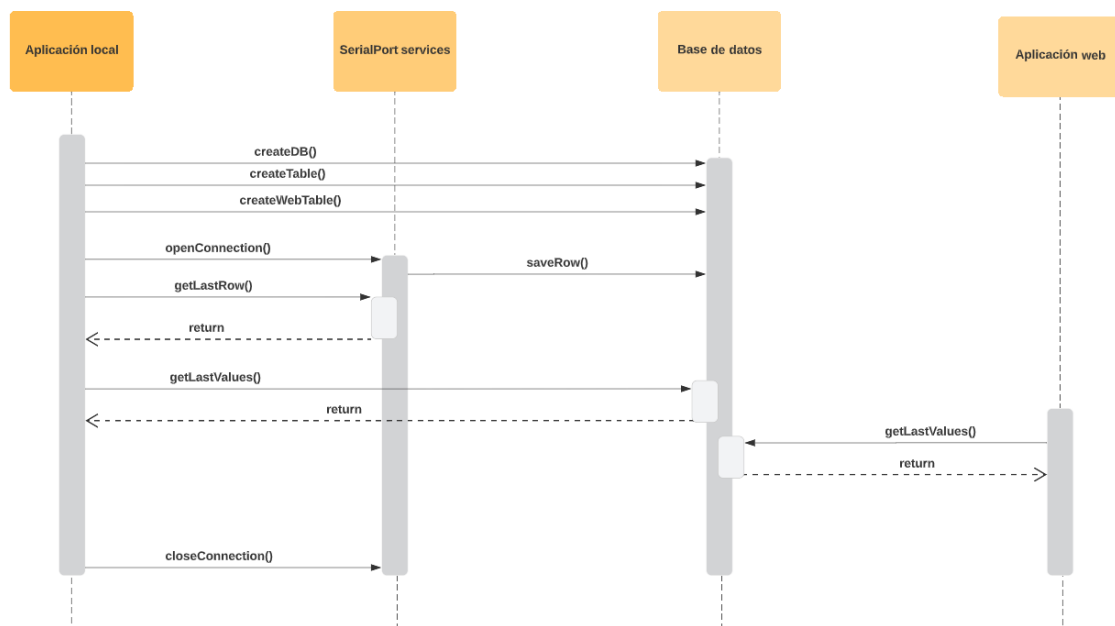


Figura C.18: Diagrama de secuencia de los distintos componentes de la aplicación.

### C.4. Diseño arquitectónico

Subapartado destinado a explicar los patrones arquitectónicos que siguen las aplicaciones.

## Aplicación local

Como ya se explicó en la memoria, Francisco Crespo Diez [4] en su primera versión implementó el patrón arquitectónico Modelo - Vista - Controlador, el cual busca separar la aplicación en tres capas:

- **Modelo:** capa que trata los datos.
- **Vista:** interfaz de usuario.
- **Controlador:** intermediario entre el modelo y la vista.

Dicho patrón arquitectónico fue respetado durante esta versión de la aplicación local, para seguir un proceso de desarrollo coherente. Identificamos sus siguientes componentes:

- **Modelo:** la base de datos.
- **Vista:** formularios con los que el usuario interactúa.
- **Controlador:** la parte lógica de la aplicación, encargada de establecer las comunicaciones y la llamada de métodos.

## Aplicación web

Esta aplicación fue desarrollada en ASP.NET, bajo el modelo ASP.NET Web Forms. Se optó por este modelo frente al ASP.NET MVC (Modelo - Vista - Controlador) porque, según la documentación oficial [7], se trata de un modelo controlado por eventos desarrollado para la creación de aplicaciones web centradas en la utilización de datos y el manejo mediante una interfaz de usuario, descripción que cuadra con la naturaleza de la aplicación a la perfección. Al tratarse de una aplicación de intercambio de información con una base de datos, primordialmente centrada en la experiencia de usuario y únicamente conformada por dos formularios web, se consideró esta opción como la más acertada frente al resto.

Aun no estando centrado en el patrón Modelo - Vista - Controlador, sí que es cierto que este patrón se respeta hasta el punto de poder identificar cada una de las partes de forma clara:

- **Modelo:** La base de datos, común a la aplicación local.
- **Vista:** Los ficheros HTML y ASP de la aplicación web, donde se

encuentran definidas todas las interfaces de usuario.

- **Controlador:** Las clases .CS que hay detrás de cada página, encargadas de la gestión de eventos de usuario y llamadas a métodos.

En la Figura C.19 se puede observar un sencillo diagrama de clases de la aplicación web con las relaciones más importantes del sistema.

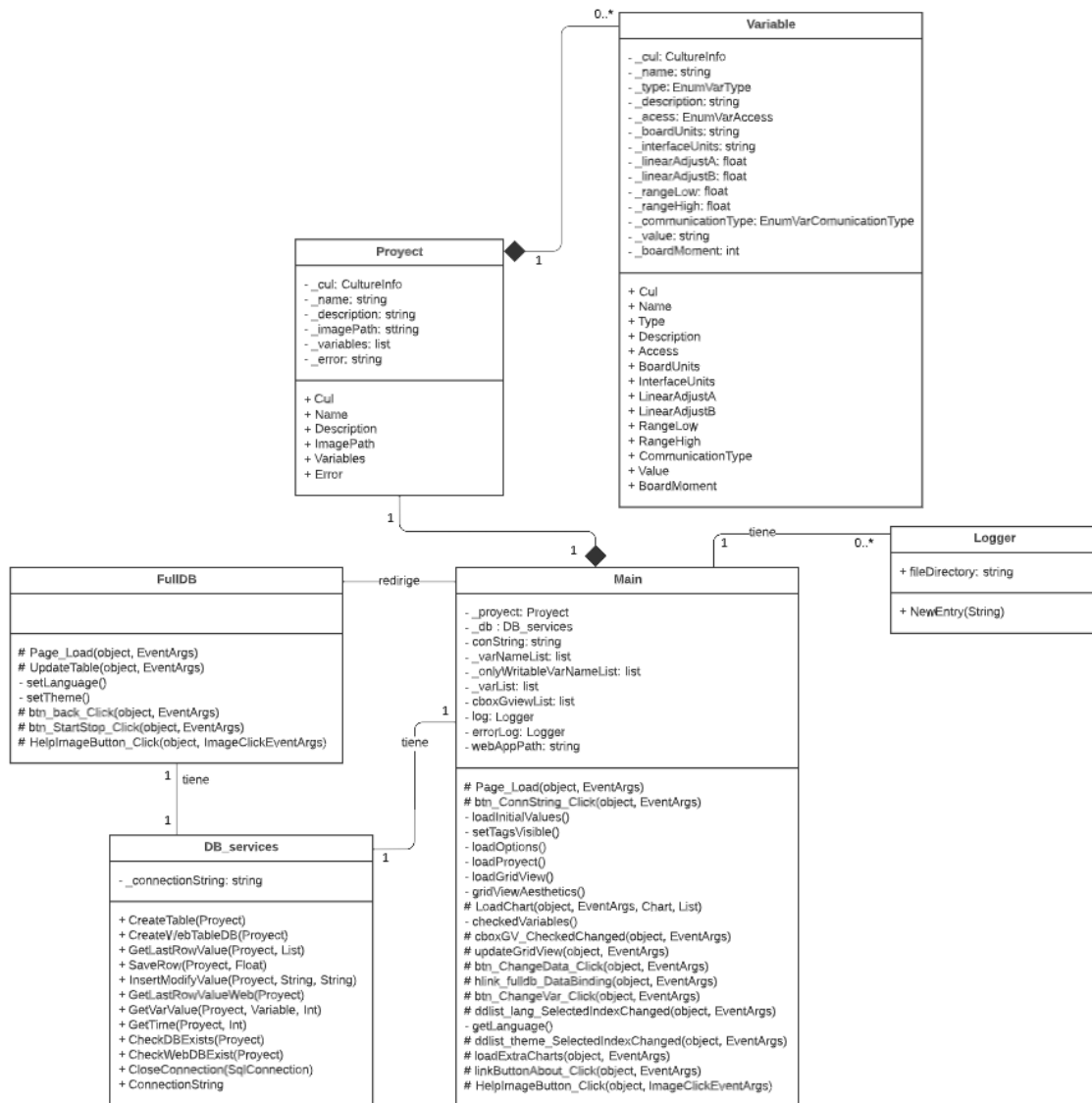


Figura C.19: Diagrama de clases UML de la Aplicación web.

## *Apéndice D*

---

# Documentación técnica de programación

---

### D.1. Introducción

Apéndice destinado a documentar los aspectos relacionados con cómo ha sido programada la aplicación, cómo se distribuyen los directorios, cómo descargarse y compilar el repositorio, el proceso de pruebas, etc. Todo esto con el objetivo de facilitar el trabajo a futuros desarrolladores que quieran crear su propia versión.

### D.2. Estructura de directorios

Se explicará la estructura de directorios del repositorio en [GitHub](#) del proyecto.

- `/`: Directorio raíz del repositorio. En él se encuentran todos los directorios del repositorio y los archivos de licencia y readme.
- `/Memoria`: Directorio donde se encuentra la memoria y los anexos del proyecto.



- **/PlantaPiloto:** Directorio donde se encuentran las dos aplicaciones.
- **/PlantaPiloto/PlantaPiloto:** Directorio de la aplicación local.
- **/PlantaPiloto/PlantaPiloto/ApplicationData:** Directorio donde se encuentran ficheros necesarios para su lanzamiento desde la aplicación. Aquí se encuentran los ficheros de ayuda y el manual de usuario.
- **/PlantaPiloto/PlantaPiloto/Classes:** Directorio donde se encuentran las clases definidas no relativas a los formularios.
- **/PlantaPiloto/PlantaPiloto/Enum:** Directorio donde se encuentran las enumeraciones.
- **/PlantaPiloto/PlantaPiloto/Forms:** Directorio donde se encuentran los formularios de la aplicación.
- **/PlantaPiloto/PlantaPiloto/Properties:** Directorio donde se encuentran los ficheros de propiedades.
- **/PlantaPiloto/PlantaPiloto/Resources:** Directorio donde se encuentran los archivos de idioma del proyecto.
- **/PlantaPiloto/PlantaPiloto/Services:** Directorio con las clases de servicio de comunicación con la base de datos y el puerto en serie.
- **/PlantaPiloto/PlantaPiloto/images:** Directorio con las imágenes usadas en el proyecto.
- **/WebPlantaPiloto:** Directorio de la aplicación web. Aparte de los directorios, también encontramos en esta carpeta los archivos de las dos clases de formulario de la aplicación y archivos autogenerados por Microsoft Visual Studio 2019.
- **/WebPlantaPiloto/App\_Start:** Directorio con los archivos relativos al lanzamiento de la aplicación.
- **/WebPlantaPiloto/Classes:** Directorio donde se encuentran las clases no relativas a los formularios web. Aquí se encuentran las clases abstractas de idioma.
- **/WebPlantaPiloto/Content:** Directorio donde se encuentran los archivos de estilo de la aplicación web.

- **/WebPlantaPiloto/html**: Directorio donde se encuentran las páginas HTML de ayuda y “Acerca de”.
- **/WebPlantaPiloto/html/images**: Directorio donde se encuentran las imágenes utilizadas por la aplicación.
- **/WebPlantaPiloto/Properties**: Directorio donde se encuentran los ficheros de propiedades.
- **/WebPlantaPiloto/Scripts**: Directorio donde se encuentran los scripts de Javascript utilizados por la aplicación.
- **/WebPlantaPiloto/fonts**: Directorio donde se encuentran las fuentes utilizadas por la aplicación.

### D.3. Manual del programador

En este apartado se tratará cómo un programador puede descargar el proyecto y desarrollar su propia versión.

#### Descarga del código

Como ya se ha comentado, el código se encuentra disponible en un **repositorio de GitHub público**. Por lo tanto, se puede descargar mediante la aplicación web (Figura D.20) o mediante la herramienta Git y el comando *git clone https://github.com/ica1006/TFG*.

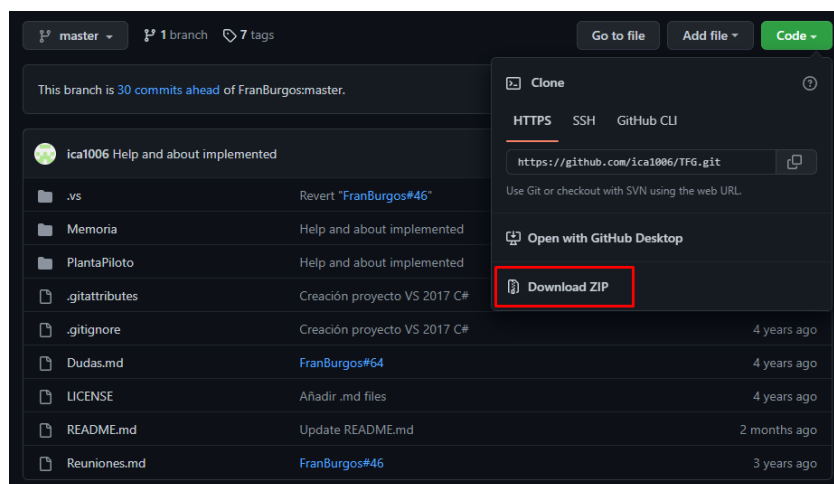


Figura D.20: Descarga del repositorio mediante el entorno web.

Si se quiere hacer una nueva versión de la aplicación, lo mejor sería realizar

un *fork* en GitHub para que el control de versiones se pudiera seguir de la manera más coherente posible.

## Apertura del proyecto en el entorno de desarrollo

El entorno de desarrollo recomendado es Microsoft Visual Studio (Figura D.21), ya que es desarrollado por Microsoft, la misma empresa que desarrolla .NET Framework, el entorno de ejecución para el que el proyecto está pensado.

Además, también es necesario tener instalado en el equipo .NET Framework en su versión 4.6.1 o posterior.

Una vez con el entorno de desarrollo instalado, se podrá importar el proyecto en Microsoft Visual Studio abriendo el fichero de solución “PlantaPiloto.sln”, disponible en el directorio /PlantaPiloto.

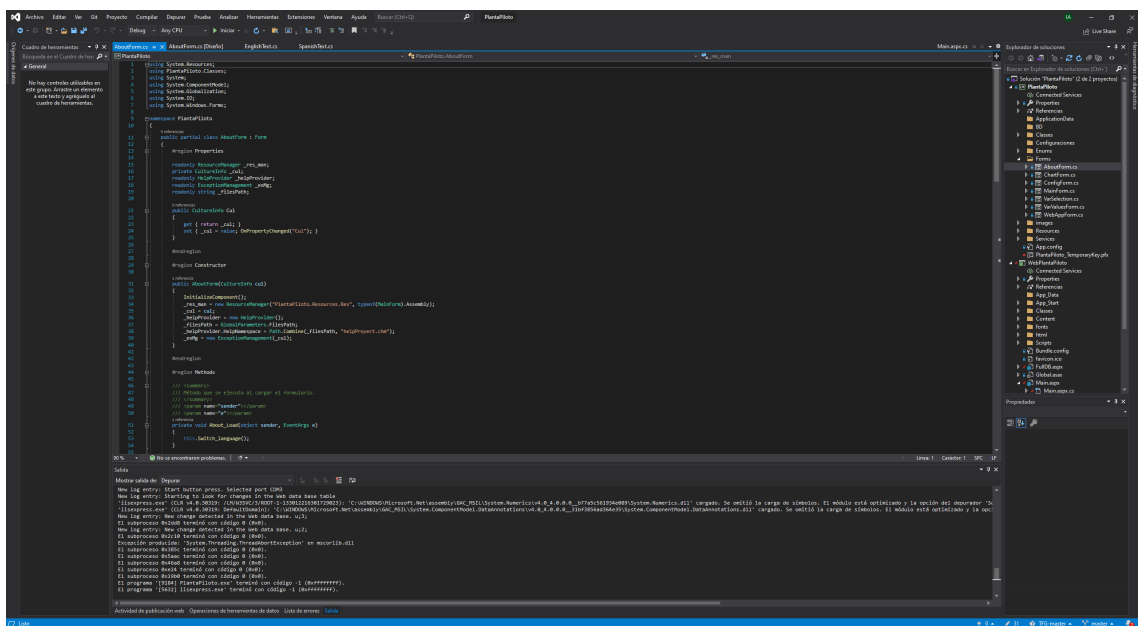


Figura D.21: Proyecto abierto en el IDE Microsoft Visual Studio 2019.

## D.4. Compilación, instalación y ejecución del proyecto

Microsoft Visual Studio permite desde su interfaz el debug y compilación de cada aplicación por separado. Para ello, bastará con dar click derecho en una de las dos aplicaciones y seleccionar la opción deseada (Figura D.22).

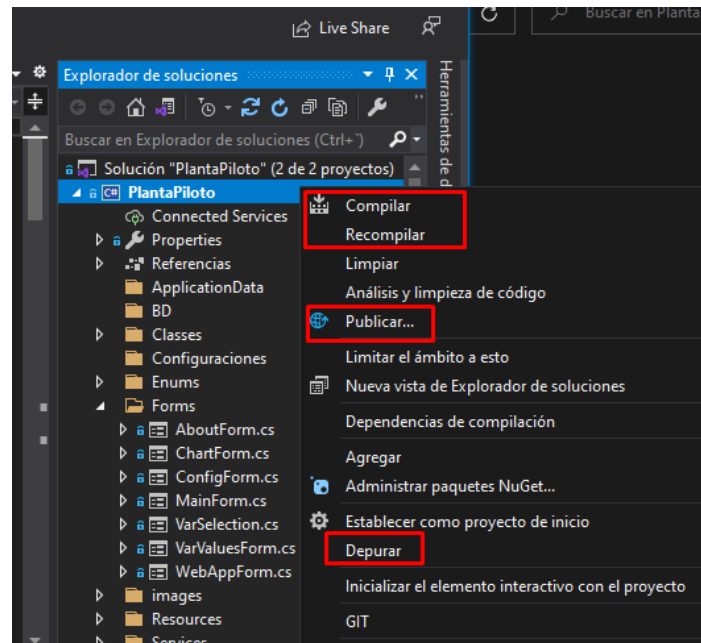


Figura D.22: Menú contextual con las opciones relativas a la compilación y debug señaladas.

Una vez comenzada la depuración del proyecto, en el entorno de desarrollo aparecerán los controles de la depuración (Figura D.23).

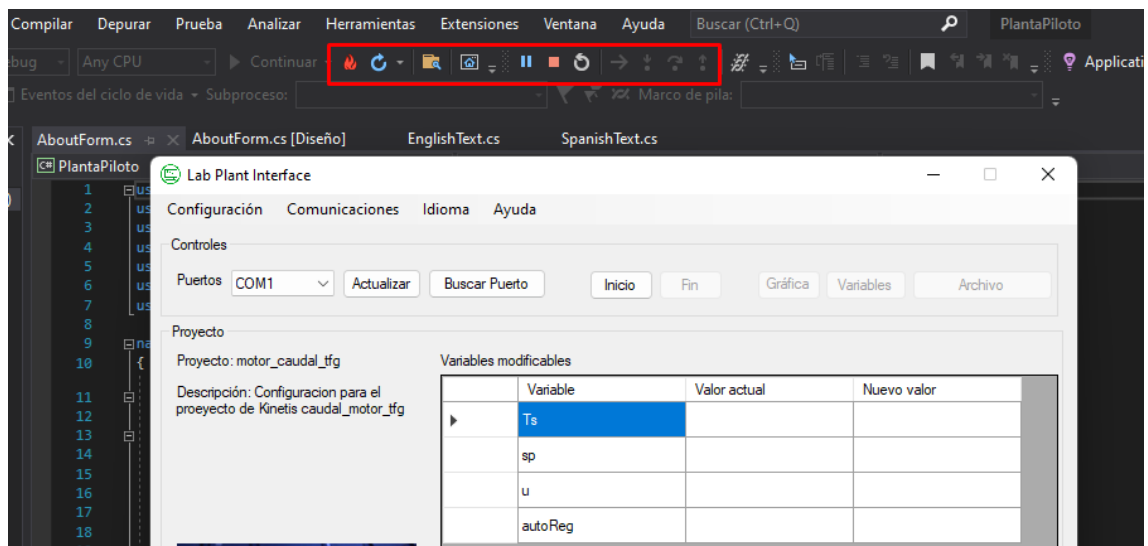


Figura D.23: Microsoft Visual Studio 2019 depurando la aplicación local. Controles de depuración señalados.

## D.5. Pruebas del sistema

Para las nuevas funcionalidades implementadas en la aplicación local las

pruebas fueron realizadas manualmente, como en su día hizo Francisco Crespo Diez. Esto es debido a que su éxito no depende de un gran número de variables, lo que hacía que probar su efectividad no lleva mucho tiempo.

Sin embargo, para el desarrollo de la aplicación web sí que se desarrolló un sencillo plan de pruebas, ya que se tenían que probar una serie de posibles datos de entrada del usuario que podían conllevar a errores en la aplicación. Los casos de prueba han sido los siguientes.

### **Caso de prueba #1**

Se ha probado a introducir un Connection String no válido, para probar si el sistema es capaz de detectarlo e imprimir el mensaje de error, así como el correspondiente indicador de la interfaz de usuario.

### **Caso de prueba #2**

Se ha probado a introducir un Connection String válido, para probar si el sistema era capaz de conectarse adecuadamente a la base de datos y empezar a leer datos, así como cambiar el correspondiente indicador de la interfaz usuario.

### **Caso de prueba #3**

Se ha probado a introducir en el campo relativo al cambio del valor de una variable del proyecto caracteres no válidos (todo carácter que no sea numérico, punto o coma), para ver si el sistema era capaz de detectarlo, mostrar el mensaje de error correspondiente y no enviar el mensaje a la base de datos.

### **Caso de prueba #4**

Se ha probado a introducir en el campo relativo al cambio del valor de una variable del proyecto únicamente caracteres numéricos. Para probar si el sistema era capaz de detectarlo, no mostrar ningún mensaje de error y enviar el mensaje a la base de datos.

### **Caso de prueba #5**

Se ha probado a introducir en el campo relativo al cambio del valor de una variable del proyecto caracteres numéricos en combinación con el carácter punto y el carácter coma, conformando números decimales. En ambos casos el sistema debe ser capaz de transformar la cadena en un número decimal y enviar el

mensaje a la base de datos, sin mostrar ningún mensaje de error.

### **Caso de prueba #6**

Se ha probado a introducir en el campo relativo al cambio del valor de una variable del proyecto la combinación de carácter no válidos (todo aquel carácter que no sea numérico, punto o coma) con caracteres válidos (números, punto y coma), para confirmar que el sistema es capaz de no enviar el mensaje a la base de datos y mostrar el mensaje de error en la interfaz de usuario.

### **Caso de prueba #7**

Se ha probado a introducir en el campo relativo a la cantidad de datos mostrados en los gráficos combinaciones de caracteres no validos (todo aquel carácter que no sea un dígito), para comprobar que el sistema no realizaba el cambio y mostraba el mensaje de error correspondiente en la interfaz.

### **Caso de prueba #8**

Se ha probado a introducir en el campo relativo a la cantidad de datos mostrados en los gráficos únicamente caracteres validos (dígitos), para comprobar que el sistema realizaba el cambio y no mostraba ningún mensaje de error en la interfaz.

### **Caso de prueba #9**

Se ha probado a introducir en el campo relativo a la cantidad de datos mostrados en los gráficos combinaciones de caracteres no validos (todo aquel carácter que no sea un dígito) y caracteres válidos (dígitos), para comprobar que el sistema no realizaba el cambio y mostraba el mensaje de error correspondiente en la interfaz.

### **Otros**

Todos estos casos de prueba fueron probados una vez de forma manual mientras eran grabados con el software **Katalon Recorder**, el cual permite repetir acciones ya realizadas de manera automática. El resto de las pruebas, menos repetitivas y más difíciles de automatizar, fueron realizadas de manera completamente manual (redirección de páginas, cambio de idioma, cambio de tema visual, etc).

Dejando a un lado las pruebas de entrada del usuario, para las pruebas de calidad del código fue utilizada la herramienta **SonarQube**, capaz de detectar bugs, vulnerabilidades, código repetido, etc. La versión final de ambas aplicaciones sacó una calificación de “A” en SonarQube (Figuras D.24 y D.25).

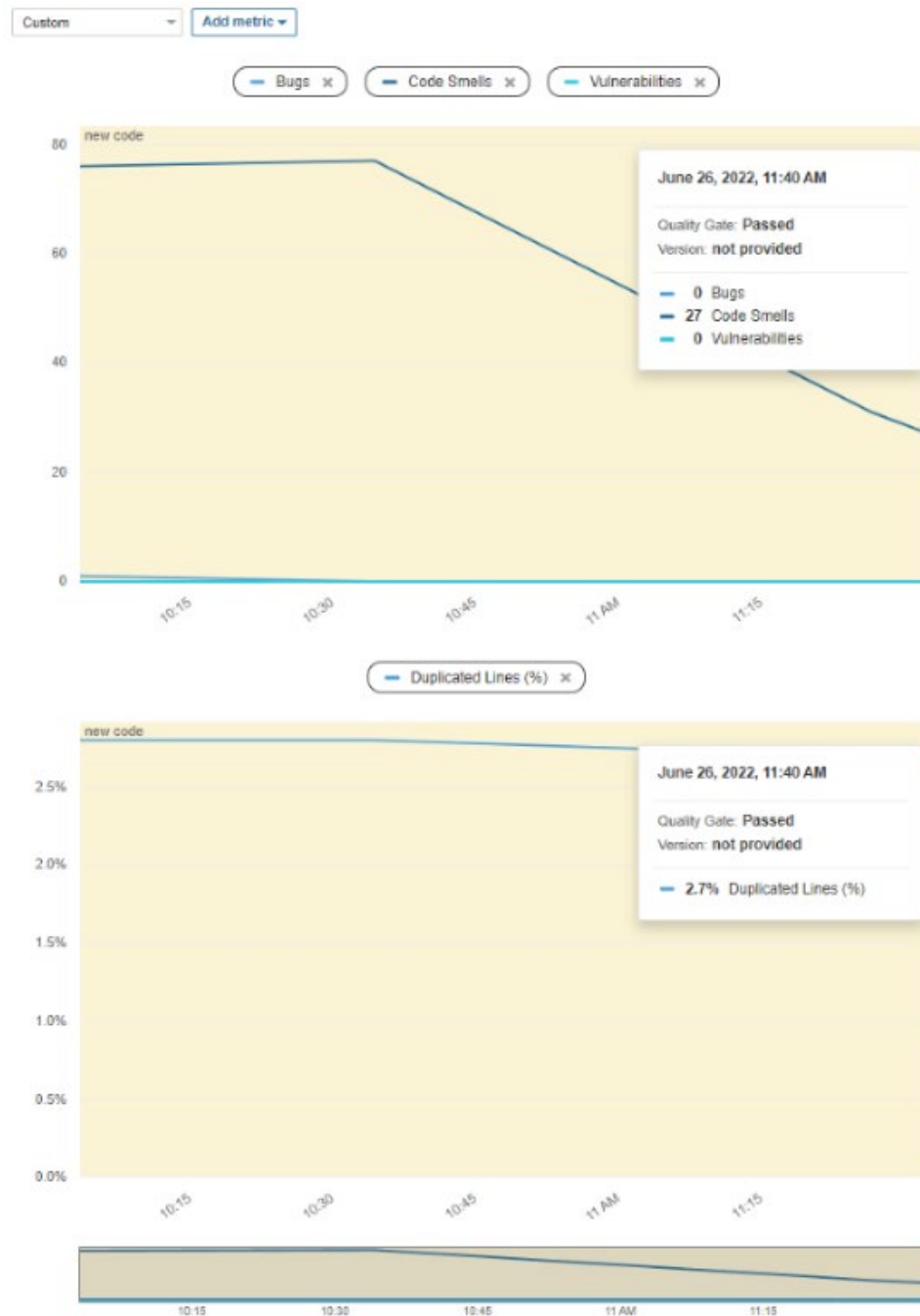


Figura D.24: Gráfico de actividad de la aplicación local en SonarQube.

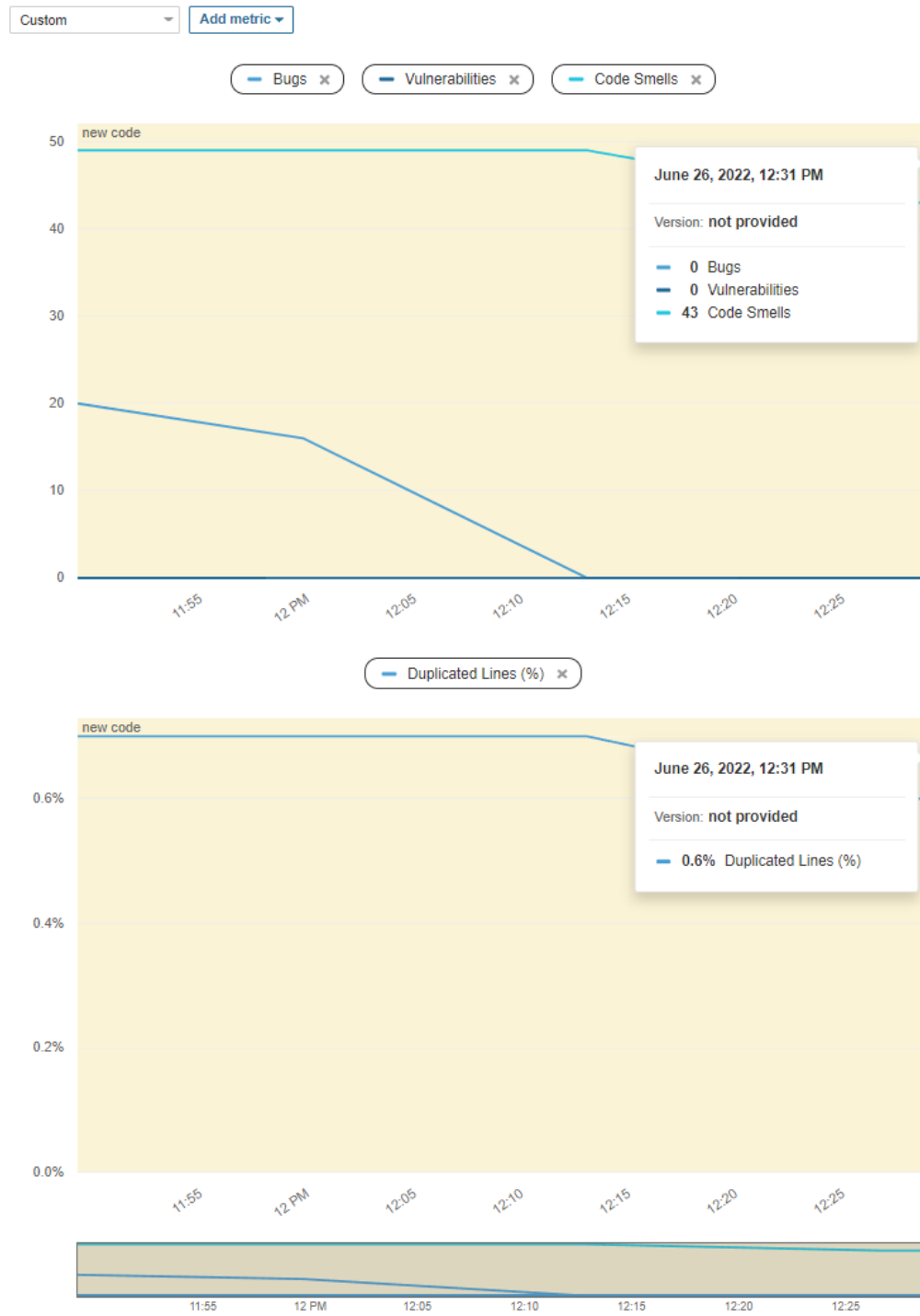


Figura D.25: Gráfico de actividad de la aplicación web en SonarQube





## *Apéndice E*

---

# Documentación de usuario

---

### E.1. Introducción

Apéndice destinado a tratar los temas relacionado con la parte del usuario. Se tratarán los temas relativos a requisitos, instalación de la aplicación y el manual de uso del usuario.

### E.2. Requisitos de usuarios

Aparte de disponer de una planta piloto con las especificaciones tratadas en la memoria y un ordenador conectado a ella, se deben cumplir los siguientes requisitos:

- **Sistema operativo Windows.** Esto es debido a que la aplicación fue desarrollada para ser utilizada en este sistema operativo, dado que es el presente en los ordenadores del laboratorio. Aún así, la aplicación podría llegar a funcionar en otros sistemas operativos haciendo algunos cambios en las dependencias y en el código de la aplicación, pero esta opción no ha sido probada.
- **.NET Framework.** Es la base de la aplicación, el marco de ejecución

para el que fue desarrollada. Sin él, la aplicación no podría llegar a funcionar. Es necesaria una versión igual o superior a la 4.6.1.

- **P&E Driver.** Como ya comentamos en la memoria, es el controlador necesario para emular un puerto UART con una conexión en serie en un puerto USB del equipo, permitiendo la comunicación entre la planta piloto y el ordenador. Si nuestro sistema tuviera un puerto UART ya disponible en la placa base y los drivers correspondientes instalados, este requisito no sería necesario, pero hace ya bastantes años que este puerto dejó de incluirse en los ordenadores de uso cotidiano y quedó obsoleto.
- **IIS Express.** Servidor web para el que se desarrolló el despliegue automático de la aplicación web. Si se quiere utilizar la funcionalidad de la aplicación web, será necesario tenerlo instalado en el equipo.
- **La aplicación en su última release.** Esta puede ser obtenida directamente del [apartado releases del repositorio de la aplicación](#). A día de hoy, la versión más actualizada se encuentra recogida en el fichero “ReleaseV2.0.zip”.

### E.3. Instalación

No existe un proceso de instalación más allá del propio de las dependencias ya citadas, ya que la aplicación es autocontenida, es decir, el resto de los ficheros y librerías necesarias ya se incluyen dentro de la carpeta de la *release*. Esto se decidió hacer de esta manera para simplificar la tarea de los profesores y alumnos a la hora de instalarlo en los equipos del laboratorio.

Para su ejecución, bastará con entrar dentro de la carpeta de la *release* (una vez descomprimido el fichero) y ejecutar el archivo “PlantaPiloto.exe”.

Hay que tener en cuenta que el número de las funcionalidades de la aplicación sin la planta piloto conectada son prácticamente ninguna.

### E.4. Manual del usuario

Subapartado destinado a redactar un sencillo manual que el usuario pueda emplear para entender cómo utilizar la aplicación.

#### Aplicación local

En su día, Francisco Crespo Diez [4] ya desarrolló un manual de usuario explicando las funcionalidades básicas de la aplicación. Por tanto, en esta ocasión yo me centraré en las nuevas funcionalidades implementadas para esta versión.

### Búsqueda de puerto en uso

En la anterior versión, los alumnos tenían que lidiar con buscar manualmente cuál de todos los puertos del equipo es el que estaba recibiendo la comunicación en serie desde la planta piloto, pues los ordenadores del laboratorio tenían gran cantidad de puertos y el nombre de estos no era muy descriptivo.

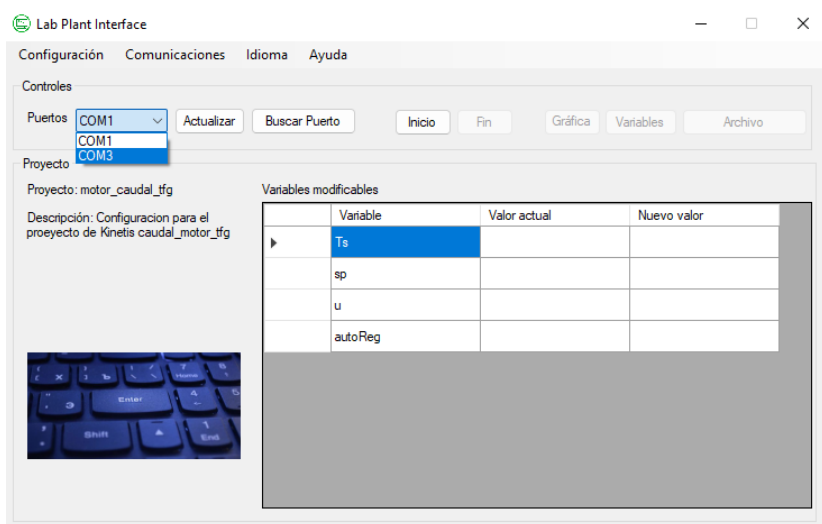


Figura E.26: Búsqueda de puerto en uso.

En el formulario principal, será suficiente con dar click en el botón “Buscar puerto” (Figura E.26). Si se encuentra un puerto en uso, este será indicado mediante una ventana emergente, y será seleccionado en el desplegable de puertos de manera automática. En el caso de que no se encuentre ningún puerto en uso después de veinte segundos, se indicará al usuario mediante una ventana emergente que la búsqueda no tuvo éxito.

### Despliegue de la aplicación web

Para desplegar la aplicación web desde la aplicación local, bastará con seleccionar la opción “Web Server” en el apartado de “Comunicaciones” una vez la comunicación con la planta piloto haya comenzado. Eso nos mostrará un formulario con diferentes apartados (Figura E.27)

The screenshot shows a 'WebApp' deployment configuration window. It is divided into two main sections: 'IIS Express' and 'Aplicación Web'. The 'IIS Express' section contains three fields: 'Ruta IISExpress' (C:\Program Files (x86)\IIS Express), 'Dirección IP' (localhost), and 'Puerto' (1337). The 'Aplicación Web' section contains four fields: 'Ruta Aplicación Web' (C:\Users\Iván\OneDrive - Universidad de Burgos\4o Año\TFG-master\PlantaPiloto\PlantaPiloto\bin\Debug\WebAppRelease), 'Ruta del archivo de proyecto' (C:\Users\Iván\OneDrive - Universidad de Burgos\4o Año\TFG-master\PlantaPiloto\PlantaPiloto\bin\Debug\Configuraciones\motor\_caudal\_tfg3.txt), 'String de conexión de la base de datos' (Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Iván\OneDrive - Universidad de Burgos\4o Año\TFG-master\PlantaPiloto\PlantaPiloto), and a 'Lanzar' button. Red numbers 1 through 8 are placed next to the fields to indicate specific configuration points.

Figura E.27: Formulario de despliegue de la aplicación web.

Los campos señalados con números rojos hacen referencia a los campos que los usuarios pueden modificar, los cuales hacen referencia a los siguientes apartados:

1. Ruta donde se encuentra presente IIS Express (el servidor web).
2. Dirección IP en la que se lanzará el servidor.
3. Puerto en el que se lanzará el servidor.
4. Ruta donde se encuentran presentes los archivos de la aplicación web.
5. Ruta donde se encuentra presente el fichero de texto del proyecto.
6. Cadena de conexión de la base de datos del proyecto
7. Botón de acceso a la ayuda de este formulario.
8. Botón para lanzar el servidor web IIS Express.

## Aplicación web

### Página principal

Una vez se haya lanzado el servidor web, se nos abrirá una ventana del navegador web por defecto accediendo a la dirección que hayamos especificado que se lance el servidor web, junto con la extensión */Main*. El valor por defecto de la dirección es *http://localhost:1337/Main*. En esta dirección encontraremos la página principal (Figura E.28).

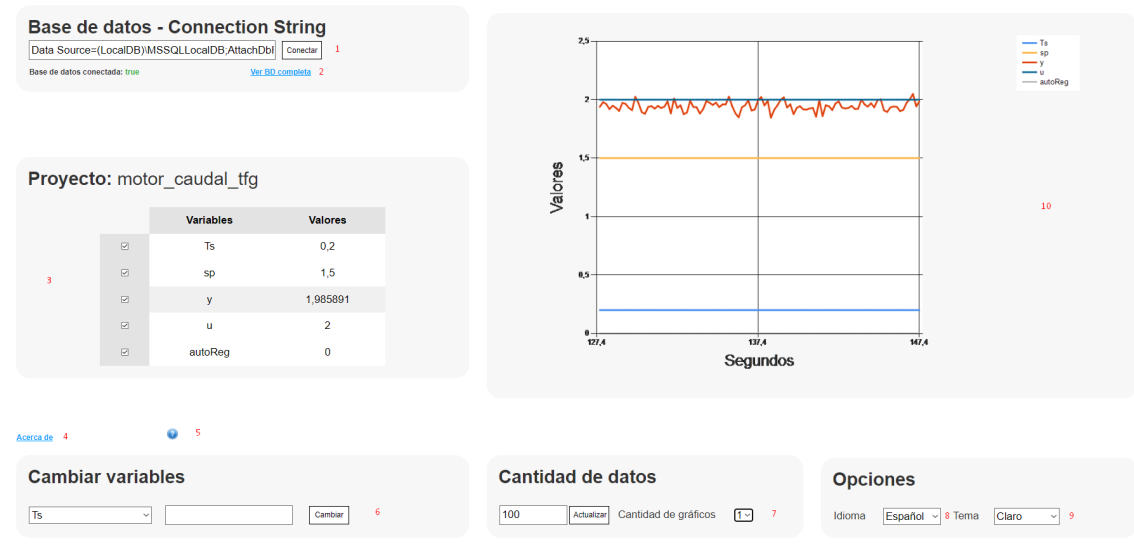


Figura E.28: Página principal de la aplicación web.

A continuación, se explican cada uno de los apartados señalados con números rojos:

1. Connection String para la conexión con la base de datos.
2. Enlace para ver todas las tablas del proyecto de la base de datos.
3. Variables del proyecto en tiempo real. Los controles permiten indicar cuales se mostrarán en el gráfico de la derecha (10).
4. Enlace para ver la información "Acerca de".
5. Botón para mostrar la página de ayuda.
6. Campo para el cambio de variables. En el desplegable se seleccionará la variable a editar y en el campo de la derecha el nuevo valor que va a tomar.
7. Cantidad de datos a mostrar en los gráficos y número de gráficos a mostrar.
8. Menú de cambio de idioma.
9. Menú de cambio del tema visual.
10. Apartado donde se muestra el gráfico principal con las variables marcadas en la tabla de la izquierda.

## Página con la información de la base de datos

Ubicada en la dirección */FullDB*. En esta página se muestran toda la información recogida en la base de datos, de tal forma que los usuarios puedan de manera cómoda realizar un estudio de cómo han ido cambiando los valores de la planta piloto en el tiempo (Figura E.29).

The screenshot shows a web interface for a database. At the top, there are two buttons: 'Volver' (labeled 1) and 'Parar Auto-Actualizar' (labeled 4). Below these are two tables. The first table, 'Base de Datos de valores' (labeled 2), has columns: Id, Time, Ts, sp, y, u, and autoReg. It contains 19 rows of data. The second table, 'Base de Datos de cambios' (labeled 5), has columns: Id, Time, Variable, and NuevoValor. It contains 4 rows of data. A small blue icon with the number 3 is located above the 'Parar Auto-Actualizar' button.

| Id | Time | Ts  | sp  | y        | u | autoReg |
|----|------|-----|-----|----------|---|---------|
| 1  | 0,2  | 0,2 | 1,5 | 1,946061 | 2 | 0       |
| 2  | 0,4  | 0,2 | 1,5 | 1,929494 | 2 | 0       |
| 3  | 0,6  | 0,2 | 1,5 | 2,050044 | 2 | 0       |
| 4  | 0,8  | 0,2 | 1,5 | 1,892383 | 2 | 0       |
| 5  | 1    | 0,2 | 1,5 | 2,069883 | 2 | 0       |
| 6  | 1,2  | 0,2 | 1,5 | 1,967562 | 2 | 0       |
| 7  | 1,4  | 0,2 | 1,5 | 1,96575  | 2 | 0       |
| 8  | 1,6  | 0,2 | 1,5 | 1,938054 | 2 | 0       |
| 9  | 1,8  | 0,2 | 1,5 | 1,881707 | 2 | 0       |
| 10 | 2    | 0,2 | 1,5 | 1,918466 | 2 | 0       |
| 11 | 2,2  | 0,2 | 1,5 | 2,000897 | 2 | 0       |
| 12 | 2,4  | 0,2 | 1,5 | 1,974008 | 2 | 0       |
| 13 | 2,6  | 0,2 | 1,5 | 1,957995 | 2 | 0       |
| 14 | 2,8  | 0,2 | 1,5 | 1,91328  | 2 | 0       |
| 15 | 3    | 0,2 | 1,5 | 1,994049 | 2 | 0       |
| 16 | 3,2  | 0,2 | 1,5 | 1,945306 | 2 | 0       |
| 17 | 3,4  | 0,2 | 1,5 | 1,976626 | 2 | 0       |
| 18 | 3,6  | 0,2 | 1,5 | 2,051403 | 2 | 0       |
| 19 | 3,8  | 0,2 | 1,5 | 1,930854 | 2 | 0       |

| Id | Time  | Variable | NuevoValor |
|----|-------|----------|------------|
| 1  | 35,6  | u        | 3          |
| 2  | 50,2  | u        | 2          |
| 3  | 111,6 | Ts       | 0.3        |
| 4  | 121   | Ts       | 0.2        |

Figura E.29: Página de información de la base de datos.

Los apartados señalados en rojo hacen referencia a los siguientes apartados:

1. Botón para volver a la página principal.
2. Tabla con todos los datos de todas las variables recibidas por el puerto en serie hasta el momento.
3. Botón para mostrar este menú de ayuda.
4. Botón para parar / reanudar la actualización automática de los datos en la página.
5. Tabla con todos los cambios enviados desde la aplicación web.

## Página Acerca de

En esta página se muestra información básica de la autoría de la aplicación web (Figura E.30).



Figura E.30: Información presente en la página Acerca De.





---

## Bibliografía

---

- [1] David J. Anderson. *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010.
- [2] Jobted, “*Sueldo del Ingeniero Informático en España*”, 2022. [Internet; descargado 01-julio-2022].
- [3] CincoDías, “Calculadora IRPF”, 2022. [Internet, descargado 01-julio-2022].
- [4] F.C. Diez, “*Desarrollo de una interfaz para planta piloto*”, tech.rep., Grado en Ingeniería Informática - Universidad de Burgos, 2019.
- [5] CreativeCommons, “Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0)”, 2013. [Internet; descargado 01-julio-2022].
- [6] IEEE Computer Society. “*IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications*”. IEEE, 1998
- [7] Microsoft, “*Información general de ASP.NET*”, 2022. [Internet; descargado 19-junio-2022].