



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Desarrollo de una interfaz
para planta piloto**



Presentado por Francisco Crespo Diez
en Universidad de Burgos — 10 de febrero
de 2019

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



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



Dr. Daniel Sarabia Ortiz, profesor del departamento de Ingeniería Electromecánica, área de Ingeniería de Sistemas y Automática, y Dr. Alejandro Merino Gómez, profesor del departamento de Ingeniería Electromecánica, área de Ingeniería de Sistemas y Automática.

Exponen:

Que el alumno D. Francisco Crespo Diez, con DNI 71296830G, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado "Desarrollo de una interfaz para planta piloto".

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, 10 de febrero de 2019

Vº. Bº. del Tutor:

D. Daniel Sarabia Ortiz

Vº. Bº. del co-tutor:

D. Alejandro Merino Gómez

Resumen

El objetivo de este proyecto es el desarrollo de una interfaz que actúe como intermediario entre el usuario y una placa Freescale FRDMK64F, con la finalidad de monitorizar variables del proceso que esté contenido en la placa mencionada, así como para actuar sobre dichas variables.

La placa Freescale FRDMK64F estará a su vez conectada a una planta piloto en la cual se podrá tener acceso al control de la temperatura y del caudal de aire que ocurre en su interior.

Descriptores

Interfaz hombre-máquina, monitorización, NXP, Freescale, FRDMK64F, .NET, Windows Forms, C#, SCADA.

Abstract

The goal of this project is the development of an interface that acts as an intermediary between the user and a Freescale FRDMK64F board, in order to monitor process variables contained in the board mentioned, as well as to act on these variables.

The Freescale FRDMK64F board will, in turn, be connected to a pilot plant in which it will be possible to have access to the control of the temperature and the flow of air that occurs inside.

Keywords

Man machine interface, monitoring, NXP, Freescale, FRDMK64F, .NET, Windows Forms, C#, SCADA.

Índice general

Índice general	III
Índice de figuras	v
Índice de tablas	vii
Introducción	1
Objetivos del proyecto	5
2.1. Objetivos generales	5
2.2. Objetivos técnicos	5
2.3. Objetivos personales	6
Conceptos teóricos	7
3.1. .NET Framework	7
3.2. Base de datos	9
3.3. Comunicación Serie	9
3.4. Planta piloto	11
3.5. SCADA	17
3.6. Entorno de desarrollo integrado (IDE)	18
Técnicas y herramientas	19
4.1. Metodología	19
4.2. Patrones de diseño	20
4.3. Control de versiones	20
4.4. Hosting de repositorio	20
4.5. Gestión del proyecto	21

4.6. Entorno de desarrollo integrado (IDE)	21
4.7. Comunicación	22
4.8. Documentación	23
4.9. Test	24
4.10. Otras herramientas	25
4.11. Base de datos	26
4.12. Editores	27
Aspectos relevantes del desarrollo del proyecto	29
5.1. Idea e inicio del proyecto	29
5.2. Formación	30
5.3. Desarrollo de la interfaz y la internacionalización	30
5.4. Desarrollo de la comunicación a través del puerto serie	31
5.5. Desarrollo de la comunicación con la base de datos	32
5.6. Control de excepciones	33
5.7. Pruebas	33
Descripción de la herramienta	35
6.1. Descripción	35
6.2. Funcionalidad	36
6.3. Características	45
Trabajos relacionados	47
7.1. Primera versión	47
7.2. Segunda versión	47
Conclusiones y líneas de trabajo futuras	49
8.1. Conclusiones	49
8.2. Líneas de trabajo futuras	50
Bibliografía	53

Índice de figuras

3.1. Contexto .NET	8
3.2. Ejemplo de puerto de comunicación paralelo.	10
3.3. Ejemplo de puerto de comunicación serie.	10
3.4. Ejemplo de comunicación UART.	11
3.5. Serie de plantas piloto en el laboratorio de ingeniería de sistemas y automática de la UBU.	11
3.6. Esquema del contenido de la planta.	12
3.7. Repartición de los componentes de la planta piloto.	13
3.8. Hardware NXP K64 (FRDM-K64F).	14
3.9. Planta Piloto.	15
3.10. Esquema del funcionamiento del microcontrolador.	15
3.11. Esquema acondicionamiento de señales.	16
3.12. Esquema de la conexión entre la planta y el ordenador.	17
3.13. Ejemplo esquema implementación SCADA.	18
4.14. ZenHub Overview	21
4.15. Ventana de desarrollo del proyecto en Visual Studio 2017	22
4.16. CHM File	24
4.17. Codacy Dashboard	25
6.18. Ventana principal de la aplicación.	37
6.19. Ventana para la creación de un nuevo proyecto.	38
6.20. Ventana para la modificación de un proyecto cargado.	39
6.21. Ventana para la selección de variables.	40
6.22. Ventana de graficado de valores de variables.	41
6.23. Ventana de dibujado de gráficas.	42
6.24. Contenido del archivo de guardado de variables.	42
6.25. Ventana de ayuda.	43
6.26. Ventana Acerca de.	44

6.27. PDF - Manual de usuario.	45
--	----

Índice de tablas

Introducción

Desde el inicio del estudio de ingenierías, siempre ha resultado más sencilla la comprensión por parte de los alumnos de conceptos teóricos si estos son vistos en la práctica. Sin embargo, no siempre es posible tener acceso a estas facilidades debido a no disponer de un software dirigido al estudiante, entre otras desventajas.

Hasta el momento los alumnos de la Universidad de Burgos han tenido que trabajar con las placas NXP FRDM-K64F de una manera arcaica, con un software de terceros, a través de una consola de comandos que enviaba mensajes directamente a la placa, sin poder optar a una visualización apropiada de los datos.

Partiendo de este principio y teniendo en cuenta los últimas tecnologías implantados en la Universidad de Burgos, el objetivo de este proyecto es el desarrollo de una interfaz hombre-máquina capaz de comunicarse con una placa NXP FRDM-K64F en un entorno amigable, permitiendo al alumno una mayor libertad para trabajar con dicho dispositivo.

1.1. Estructura de la memoria

La memoria sigue la siguiente estructura:

- **Introducción:** breve descripción del problema a resolver y la solución propuesta. Estructura de la memoria y listado de materiales adjuntos.
- **Objetivos del proyecto:** exposición de los objetivos que persigue el proyecto.
- **Conceptos teóricos:** en este apartado se describen los conceptos que se necesitan saber antes de poder comprender todo el proyecto.

- **Técnicas y herramientas:** donde se indican todas las técnicas, bibliotecas, lenguajes, y otras herramientas que se han utilizado durante el proyecto.
- **Aspectos relevantes del desarrollo:** descripción del desarrollo general que ha llevado el proyecto.
- **Trabajos relacionados:** se nombran otros trabajos similares o relacionados con este proyecto.
- **Conclusiones y líneas de trabajo futuras:** conclusiones obtenidas tras la realización del proyecto y posibilidades de mejora o expansión de la solución aportada.

Junto con la memoria se proporcionan los siguientes anexos:

- **Plan del proyecto software:** planificación temporal y estudio de viabilidad del proyecto.
- **Especificación de requisitos del software:** se describe la fase de análisis; los objetivos generales, el catálogo de requisitos del sistema y la especificación de requisitos funcionales y no funcionales.
- **Especificación de diseño:** se describe la fase de diseño; el ámbito del software, el diseño de datos, el diseño procedimental y el diseño arquitectónico.
- **Manual del programador:** recoge los aspectos más relevantes relacionados con el código fuente (estructura, compilación, instalación, ejecución, pruebas, etc.).
- **Manual de usuario:** guía de usuario para el correcto manejo de la aplicación.

1.2. Materiales adjuntos

Los materiales entregados son:

- Aplicación de escritorio para Windows.
- Dataset de vídeos de pruebas en [SharePoint](#).
- Memoria en formato PDF.

- Anexos en formato PDF.

Además, los siguientes recursos están disponibles en Internet:

- Repositorio del proyecto en [GitHub](#).

Objetivos del proyecto

A continuación se detallan los diversos objetivos que han motivado la realización del proyecto.

2.1. Objetivos generales

- Crear una interfaz hombre-máquina amigable que permita la monitoreo y modificación de los datos gestionados por una placa NXP.
- Ayudar a los nuevos estudiantes de la Universidad de Burgos en su comprensión del funcionamiento de una planta piloto y los sistemas de control necesarios para controlarla. Fundamentalmente orientado para las prácticas de las asignaturas de control en el grado de Electrónica Industrial y Automática, el grado de ingeniería Informática y el máster de Ingeniería Industrial.
- Aportar información extra a los datos recibidos por la placa que ayude a la comprensión de los cambios experimentados en la planta piloto.
- Dar la posibilidad de almacenar los datos recibidos de la planta piloto de una manera concisa, estructurada y de fácil acceso.

2.2. Objetivos técnicos

- Desarrollar una aplicación en .NET Framework - Windows Forms para entornos Windows.

- Utilizar Git como herramienta de control de versiones distribuido junto con GitHub.
- Aplicar la metodología ágil Scrum en el desarrollo del software.
- Utilizar la herramienta ZenHub como herramienta de gestión de proyectos.
- Documentar a través de la herramienta L^AT_EX.
- Implementar la comunicación serie entre la aplicación y la la placa para conseguir una comunicación online con la misma y el proceso
- Distribuir la aplicación resultante para entornos Windows.

2.3. Objetivos personales

- Abarcar el máximo número de conocimientos vistos en el grado, como por ejemplo programación orientada a objetos, bases de datos, ingeniería del software, comunicaciones, etc.
- Realizar una aportación a la modernización de los recursos de la Universidad de Burgos.
- Profundizar en el desarrollo de aplicaciones .NET y en la utilización del entorno de desarrollo Visual Studio.
- Trabajar con placas Kinetis de NXP.
- Explorar herramientas y metodologías de vanguardia en el mercado laboral.

Conceptos teóricos

En este capítulo se expondrán los conceptos teóricos que se van a ver reflejados en el proyecto, permitiendo al lector tener una base de conocimiento en la que apoyarse para la comprensión del proyecto.

3.1. .NET Framework

.NET Framework es un entorno de ejecución para Windows que administra aplicaciones cuyo destino es .NET Framework, proporcionando diversos servicios a las aplicaciones en ejecución. El diseño de .NET Framework está enfocado a cumplir los siguientes objetivos:

- Proporcionar un entorno de ejecución de código que promueve la ejecución segura del mismo y que reduzca todo lo posible los conflictos de versiones y la implementación de software.
- Proporcionar un entorno coherente de programación orientada a objetos.
- Fomentar la integración del código de .NET con otros tipos de código basando la comunicación en estándares del sector.
- Ofrecer al programador coherencia entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en Web.

.NET Framework tiene dos componentes principales: la biblioteca de clases de .NET Framework y Common Language Runtime (CLR). En la figura 3.1 se puede apreciar la relación de Common Language Runtime y la biblioteca de clases con el sistema en su conjunto y las aplicaciones. Así

mismo, la ilustración representa el funcionamiento del código administrado dentro de una arquitectura mayor [1].

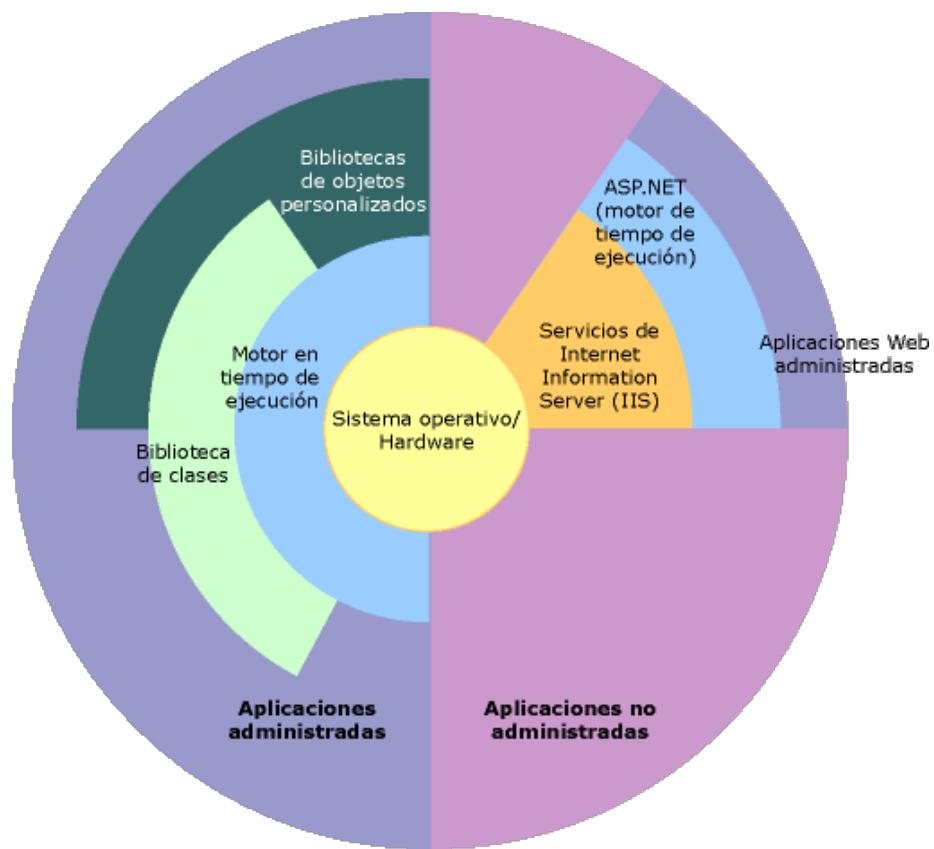


Figura 3.1: Contexto .NET.

Biblioteca de clases de .Net Framework

La biblioteca de clases de .NET Framework es una colección de tipos reutilizables que se integran estrechamente con Common Language Runtime. Al ser una biblioteca de clases orientada a objetos, proporciona tipos de los que su propio código administrado deriva funciones. Esto hace que los tipos de .NET Framework sean fáciles de usar, reduciendo así el tiempo asociado con el aprendizaje de las nuevas características de .NET Framework. Cabe añadir que los componentes de terceros se integran fácilmente con las clases de .NET Framework.

Common Language Runtime (CLR)

Common Language Runtime gestiona la memoria, la ejecución de código, la ejecución de subprocesos, la comprobación de la seguridad del código, la compilación y el resto servicios del sistema. Estas características son intrínsecas del código administrado que se ejecuta en Common Language Runtime.

3.2. Base de datos

Una base datos es una aplicación independiente que almacena una colección de datos pertenecientes a un mismo contexto organizados por registros, archivos y campos, permitiendo así un rápido acceso a dichos datos [2].

Base de datos relacional

Una base de datos es relacional cuando cumple con el modelo relacional, el cual hace referencia a la relación que existe entre las distintas entidades o tablas de la base. En este modelo, el lugar y la forma en que se almacenan los datos no es relevante (a diferencia de otros modelos como el jerárquico y el de red).

SQL

SQL (Structured Query Language, Lenguaje de Consulta Estructurado) es un lenguaje de alto nivel que, gracias al álgebra y a los cálculos relacionales, permite la inserción, actualización, consulta y borrado de datos, así como la creación y modificación de esquemas y el control de acceso a los datos dentro de una base de datos relacional. Es el lenguaje más habitual para construir las consultas a bases de datos relacionales [3].

3.3. Comunicación Serie

A diferencia de la comunicación en paralelo que se muestra en la Figura 3.2 en la que todos los bits se envían al mismo tiempo, la comunicación serie o secuencial, mostrada en la Figura 3.3 consiste en el envío de datos de un bit a la vez, de manera secuencial, sobre un canal de comunicación o un bus. A pesar de que a la misma frecuencia la comunicación paralela obtiene un

mayor rendimiento, la transmisión en serie requiere de un menor número de líneas de trasmisión, siendo muy sencillo su uso [4].

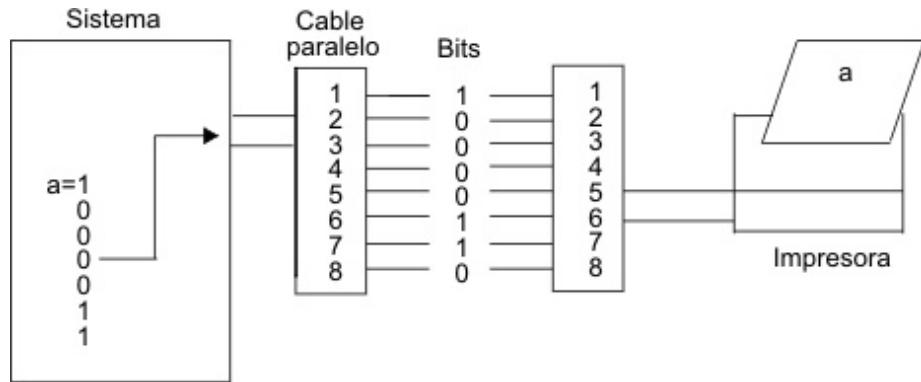


Figura 3.2: Ejemplo de puerto de comunicación paralelo.

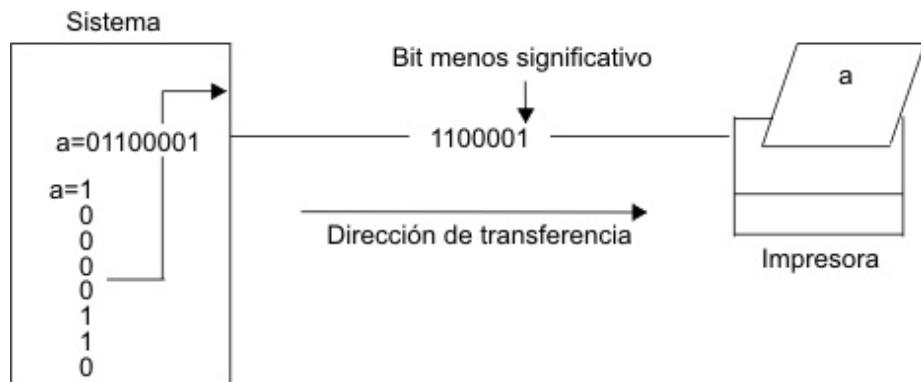


Figura 3.3: Ejemplo de puerto de comunicación serie.

Dentro del amplio abanico de tipos de comunicación que ofrece la comunicación serie, en este proyecto se utilizará el tipo UART. El UART (Universal Asynchronous Receiver/Transmitter) es el dispositivo alojado en la placa que se encarga de controlar los puertos y dispositivos serie. Su principal función de este dispositivo consiste, como se muestra en la Figura 3.4, en manejar de forma asíncrona las interrupciones de los dispositivos conectados al puerto serie y convertir los datos en formato paralelo para que el bus del sistema pueda trabajar con ellos, y viceversa [5].

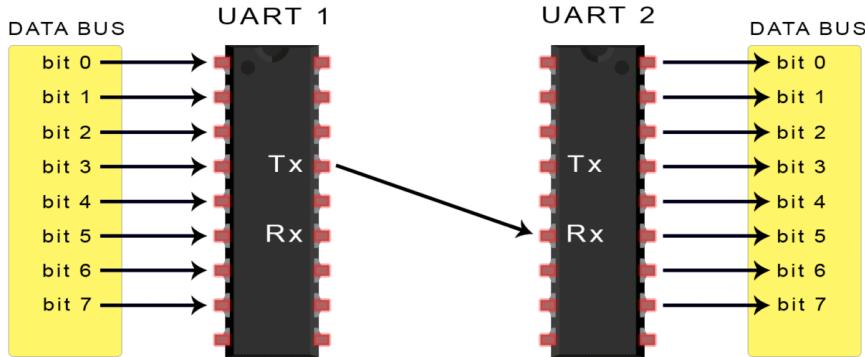


Figura 3.4: Ejemplo de comunicación UART.

3.4. Planta piloto

Para llevar a cabo este proyecto, se parte de una planta piloto ya creada, mostrada en la Figura 3.5, la cual requiere de una interfaz hombre-máquina para mejorar la interacción en la transmisión de datos, tanto para procesarlos como para modificarlos.



Figura 3.5: Serie de plantas piloto en el laboratorio de ingeniería de sistemas y automática de la UBU.

El prototipo de esta planta, que se muestra en la Figura 3.7 fue diseñado por Rubén Zambrana Rodríguez en su TFG de electrónica [6], y fue mejorado a su versión actual por María Isabel Revilla Izquierdo, egresada en Ingeniería

Electrónica Industrial y Automática, durante su trabajo de fin de grado[7], ambos bajo la tutela del Dr. Daniel Sarabia Ortiz.

La planta piloto se compone, como se muestra en la Figura 3.6, de:

- Una planta.

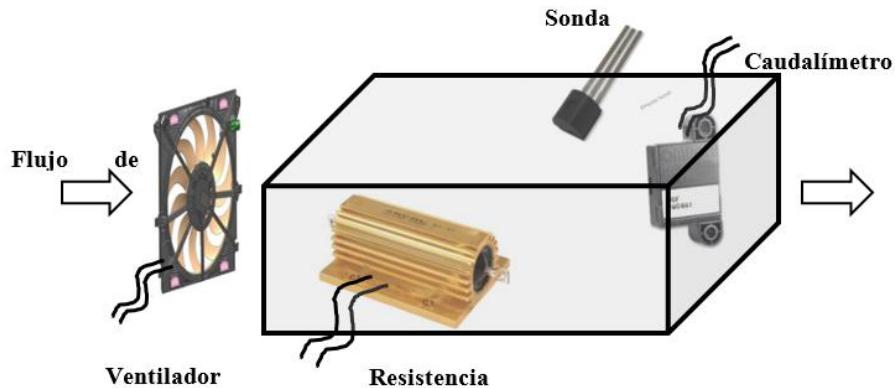


Figura 3.6: Esquema del contenido de la planta.

- Un equipo de alimentación.
- Varias etapas de acondicionamiento de las señales recibidas y enviadas a la planta, con la idea de poder suministrar la potencia requerida y ajustar los rangos de las señales a los requerimientos de la placa.
- Un sistema empotrado o placa donde se implementa el sistema de control y permite la comunicación con la instrumentación de la planta piloto, sensores de caudal, temperatura y actuadores, tensión del ventilador y tensión de la resistencia (ofrecido por la placa FRDM-K64F).

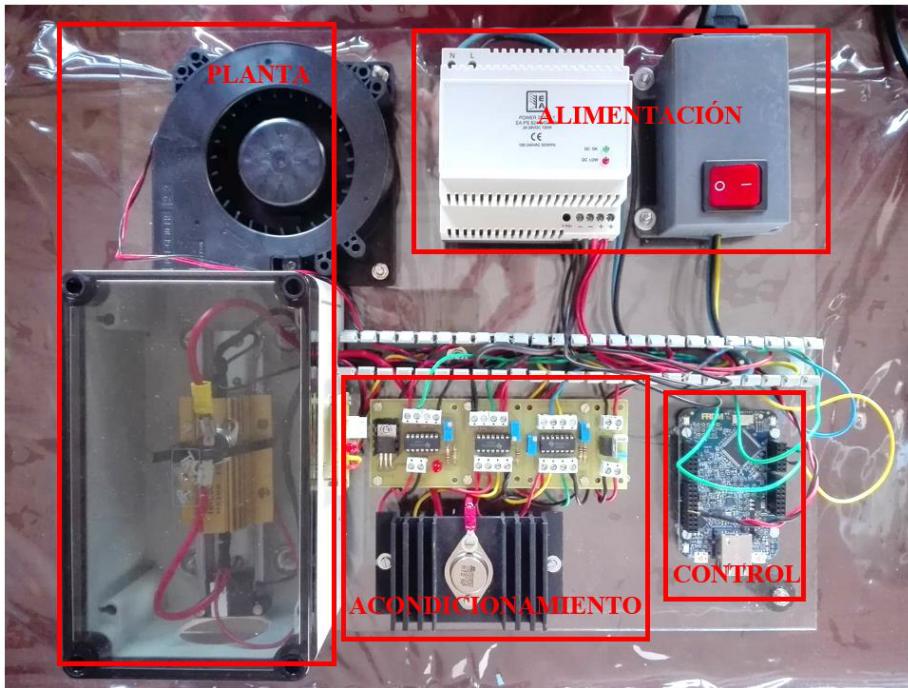


Figura 3.7: Repartición de los componentes de la planta piloto.

NXP FRDM-K64F

La plataforma de desarrollo NXP es un conjunto de herramientas software y hardware para el desarrollo y la evaluación de prototipos de aplicaciones basadas en microcontroladores. Concretamente, el hardware NXP K64 (FRDM-K64F), mostrado en la Figura 3.8, es el que va a ser usado para el proyecto [8].

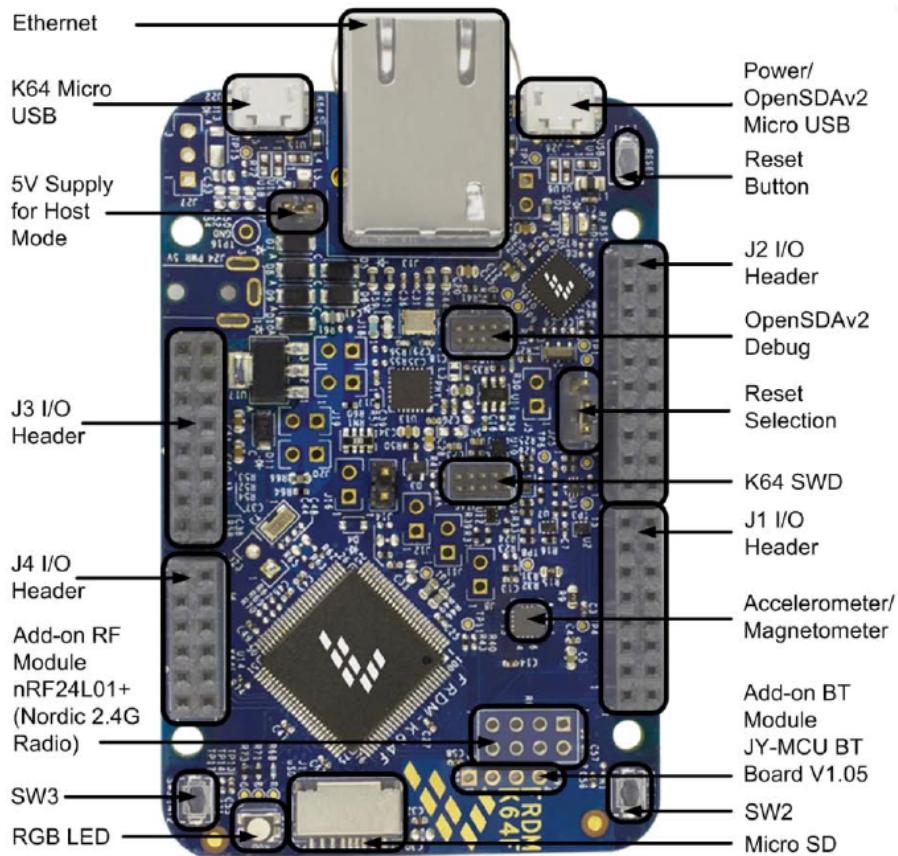


Figura 3.8: Hardware NXP K64 (FRDM-K64F).

La placa ya está programada con un sistema de control, centrándonos de esta manera en la programación de la interfaz hombre máquina. La comunicación de la aplicación con la placa se realiza mediante el puerto USB, que emula una comunicación serie con el driver “Power/OpenSDAv2 Micro USB”.

Arquitectura de la planta

La planta mencionada anteriormente y que se muestra en la Figura 3.9, utilizará un sistema de control digital implementado en el microcontrolador NXP FRDM-K64F con el fin de controlar la velocidad del aire.

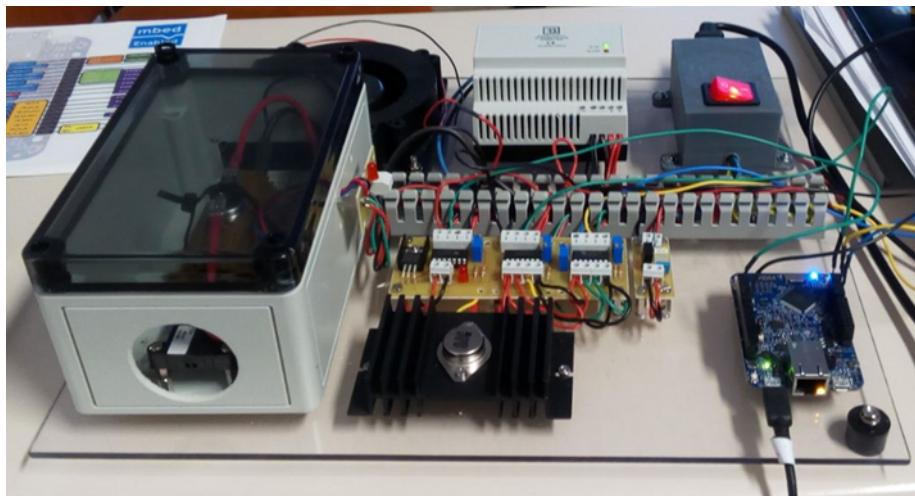


Figura 3.9: Planta Piloto.

Como se puede ver en el esquema que muestra la Figura 3.10, el microcontrolador es el encargado de recibir los datos a través de referencia o setpoint, enviarlos a través la variable manipulada al proceso, el cual le devuelve la variable controlada. Para todo esto, el microcontrolador debe realizar la conversión de datos de analógico a digital y viceversa.

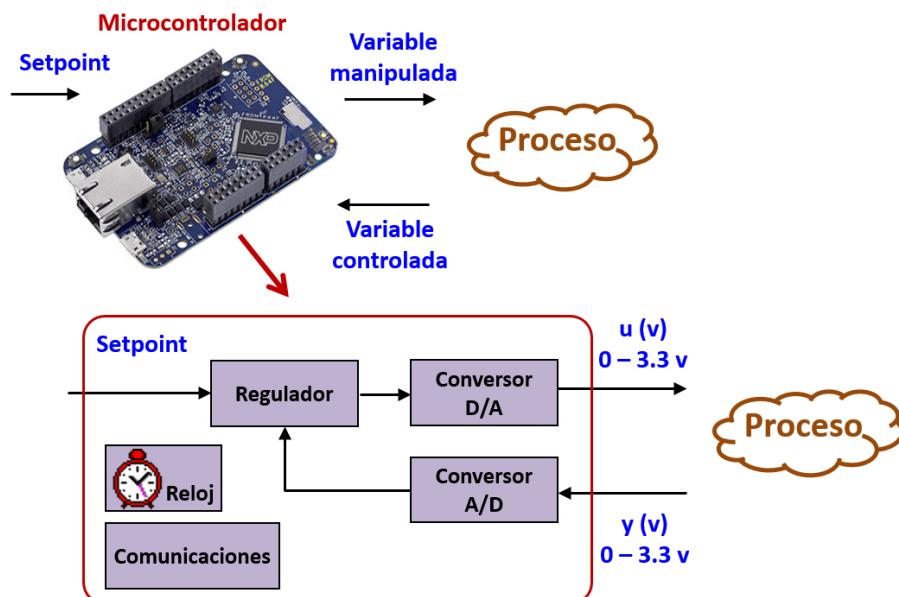


Figura 3.10: Esquema del funcionamiento del microcontrolador.

Profundizando más en el acondicionamiento de señales, y como se puede ver en la Figura 3.11, es necesario realizar una acondicionamiento tanto de la señal que recibe el microcontrolador desde el proceso, como de la que calcula el microcontrolador y envía al proceso.

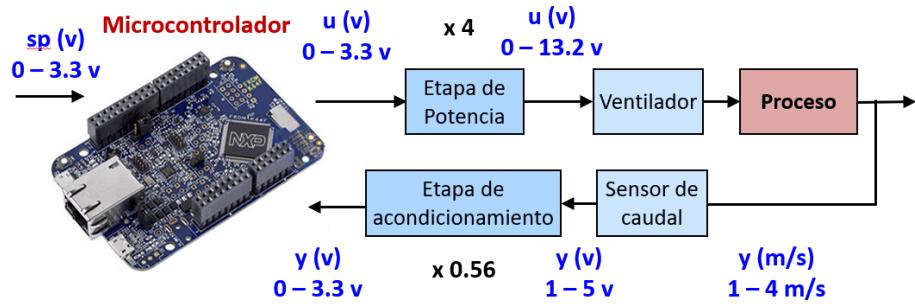


Figura 3.11: Esquema acondicionamiento de señales.

En lo referente a la comunicación entre el usuario y la placa (Figura 3.12), y como ya se ha mencionado anteriormente, se realizará a través de una conexión serie. La finalidad de esta conexión consistirá en tener la posibilidad de configurar y programar la placa, programar el regulador, debuggear el programa realizado e intercambiar información entre el usuario y la placa (por ejemplo, pasar el setpoint).



Figura 3.12: Esquema de la conexión entre la planta y el ordenador.

3.5. SCADA

SCADA (Supervisory Control And Data Acquisition, Supervisión, Control Y Adquisición de Datos) hace referencia a toda aquella aplicación que obtenga datos de un "sistema" (por ejemplo, un caudalímetro en un proceso industrial) con el fin de controlar, supervisar y optimizar dichos datos a distancia [9].

Para todo tipo de negocio, la automatización con SCADA ofrece una maximización del rendimiento de sus activos a través de la excelencia operativa. En concreto, la interfaz hombre-máquina desarrollada en este trabajo se corresponde con lo que a nivel industrial se denomina SCADA.

En la Figura 3.13 se puede ver un ejemplo de un esquema del funcionamiento a nivel industrial de un PLC implementando, además de la conexión con un SCADA (lo que en este trabajo equivaldría a la aplicación) que permite interactuar con el proceso, otras funcionalidades.

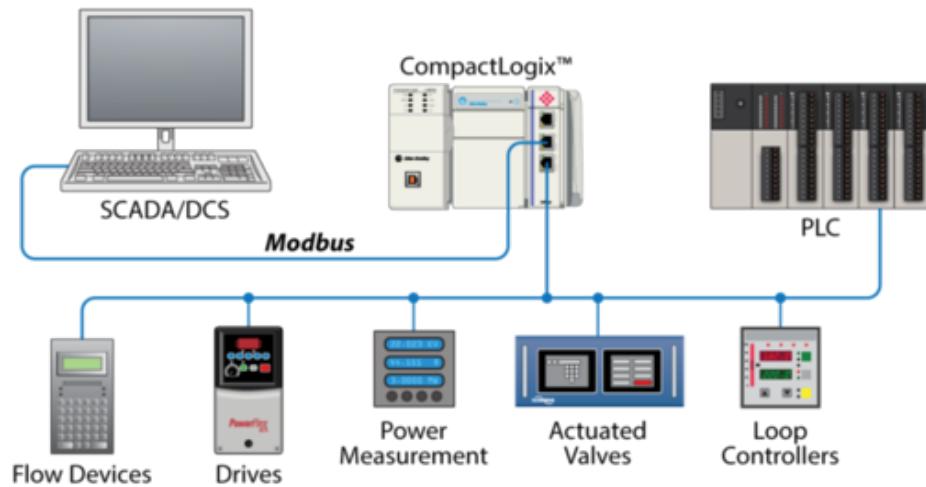


Figura 3.13: Ejemplo esquema implementación SCADA.

Básicamente, el PLC es donde reside el software de control de un proceso industrial, tiene una CPU, RAM y una serie de tarjetas de entrada y salida que se conectan mediante cables a los instrumentos del proceso (sería el equivalente a la placa NXP).

La comunicación con el SCADA u otro dispositivos actualmente y a nivel industrial no se realiza mediante comunicación serie, sino, con protocolos basados en TCP/IP o protocolos de comunicaciones propietarios (modbus, fieldbus, profibus, etc.)

3.6. Entorno de desarrollo integrado (IDE)

Un IDE (Integrated Development Environment) es una aplicación informática constituida por un editor de código fuente, un depurador y un constructor de interfaz gráfica (GUI). Algunos de estos IDEs se completan añadiendo un compilador, un intérprete o, incluso, un autocompletado de código (IntelliSense).

La finalidad de esta aplicación es facilitar el desarrollo y la programación de software. Algunos ejemplos de entornos de desarrollo integrado actuales son **NetBeans**, **Visual Studio**, **Eclipse**, **JDK** (Java) y **KDevelop**, entre otros.

Técnicas y herramientas

En este apartado se detallarán las técnicas y herramientas utilizadas para desarrollar el proyecto.

4.1. Metodología

4.1.1. Scrum

Scrum es un marco de trabajo para el desarrollo de *software* que se engloba dentro de las metodologías ágiles. Sus principales características son la adopción de una estrategia de desarrollo incremental, basar la calidad del resultado en el conocimiento tácito de los integrantes de equipos auto organizados, y en el solapamiento de las diferentes fases del desarrollo [10].

En este caso los *sprints* han sido de dos semanas, tras los cuales se realizaba una reunión para definir las nuevas tareas del siguiente *sprint* y realizar una retroalimentación del *sprint* recién hecho.

4.1.2. Técnica de Pomodoro

Para mejorar la concentración durante el desarrollo del proyecto se ha utilizado la técnica Pomodoro, método de gestión del tiempo consistente en intervalos de 25 minutos de concentración intensa seguidos de descansos de 5 minutos, y descansos de 30 minutos después de haber completado cuatro ciclos 25+5 [11].

4.2. Patrones de diseño

4.2.1. MVC - Modelo Vista Controlador

A la hora de estructurar el desarrollo de la aplicación se ha optado por la utilización del Modelo Vista Controlador, consiguiendo así separar la capa que representa la realidad, la capa que conoce los métodos y atributos del modelo, recibiendo y realizando las peticiones del usuario, y la capa visible para el usuario [12].

4.3. Control de versiones

4.3.1. Git

Git es, a día de hoy, el sistema de control de versiones distribuido más usado del mundo. Su propósito es llevar el registro de los cambios realizados en los archivos de un ordenador y coordinar el trabajo de un equipo formado por varias personas sobre un mismo archivo [13].

La otra opción tenida en cuenta para este proyecto fue Subversion, que fue descartada por la facilidad de uso que ofrece Git, la familiaridad con el sistema de control de versiones y las herramientas con las que se puede complementar.

4.4. Hosting de repositorio

4.4.1. GitHub

Una de las plataformas de desarrollo colaborativo más popular cuyo objetivo es alojar proyectos usando el sistema de control de versiones Git es GitHub. Ofrece un repositorio Git donde los desarrolladores pueden almacenar, compartir, testear y colaborar en distintos tipos de proyectos.

Al comenzar el proyecto también se tuvo en cuenta la posibilidad de utilizar GitLab pero fue descartado por la compatibilidad de GitHub con otras herramientas usadas en el desarrollo del proyecto y por la familiaridad con dicho hosting de repositorio.

4.5. Gestión del proyecto

4.5.1. ZenHub

ZenHub es una herramienta gratuita para proyectos pequeños u *open source* cuya finalidad es la gestión de proyectos en GitHub. Esta herramienta trabaja de manera totalmente integrada ofreciendo un tablero canvas¹ en el que cada *issue* nativo de GitHub se representa como una tarea. A estas tareas se le pueden asignar prioridad, dependencias, estimaciones de tiempo de realización, colaboradores y el *sprint* al que pertenece. ZenHub también permite graficar los avances llevados a cabo en el proyecto, como se puede apreciar en la Figura 4.14.

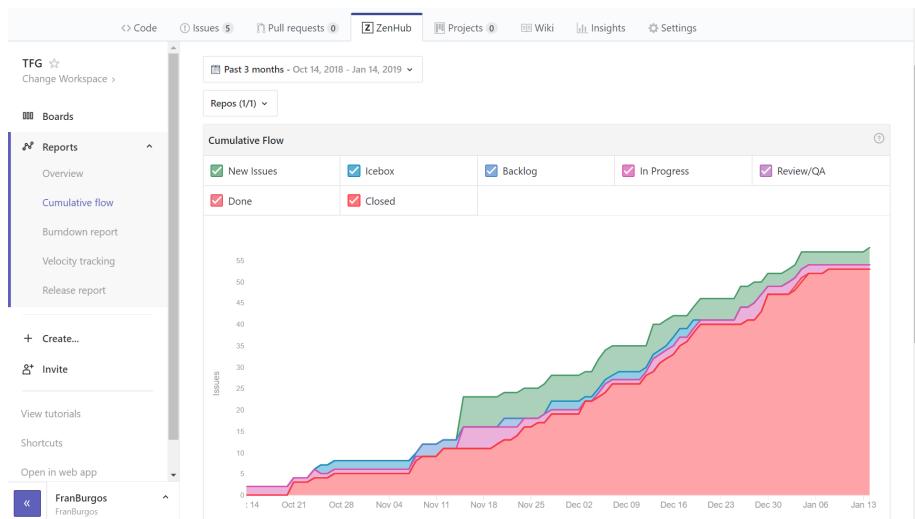


Figura 4.14: ZenHub Overview

En un principio también se barajó utilizar GitHub Proyect pero tras ver los vídeos explicativos subidos en la plataforma de la universidad me decanté por ZenHub.

4.6. Entorno de desarrollo integrado (IDE)

4.6.1. Microsoft Visual Studio 2017

Microsoft Visual Studio 2017 es un entorno de desarrollo integrado para

¹Tablero canvas: lienzo en el que se muestran las tareas pertenecientes a un proyecto ordenadas según el estado de la misma. Admite filtros, modificaciones y múltiples funcionalidades según la herramienta usada.

sistemas operativos Windows y macOS con todas las características para Android, iOS, Windows, la Web y la nube. Este IDE permite escribir código de una manera eficiente y precisa sin perder el contexto del archivo, dotando al programador de facilidades en la programación [14]. En la Figura 4.15 se puede ver un ejemplo del IDE utilizado con la aplicación abierta.

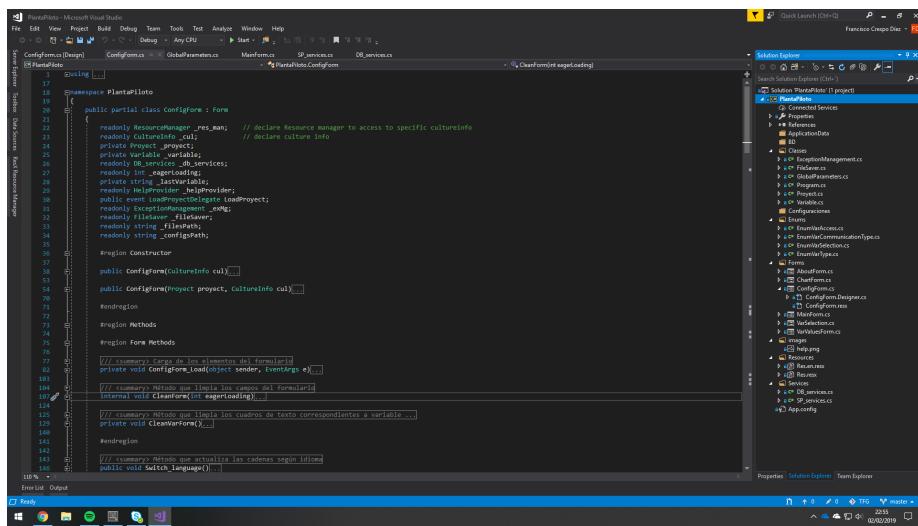


Figura 4.15: Ventana de desarrollo del proyecto en Visual Studio 2017

Cuando se inició el proyecto también se tuvo en cuenta usar versiones anteriores de Visual Studio por sus compatibilidades a la hora de desarrollar Windows Forms sobre C++, pero tras enfrentar las ventajas de cada IDE se decidió usar el Visual Studio 2017 con Windows Forms en C#.

4.7. Comunicación

La comunicación con los tutores se ha realizado a través de varias vías:

- Correo electrónico.
- Reuniones presenciales.
- Canvas de ZenHub, donde se puede comentar cada tarea.

También se pensó en utilizar herramientas como **Slack**² pero se descartaron porque se vieron innecesarias.

4.8. Documentación

4.8.1. LaTeX

LA**T**E**X** es un sistema de composición de textos de alta calidad distribuido como software libre, el cual incluye características diseñadas para la producción tanto técnica como científica, siendo así el estándar de facto para la publicación y comunicación de documentos de carácter científico. [15].

Para la correcta presentación de esta documentación que está siendo leída se ha utilizado la plantilla ofrecida por la Universidad de Burgos, con la finalidad de alcanzar una mayor afinidad en el formato de la misma.

En un principio se planteó la posibilidad de utilizar tanto Open Office Writer como Microsoft Word, en el primer caso porque la Universidad de Burgos ofrecía una plantilla oficial, y en el segundo porque la universidad incluye una licencia para su uso, sin olvidar lo familiarizado que estoy a ella.

4.8.2. HTML Help Workshop

HTML Help Workshop es un programa que permite crear, editar y compilar archivos HTML para convertirlos en proyectos CHM. Este archivo CHM será el encargado de, como se muestra en la Figura 4.16, mostrar la ayuda presente en todas las ventanas de la aplicación.

²Slack: herramienta de comunicación a nivel empresarial que ofrece organización por temas, chats y grupos privados, así como integración en distintas herramientas software, como por ejemplo, Visual Studio.

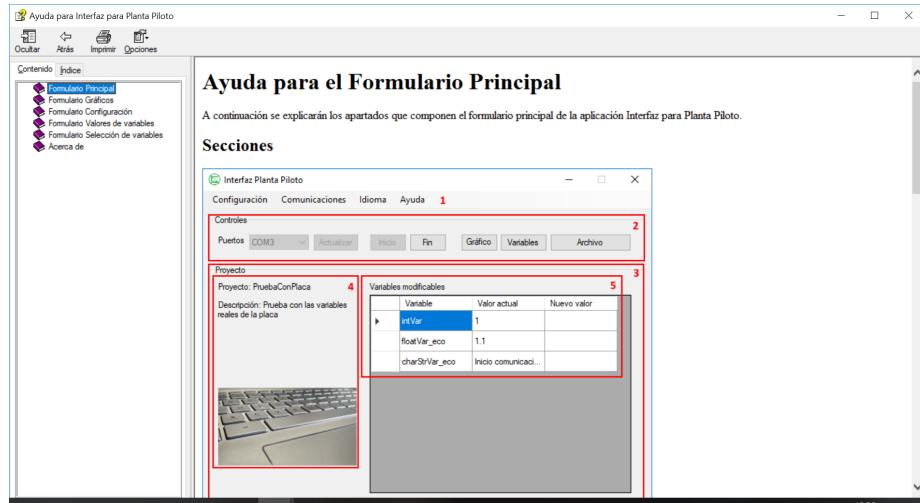


Figura 4.16: CHM File

4.9. Test

4.9.1. Codacy

Para llevar a cabo una revisión del código de la aplicación se utilizó [Codacy](#), herramienta que ejecuta de manera automática exámenes sobre el código de cada *commit* que se realice en Git. Codacy informa de los problemas de seguridad, de cobertura, de duplicidad y de complicidad que pueda tener el código, catalogando a este según los baremos de la propia aplicación y graficando los resultados. En la Figura 4.17 se puede apreciar el avance de los errores en el proyecto en el último mes.



Figura 4.17: Codacy Dashboard

4.10. Otras herramientas

4.10.1. Paint.NET

[Paint.NET](#) es una herramienta de edición de imágenes sobre entorno de Windows. Ofrece una gran cantidad de posibilidades a la hora de editar una imagen, es un programa ligero y de muy fácil utilización. También se tuvo en cuenta [Gimp](#) como herramienta de edición de imágenes pero se descartó porque tardaba mucho en ejecutarse.

4.10.2. Termite

[Termite](#) es un terminal RS232 que actúa como una interfaz de chat teniendo como canal un puerto serie. Cabe destacar la simplicidad de esta herramienta y su facilidad de instalación. Su uso permitió la comprobación de la comunicación con la placa y la realización de pruebas sobre la misma.

4.10.3. Active Presenter

[Active Presenter](#) es un programa dedicado a la grabación de pantalla, edición de vídeo y creación de tutoriales. Antes de elegir este producto se

tuvieron presentes otras opciones como **Camtasia** o **Ice Cream Screen Recorder** pero se eligió Active Presenter por haber trabajado con ello previamente y por las herramientas de edición que ofrece.

4.10.4. P&E Driver

P&E driver contiene el software que provee al equipo de los últimos drivers, aplicaciones y actualizaciones de firmware para los productos OpenSDA de NXP. Es el que permite que se emule una comunicación serie RS232 sobre un puerto UART. Tanto para programar la placa directamente como para comunicarse con ella online.

Estos drivers fueron facilitados por los tutores al inicio del proyecto puesto que si no la comunicación con la placa era imposible, ni siquiera utilizando el programa Termite.

4.11. Base de datos

4.11.1. Microsoft SQL Server

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales, diseñado por Microsoft, que se ejecuta en T-SQL (Transact SQL), consiguiendo así añadir características varias a SQL estándar (control de transacciones, excepción y manejo de errores, procesamiento fila, entre otras) [16].

Dentro de las posibilidades que ofrecía el mercado para cubrir la necesidad de base de datos que tenía la aplicación se tuvieron en cuenta **MariaDB** y **MySQL**. La decisión de usar Microsoft SQL Server se vio motivada a la compatibilidad con el resto de aplicaciones que se estaban usando y por la facilidad de uso que presentaba.

4.11.4. Microsoft SQL Server Management Studio 17

Microsoft SQL Server Management Studio 17 es un entorno integrado que permite administrar cualquier infraestructura de SQL.

Se tuvo presente esta herramienta durante el desarrollo de la aplicación para poder interactuar con la base de datos.

4.12. Editores

4.12.1. Texmaker

Texmaker es un editor multiplataforma gratuito (licencia GNU GLP v3.0) para L^AT_EX que integra la mayoría de herramientas necesarias para la escritura de documentos, corrector ortográfico, auto-completado, resultado de análisis, visor de PDF integrado, entre otros.

En un principio se trató de utilizar **Visual Studio Code** con la extensión LaTeX Workshop pero no ofreció un uso eficiente a la hora de compilar toda la solución L^AT_EX.

4.12.2. Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y MacOs. Visual Studio Code se basa en **Electron**, un framework que se utiliza para implementar aplicaciones **Node.js** para el escritorio. Este editor permite trabajar no sólo con C# y VB (lenguajes propietarios de Microsoft), sino con toda una colección de lenguajes Open Source (Java, Python, PHP, JavaScript, JSON, entre otros) [17].

Aparte de incluir IntelliSense (predicciones ofrecidas por el editor mientras el usuario está escribiendo) y control de versiones a través de Git, el software permite instalar un serie de plugins que completan las necesidades que puedan aparecer según el lenguaje con el que se esté programando.

Este editor fue utilizado para el diseño de las páginas HTML usadas en los archivos de ayuda CHM.

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más importantes del desarrollo del proyecto, presentando los problemas que fueron aconteciendo y los caminos que se eligieron para solventarlos.

5.1. Idea e inicio del proyecto

La elección de este proyecto se vio motivada por varias razones. Por un lado, el interés personal por desarrollar una aplicación que se comunicara con un dispositivo hardware programable, siempre había sido un asunto pendiente. Por otro lado, la posibilidad de ayudar a futuros alumnos de la Universidad de Burgos a tener una facilidad más con la que pudieran contar a la hora de llevar a cabo sus estudios, y evitando de este modo, que el proyecto cayera en el olvido.

Durante las primeras tomas de contacto con los tutores, se barajaron las posibilidades en las que el proyecto podría ser desarrollado, eligiendo en un principio trabajar en C++. Esta decisión se vio motivada porque la placa viene programada en ese lenguaje y por los conocimientos de los tutores en el mismo. Tras un pequeño periodo de formación en C++ y tras investigar las posibilidades que ofrecía C# para cubrir las necesidades del proyecto, se replanteó el desarrollo del mismo, decidiendo cambiar a C# y a la última versión de Visual Studio en la que me encontraba más cómodo.

Una vez definido el lenguaje de desarrollo se trataron los temas relacionados con el repositorio que se iba a utilizar, en este caso GitHub. También se comentaron las herramientas y entornos de desarrollo que se iban a uti-

lizar (ZenHub, Visual Studio 2017), y la frecuencia con la que se iban a realizar reuniones para tratar los avances del proyecto, siendo periodos de dos semanas.

5.2. Formación

Desde que el tema fue elegido, hasta prácticamente la entrega de toda la documentación, ha sido una formación continua. Los distintos problemas y lagunas personales que se iban encontrando durante el desarrollo de la aplicación han servido como base para buscar el conocimiento necesario para poder finalizar con éxito dicha aplicación.

La mayoría de las herramientas utilizadas desde el inicio requirieron de una formación previa, ya fuera para ampliar conocimiento, como podría ser el caso de GitHub, o para crearlo, como en el caso de L^AT_EX. Al igual que con las herramientas, también se realizó una investigación sobre el tema elegido, sumergiéndose así en el mundo del hardware NXP boards y las interfaces hombre-máquina.

Dentro de todo el conocimiento adquirido, cabe destacar todo aquél relacionado con C#, tanto en la forma de estructurar un proyecto desde cero, como a la hora de programar. Por ejemplo, muchas líneas de código han sido ahorradas gracias a las expresiones Lambda, potenciando así la lógica y el pensamiento racional ante la programación básica con la que se contaba hasta el momento. Los manuales más consultados durante todo este tiempo han sido los proporcionados por Microsoft para dar luz sobre dudas a la hora de implementar clases o instrucciones no conocidas o no utilizadas, ergo se podría concluir que el proceso de formación más notable y provechoso en este proyecto ha sido en lo referente a la programación en C# y la utilización de la herramienta Visual Studio 2017.

5.3. Desarrollo de la interfaz y la internacionalización

Los primeros pasos llevados a cabo en el desarrollo de la aplicación fueron para construir las interfaces gráficas. Utilizando los conocimientos adquiridos durante mis estudios más los que se han ido añadiendo por los requerimientos del proyecto, se ha empezado la construcción de cada ventana definiendo su interfaz. Esto no implica que todas las interfaces fueran creadas en un inicio,

sino que a la hora de crear una nueva ventana lo primero que se desarrollaba era la interfaz, pudiendo ésta sufrir actualizaciones por posibles mejoras.

Aunque en un principio no se contempló en los requerimientos funcionales de la aplicación, me pareció útil y provechoso añadir multiculturalidad al proyecto dotándolo de la posibilidad de cambiar de idioma durante la ejecución del mismo. Visto que esta aplicación iba a ser utilizada por alumnos de la Universidad de Burgos y atendiendo al aumento de su internacionalidad recibiendo alumnos de otras universidades, resulta positivo para estos nuevos estudiantes ofrecer métodos de aprendizaje que tengan cierta flexibilidad que les ayude durante sus estudios. El método usado para implantar varios idiomas otorga facilidad en las líneas de trabajo futuras que quieran incluir nuevos idiomas, teniendo sólo que añadir las cadenas de texto traducidas a un archivo de recursos.

5.4. Desarrollo de la comunicación a través del puerto serie

Primera versión

La primera forma implantada para dar una solución a la comunicación serie consistió en dotar a la ventana principal de acceso directo a la clase que se encaraba de la comunicación serie. Conseguir la comunicación fue relativamente sencillo, puesto que, gracias a las librerías que incluye Visual Studio, lo único que hay que contemplar son los valores a asignar para realizar la conexión e instanciar la clase.

Tras realizar pruebas sobre esta primera ventana no se encontraron problemas, la aplicación funcionaba con normalidad. El problema apareció cuando se quería permanecer escuchando el puerto serie mientras se ejecutaba el resto de la aplicación, puesto que esta no permitía hacer nada más, quedándose bloqueada. Debido a este problema surgió la segunda versión.

Segunda versión

Tras los problemas encontrados en la primera versión, el desarrollo de todo aquello relacionado con el puerto serie fue movido a una clase apartada del resto del código. Para evitar que la aplicación no permitiera realizar más acciones cuando la conexión al puerto serie estuviera abierta, se implementó el uso de un hilo dedicado exclusivamente a la gestión de dicho puerto, incluyendo todos los métodos que tuvieran relación con él.

5.5. Desarrollo de la comunicación con la base de datos

Se podría decir que el desarrollo de la comunicación con la base de datos ha sido la parte más problemática en la construcción de la aplicación, aunque a su vez, también ha sido la parte que nos ha mostrado las carencias que presentaba el código, y que, si lo que se buscaba era implantar la herramienta para el uso docente, era necesario resolver.

Inicialmente, todo aquello relacionado con la conexión a la base de datos se encontraba en la misma clase que los servicios de conexión al puerto serie. Al poco tiempo, se decidió que era más eficiente tener ambas características separadas.

Primera versión

La primera versión que se utilizó sobre la aplicación para resolver la necesidad de implementar una base de datos fue la creación de una conexión sobre una base de datos ya creada. La creación de esta conexión se hizo a través del ayudante que ofrece Visual Studio para añadir fuentes de datos a un proyecto.

Esta primera versión fue muy útil para desarrollar la aplicación y hacer pruebas en mis equipos. Los problemas aparecieron cuando se lanzaron las distintas releases del producto y se quisieron probar en los ordenadores de los tutores del proyecto.

Segunda versión

Para solucionar los problemas de que presentaban los ordenadores de los tutores, se automatizó la creación tanto de la base de datos necesaria como la conexión a esta. Los problemas persistieron por culpa de no haber eliminado de manera completa la conexión creada en la primera versión y por usar una versión distinta de SQL Server.

Cabe añadir que durante este periodo, no sólo se trabajó con consultas de bases de datos, sino que también hubo que trabajar con los servicios de SQL Server y la asignación de permisos a usuarios concretos para crear los archivos necesarios para crear una base de datos de manera desasistida.

5.6. Control de excepciones

Una de las últimas características añadidas al proyecto fue el control de las excepciones ocurridas durante la ejecución de la aplicación. Para poder canalizar todos los tipos de excepciones se creó una clase cuya única finalidad fue recoger las excepciones ocurridas y plasmarlas en un archivo de log para poder tener un histórico de las mismas.

Esta funcionalidad resulta muy útil tanto en el desarrollo de la aplicación como en la implantación de la misma en los ordenadores de los alumnos, para conocer el origen de los errores que vayan apareciendo. Controlar todas las excepciones que acontecen durante la ejecución dota a la aplicación de fluidez a ojos del usuario final.

5.7. Pruebas

Durante todo la programación de la aplicación se han llevado a cabo pruebas sobre todas las funcionalidades que se iban añadiendo, teniendo en cuenta todos los casos posibles y aceptados, así como los que debían de descartarse. Se barajó la posibilidad de utilizar TDD durante el desarrollo, pero la falta de conocimiento sobre el tema y la falta de tiempo hizo que esta opción se descartara.

Possiblemente uno de los momentos más satisfactorios de este proyecto fue ver como la aplicación funcionaba con la planta piloto conectada. Ver el fruto del trabajo de estos últimos meses en un entorno de pruebas real se convierte en una sensación verdaderamente gratificante para el desarrollador.

Descripción de la herramienta

A lo largo de este capítulo el lector podrá conocer la descripción, las características y funcionalidad de la herramienta. De esta manera, éste tendrá un acercamiento a la misma previo a su ejecución.

6.1. Descripción

Se define este software como aquel encargado de proporcionar una interfaz a la comunicación serie entre una placa NXP FRDM-K64F y un ordenador, permitiendo a su vez realizar una serie de acciones sobre dichos datos que se verán en el siguiente apartado.

La aplicación se compone de las siguientes ventanas:

- Ventana principal.
- Ventana para la creación de un nuevo proyecto.
- Ventana para la modificación de un proyecto cargado.
- Ventana para la selección de variables.
- Ventana de graficado de valores de variables.
- Ventana de valores de las variables cargadas en el proyecto.
- Ventana de ayuda.
- Ventana Acerca de.
- Apertura del manual de usuario.

6.2. Funcionalidad

La funcionalidad de esta aplicación está repartida en diferentes ventanas que serán vistas a continuación.

Ventana principal

La ventana principal de la aplicación, que se puede ver en la Figura 6.18, es la primera que nos encontramos cuando esta es ejecutada y a partir de la cual se va a poder tener acceso al resto de las funcionalidades incluidas.

Las principales divisiones que se encuentran en esta ventana son:

- Menú: se encuentra en la parte superior de la ventana y contiene el acceso a las funcionalidades de configuración del proyecto, configuración de la conexión, internacionalización y ayuda.
- Controles: son todos aquellos botones que se encuentran bajo la barra de menú y que permiten ejecutar órdenes directas sobre la aplicación.
- Datos del proyecto: en la parte inferior izquierda de la ventana se muestran los datos del proyecto cargado.
- Tabla editable con las variables de escritura cargadas a través del proyecto: en esta tabla se podrá ver el valor actual devuelto por la placa para cada variable y a su vez, se podrá modificar el valor de cada una.

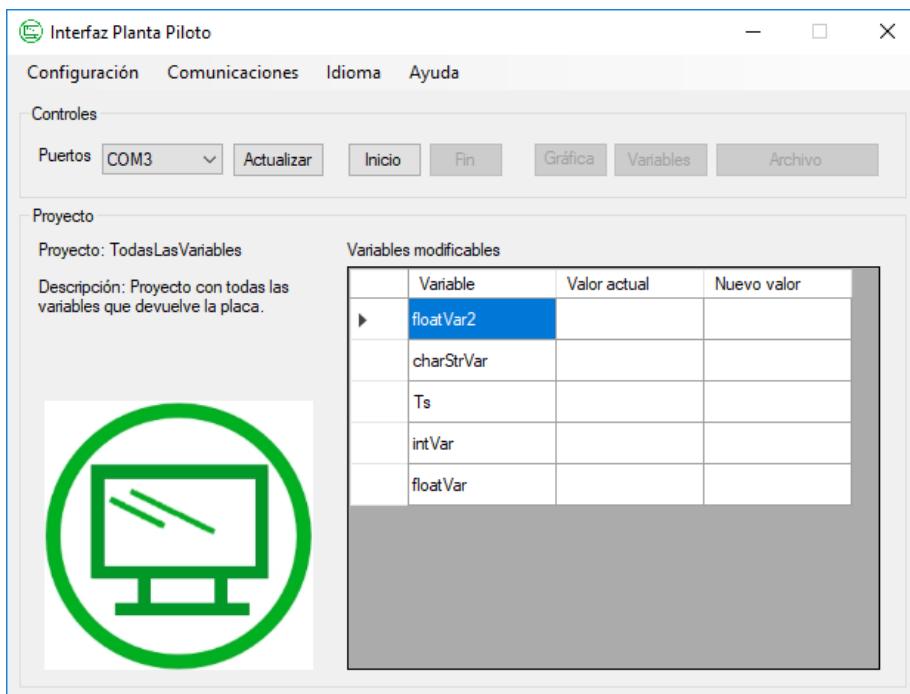


Figura 6.18: Ventana principal de la aplicación.

Ventana para la creación de un nuevo proyecto

A través de la opción de menú “Configuración - Crear configuración” se podrá acceder a la ventana que aparece en la Figura 6.19, la cual permitirá crear y cargar un nuevo proyecto en la aplicación.

Cabe destacar que, para el correcto guardado de cada variable que se quiera añadir al proyecto, hay que pulsar el botón “Añadir variable”, si no esta no será tenida en cuenta.

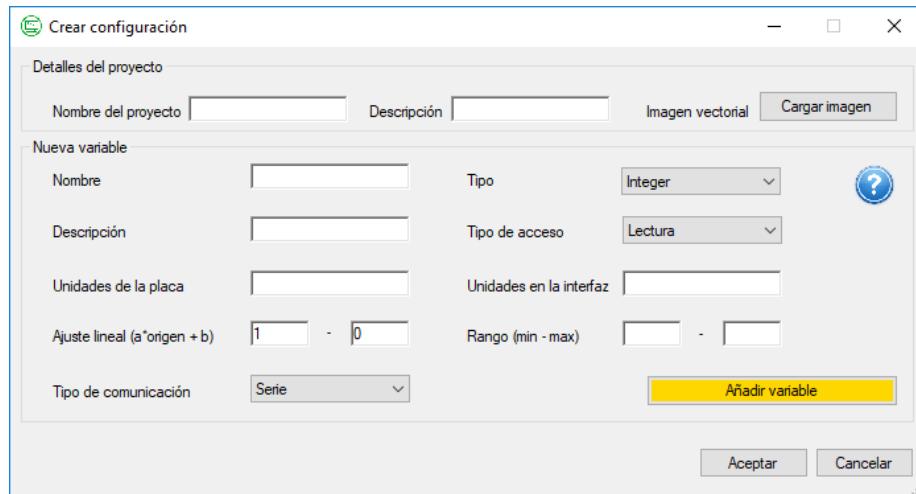


Figura 6.19: Ventana para la creación de un nuevo proyecto.

En el caso de preferir cargar una configuración ya creada, se deberá seleccionar la opción de menú “Configuración - Cargar configuración”.

Ventana para la modificación de un proyecto cargado

En el caso de que se desee modificar la configuración cargada en la aplicación, ya sean los datos del proyecto como sus variables, se podrá hacer a través de “Configuración - Modificar configuración”, como se muestra en la Figura 6.20.

Para modificar los valores de cada variable cargada en el proyecto, se accederá a través del menú desplegable que se encuentra en la parte inferior de la ventana. Al igual que ocurría en la ventana anterior, los cambios no serán guardados hasta que no se confirme a través del botón “Guardar cambios en la variable”.

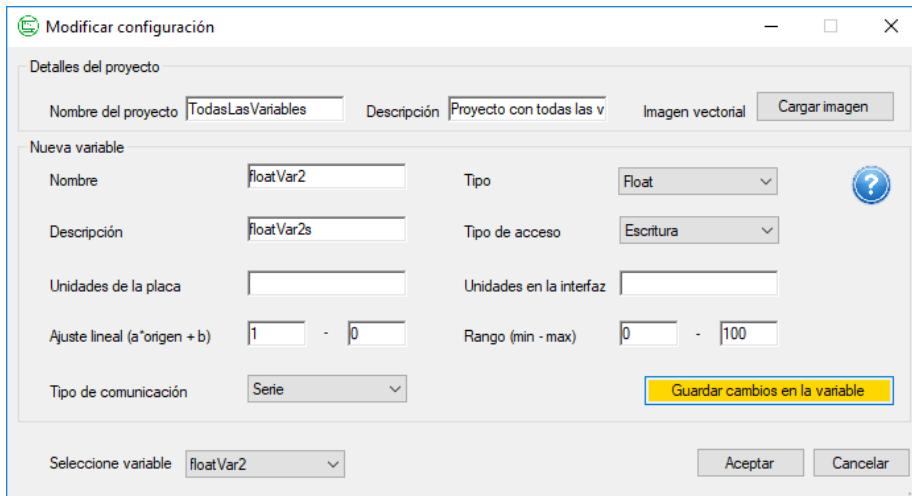


Figura 6.20: Ventana para la modificación de un proyecto cargado.

Ventana para la selección de variables

Una vez que se tenga un proyecto cargado y la conexión con el puerto serie abierta, los valores devueltos por la placa se mostrarán en la tabla de la ventana principal, y los botones que antes se encontraban inactivos en el panel de controles, pasarán a estar activos. Si se pincha sobre cualquiera de los botones “Gráfica”, “Variables” o “Archivo”, la ventana que aparecerá, mostrada en la Figura 6.21, será la que se muestra en la siguiente ilustración.

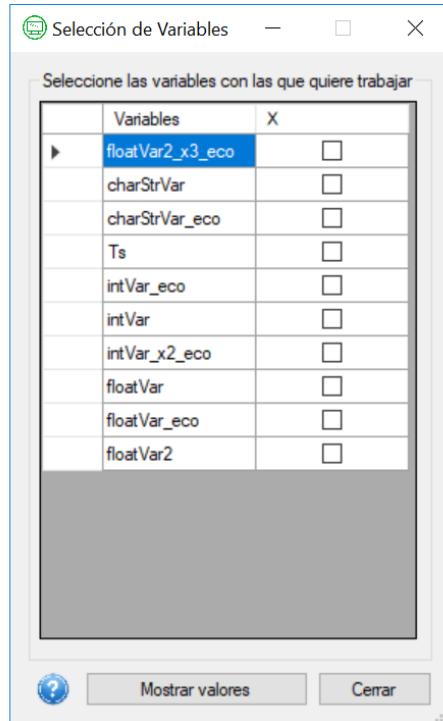


Figura 6.21: Ventana para la selección de variables.

En esta ventana se procederá a la selección de las variables con las que se va a querer trabajar según el botón que se haya pulsado.

Ventana de dibujado de gráficas

A través del botón “Gráfica” que se encuentra en la ventana principal, y habiendo seleccionado las variables que se quieren graficar, se accederá a la ventana que se muestra en la Figura 6.22.

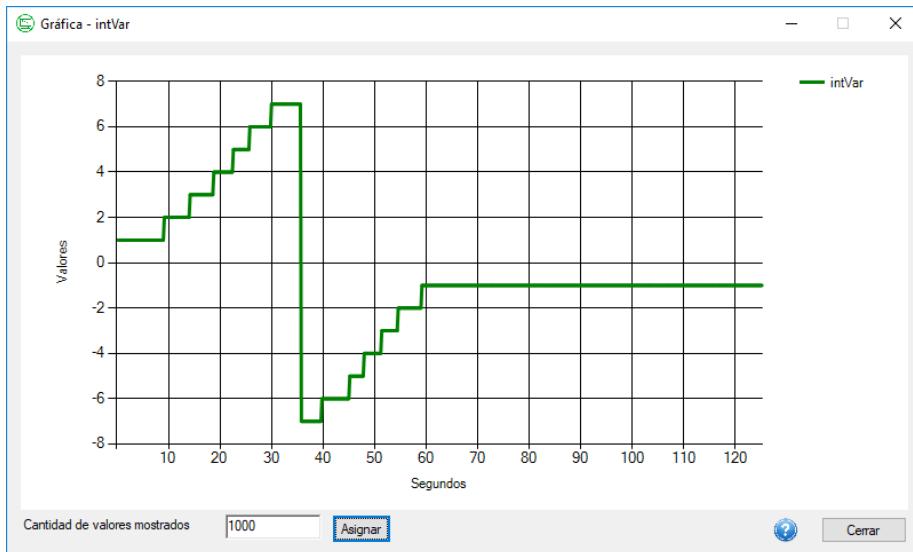


Figura 6.22: Ventana de graficado de valores de variables.

En esta ventana se podrá ver una gráfica con los valores de las variables que hayan sido seleccionadas, pudiendo modificar la cantidad de valores que se muestran. Se podrá seguir usando el resto de la aplicación mientras esta ventana permanece abierta.

Ventana de valores de las variables cargadas en el proyecto

A través del botón “Variables” que se encuentra en la ventana principal, y habiendo seleccionado las variables que se quieren mostrar, se accederá a la ventana que se muestra en la Figura 6.23.

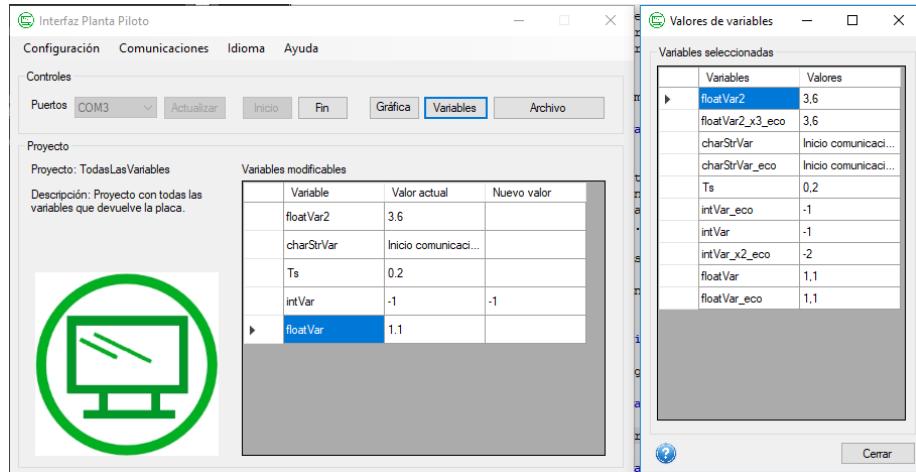


Figura 6.23: Ventana de dibujado de gráficas.

Esta ventana permite mostrar los valores actualizados de todas las variables, ya sean de lectura o de escritura. Se podrá seguir usando el resto de la aplicación mientras esta ventana permanece abierta.

Guardado del valor de las variables en un archivo

En el caso de haber pulsado el botón “Archivo” no aparecerá ninguna nueva ventana, sino que se volverá a la ventana principal, ejecutándose el guardado en archivo de los valores de las variables seleccionadas en segundo plano. Cuando esto ocurra, el texto de botón habrá cambiado a “Detener guardado”, y se pinchará sobre él en el momento en el que se quiera finalizar el guardado.

El archivo creado presentará el formato visible en la Figura 6.24:

```

TodasLasVariables_3variables.txt: Bloc de notas
Archivo Edición Fmtato Ver Ayuda
29/01/2019 21:45:19
TodasLasVariables
Proyecto con todas las variables que devuelve la placa.
C:\Users\fpiza\source\repo\ATFO\PlantaPiloto\PlantaPiloto\bin\Debug\Configuraciones\images\TodasLasVariables_3variables\icono1_1548794719.jpg
*****
Time;Ts;intVar_eco;intVar_x2_eco;
16,4;0,5;5;10;
18,4;0,5;5;10;
20,4;0,5;0;0;
22,4;0,5;0;0;
24,6;0,4;0;0;
26,6;0,4;3;6;
28,6;0,4;3;6;
30,6;0,4;3;6;
32,6;0,4;3;6;
<

```

Figura 6.24: Contenido del archivo de guardado de variables.

Se podrá seguir usando el resto de la aplicación mientras el guardado de valores esté activo.

Ventana de ayuda

En cada ventana de la aplicación se dispondrá de un acceso a la ayuda en línea de la aplicación. Se podrá acceder a esta ayuda, mostrada en la Figura 6.25, desde la ventana principal “Ayuda - Ayuda”, o pinchando el botón azul con una interrogación.

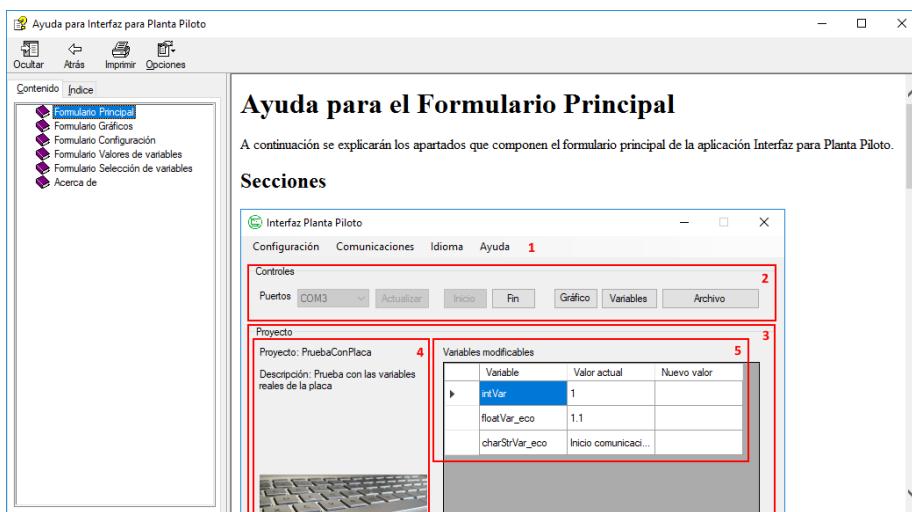


Figura 6.25: Ventana de ayuda.

En esta ventana encontraremos un breve manual acerca de la utilización de la herramienta dividido en ventanas.

Ventana Acerca de

La aplicación contiene una ventana informativa sobre los datos de la aplicación. Se accederá a ella a través de la ventana principal “Ayuda - Acerca de”, mostrada en la Figura 6.26.

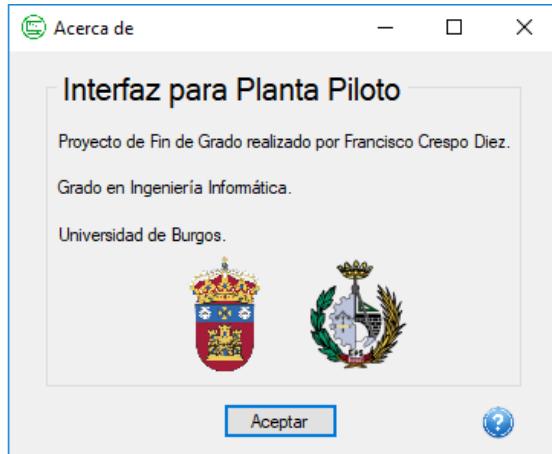


Figura 6.26: Ventana Acerca de.

Manual de usuario

También se dispone de un manual de usuario más completo que la ventana de ayuda en la que se explica con detenimiento las funcionalidades de la aplicación. Se podrá acceder a este archivo PDF a través de la opción de menú “Ayuda - Manual de usuario”, mostrado en la Figura 6.27.

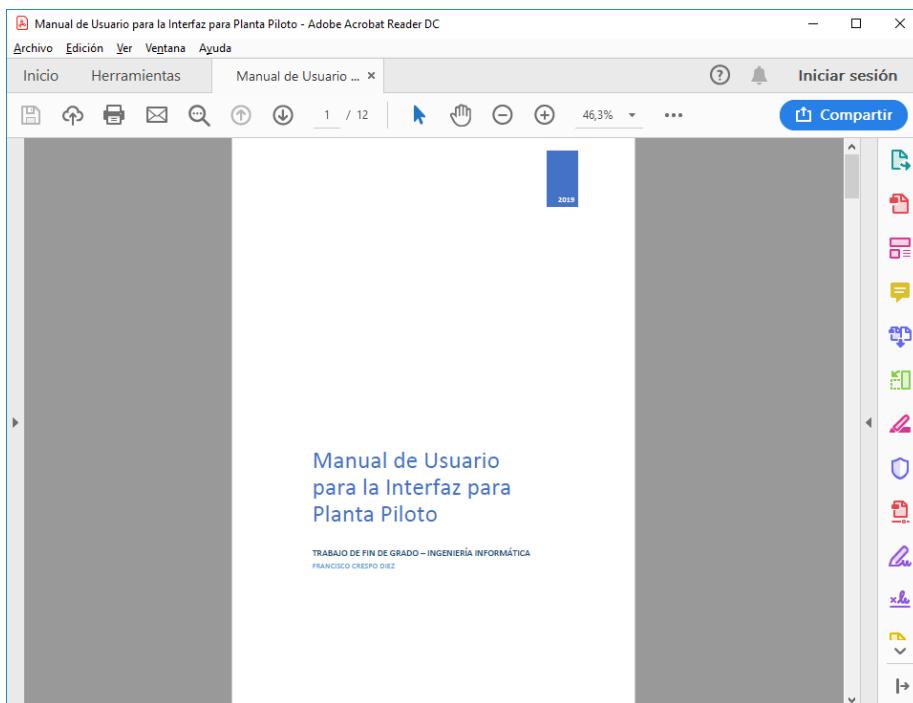


Figura 6.27: PDF - Manual de usuario.

6.3. Características

El software desarrollado en este proyecto requiere para un correcto funcionamiento de:

- Entorno Windows. La aplicación fue desarrollada en Windows 10, aunque ha sido probada en Windows 8.1 y ha funcionado sin problema.
- Driver P&E: este driver permite la correcta comunicación a través del puerto serie con las placas NXP.
- SQL Server 2017: la aplicación utilizará SQL Server como plataforma de base de datos.
- .NET Framework 4.6.1: versión del framework de Microsoft que contiene las librerías necesarias para la ejecución de la aplicación.

Trabajos relacionados

En este capítulo se hablará sobre los trabajos relacionados con el tema del que trata el proyecto.

El desarrollo de toda esta planta piloto dibuja un largo camino en el tiempo, pasando a través de tres trabajos de fin de grado hasta el momento.

7.1. Primera versión

El primer proyecto que se llevó a cabo fue el “Diseño, construcción y control de una planta de laboratorio para la regulación de temperatura y caudal de aire”, por Rubén Zambrana Rodríguez, egresado en Ingeniería Electrónica Industrial y Automática (UBU) en 2017 [6]. A lo largo de este proyecto se construyó un prototipo de la planta piloto. Esta planta cuenta con un caudalímetro, una resistencia, una sonda y un ventilador.

7.2. Segunda versión

En el proyecto “Implementación de reguladores PIDs industriales y reguladores avanzados en microcontroladores”, por María Isabel Revilla Izquierdo, egresada en Ingeniería Electrónica Industrial y Automática (UBU) en 2018 [7], se presenta una segunda versión de la planta mejorada que es la versión actual disponible para los alumnos en el laboratorio.

Conclusiones y líneas de trabajo futuras

En esta sección se exponen tanto las conclusiones derivadas del trabajo, como las posibles líneas de trabajo futuras con las que se puede dar continuidad al proyecto.

8.1. Conclusiones

Desde un punto de vista general y de una manera personal, finalizo este Trabajo de Fin de Grado llevándome conmigo una sensación positiva, tanto por todo lo que he aprendido gracias a la resolución de los diferentes problemas que me han aparecido, como por dejar una herramienta útil para la formación de futuros alumnos de la Universidad de Burgos. Poder haber participado en este proyecto comenzado por mis compañeros Rubén Zambrana Rodríguez y María Isabel Revilla Izquierdo me hace sentir que formo parte de una idea mayor que contiene una finalidad altruista y formativa.

Ciñéndome a todo lo relacionado a la documentación, resulta interesante ver como todo aquello aprendido durante la carrera sobre diagramas, previsiones y un largo etcétera de conceptos, es útil para el planteamiento de un proyecto, ayudando al alumno a ponerse en una situación real de desarrollo de software y en la que puede ver todo lo que está involucrado.

Si atiendo a la parte técnica, he tenido la suerte de poder desarrollar la aplicación en el entorno y lenguaje a los que más habituado estoy, por lo que gracias a ello he conseguido ampliar mis conocimientos sobre C#, Visual Studio y SQL Server, preparándome un poco más para el mundo laboral. Cabe destacar que a la vez que realizaba este trabajo de fin de grado estaba

acudiendo a prácticas extracurriculares en la empresa Keyland Sistemas de Gestión, de donde he obtenido muchas ideas y ayuda tanto para organizar el proyecto como para solventar los problemas que han ido aconteciendo.

En resumen, puedo concluir orgulloso que ha sido un proyecto con el que me he nutrido tanto personal como profesionalmente. Durante este tiempo he sido capaz de medirme y ver hasta dónde alcanzan tanto mis conocimientos como mi habilidad para plasmarlos en un entorno real.

8.2. Líneas de trabajo futuras

Durante la programación de la aplicación aparecieron posibles mejoras que se pueden añadir a la solución en líneas de trabajo futuras.

Comunicación a través del puerto Ethernet

Atendiendo a las posibilidades que ofrece la placa NXP FRDM-K64F resultaría interesante ampliar las posibilidades de comunicación a través del puerto Ethernet. Esta línea de trabajo se ha contemplado desde el inicio del proyecto y se puede ver reflejado en la opción de menú de “Comunicaciones” de la ventana principal, la cual ofrece la opción de “Otros”.

Internacionalización

Atendiendo a que el propósito de esta aplicación es educativo y se presenta en una universidad que tiene presencia a nivel internacional, resultaría provechoso añadir tantos idiomas como fueran posibles a la misma, ampliando así las facilidades dadas a los alumnos extranjeros.

Esta línea de trabajo se tuvo en cuenta desde el diseño de la ventana principal, dejando preparado un archivo de recursos con dos idiomas añadidos y enlazando todas las cadenas de texto de la aplicación con dicho archivo.

TDD

En mi opinión creo que resultaría producente realizar las futuras mejoras utilizando TDD (Test-Driven Development), consiguiendo de esta manera una mayor eficiencia y ahorrando tiempo a la larga en la resolución de futuros problemas que pudieran aparecer.

Variables PID

Actualmente la aplicación trabaja exclusivamente con variables normales, y no atiende a que éstas puedan ser de tipo PID, ni incluye la interfaz específica para ellas. La adición de esta mejora completaría el apartado relativo a las variables contenidas en un proyecto y abriría una abanico de funcionalidades nuevas a incluir a su vez.

Estilos

En esta versión la aplicación se muestra con los estilos predeterminados de una aplicación de Windows Forms. Podría resultar visualmente ventajoso trabajar sobre unos estilos con los que dotar a las ventanas, así como efectos y eventos al realizar acciones como el guardado de una variable o la selección de una imagen.

Mejoras en las opciones de configuración

Una de las mejoras que podría mejorar la aplicación notablemente sería la adición de una ventana en la que se pudieran configurar los parámetros de la comunicación, como por ejemplo, la velocidad de transmisión, paridad, etc.

Bibliografía

- [1] Microsoft, “Información general acerca de .net framework,” 2018. [Internet; descargado 30-enero-2019].
- [2] DevCode, “¿qué es y por qué aprender sql?,” 2018. [Internet; descargado 30-enero-2019].
- [3] Definicion.de, “Definición de sql,” 2018. [Internet; descargado 30-enero-2019].
- [4] IBM, “Comunicación serie,” 2019. [Internet; descargado 31-enero-2019].
- [5] G. Theory, “Puertos y buses 1: I2c y uart,” 2019. [Internet; descargado 13-enero-2019].
- [6] R. Z. Rodríguez, “Diseño, construcción y control de una planta de laboratorio para la regulación de temperatura y caudal de aire,” tech. rep., Grado de Electrónica Industrial y Automática - Universidad de Burgos, 2017.
- [7] M. I. R. Izquierdo, “Implementación de reguladores pids industriales y reguladores avanzados en microcontroladores,” tech. rep., Grado de Electrónica Industrial y Automática - Universidad de Burgos, 2018.
- [8] Freescale, *FRDMK64FUG, FRDM-K64F Freedom Module User’s Guide, Rev. 0.1, 04/2014*, ch. 2 FRDM-K64F hardware overview. •, 2014.
- [9] Wonderware, “Control y optimización de sistemas,” 2019. [Internet; descargado 13-enero-2019].
- [10] P. Ágiles, “¿qué es serum?,” 2018. [Internet; descargado 11-enero-2019].

- [11] Ehorus, “Técnica pomodoro: Qué es y cómo puede ayudarte en tu día a día,” 2017. [Internet; descargado 11-enero-2019].
- [12] Microsoft, “¿qué es un patrón de diseño?,” 2018. [Internet; descargado 11-enero-2019].
- [13] Git, “Git –distributed-is-the-new-centralized,” 2018. [Internet; descargado 12-enero-2019].
- [14] Microsoft, “Ide de visual studio,” 2019. [Internet; descargado 13-enero-2019].
- [15] T. L. Proyect, “Latex – a document preparation system,” 2019. [Internet; descargado 13-enero-2019].
- [16] Microsoft, “Sql server,” 2019. [Internet; descargado 28-enero-2019].
- [17] V. S. Code, “Visual studio code,” 2019. [Internet; descargado 28-enero-2019].