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 — 2 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, profesro 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.

Expone:

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, 2 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 conecatada 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#...

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#...

Índice general

Índice de figuras

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 la no inclusión 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.
- Memoria en formato PDF.

- Anexos en formato PDF.

Además, los siguientes recursos están disponibles en Internet:

- Repositorio del proyecto.

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 monitorización 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.
- 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.
- Distribuir la aplicación resultante para entornos Windows.

2.3. Objetivos personales

- Abarcar el máximo número de conocimientos vistos en el grado.
- 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 Freedom de NXP Freescale.
- 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 siguiente ilustración 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 [4].

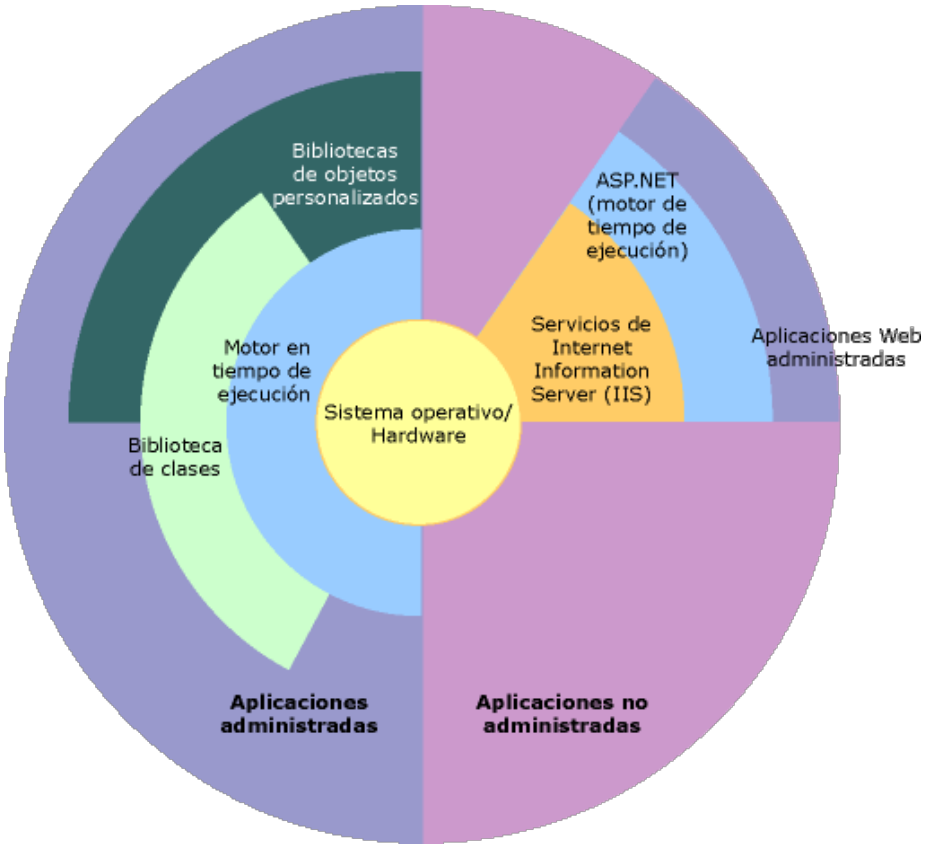

Figura 3.1: Contexto .NET

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.

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. Comunicación Serie

A diferencia de la comunicación en paralelo en la que todos los bits se envían al mismo tiempo, la comunicación serie o secuencial 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.

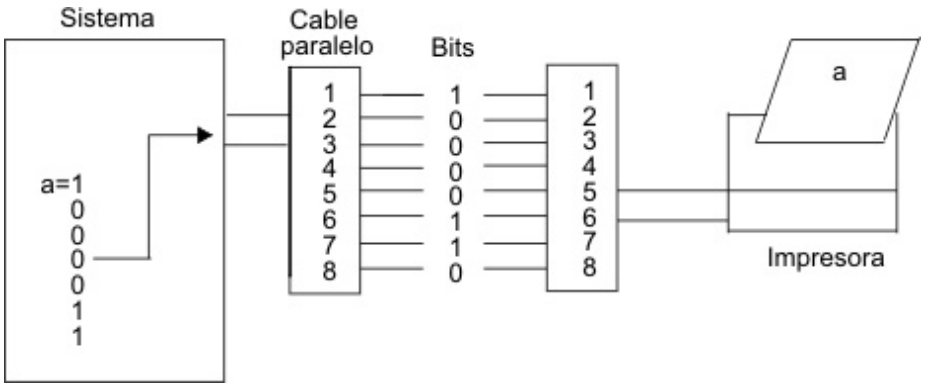

Figura 3.2: Ejemplo de puerto de comunicación paralelo [2].

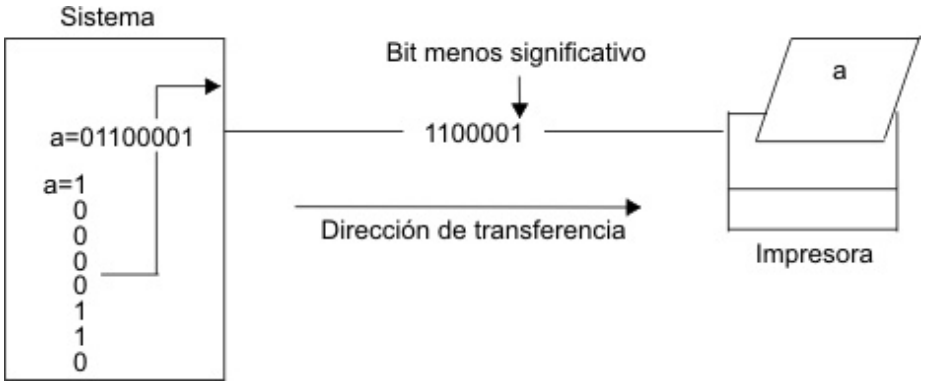

Figura 3.3: Ejemplo de puerto de comunicación serie [2].

3.4. Planta piloto

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

Esta planta piloto fue diseñada y construída por María Isabel Revilla Izquierdo, egresada en Ingeniería Electrónica Industrial y Automática, durante su trabajo de fin de grado, bajo la tutela del Dr. Daniel Sarabia Ortiz.

La planta piloto se compone de:

- Una planta.

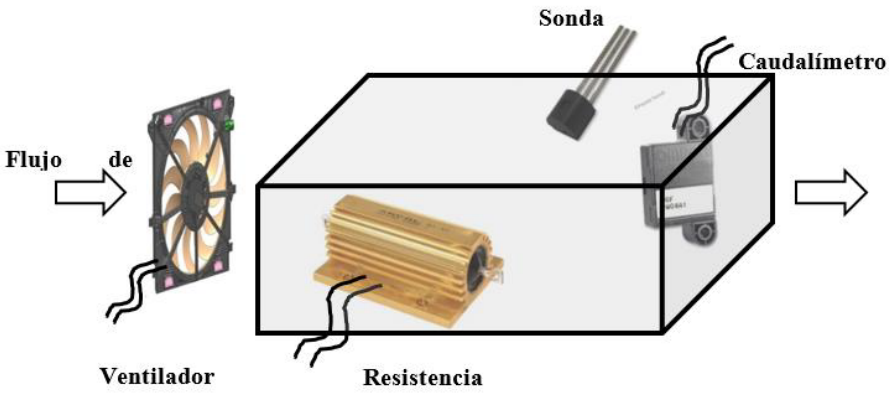

Figura 3.4: Esquema del contenido de la planta [3].

- Un equipo de alimentación.
- Un acondicionamiento.
- Un control (ofrecido por la placa FRDM-K64F).

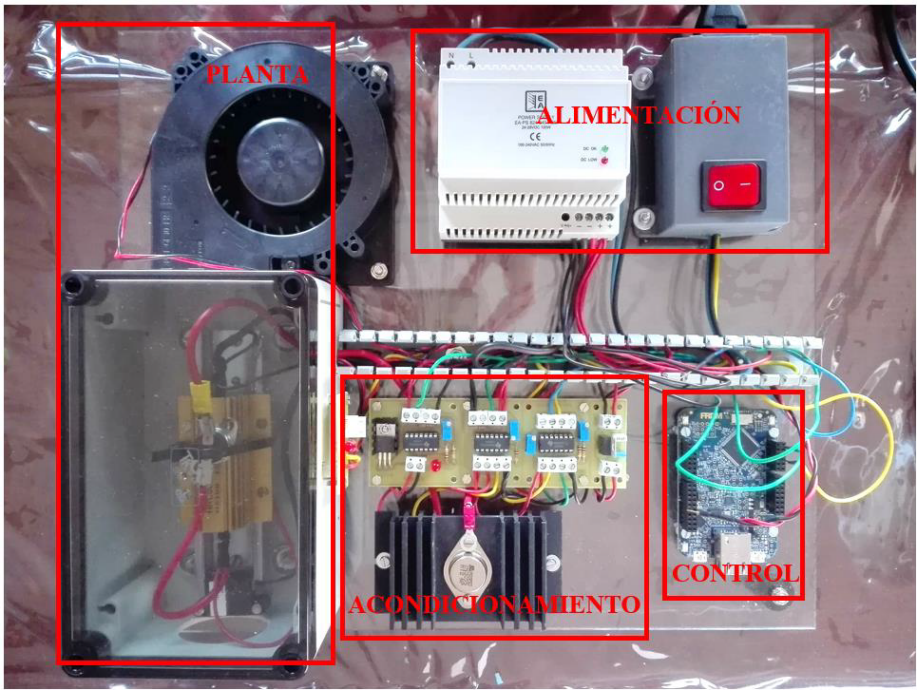

Figura 3.5: Repartición de los componentes de la planta piloto [3].

Freescale Freedom NXP FRDM-K64F

La plataforma de desarrollo Freescale Freedom es un conjunto de herramientas software y hardware para el desarrollo y la evaluación de prototipos de aplicaciones basadas en microcontroladores. Concretamente, el hardware Freescale Freedom K64 (FRDM-K64F), es el que va a ser usado para el proyecto.

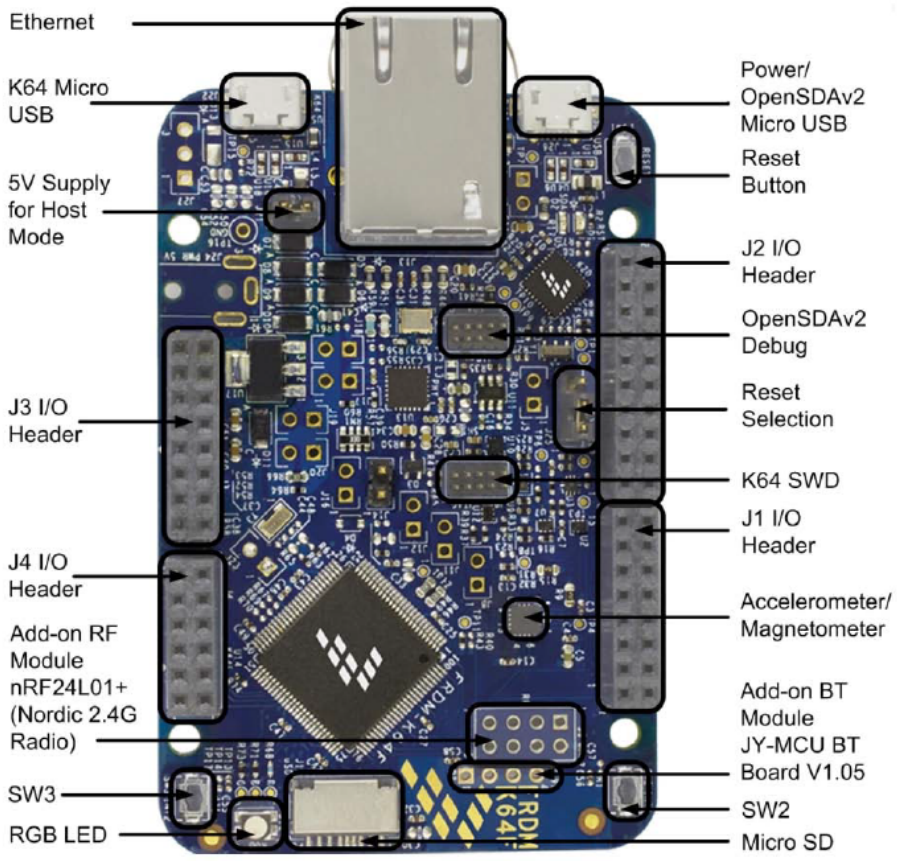

Figura 3.6: Hardware Freescale Freedom K64 (FRDM-K64F) [1]

Durante la ejecución del proyecto los componentes que se van a utilizar del hardware Freescale Freedom K64 (FRDM-K64F) serán "Power/OpenSDAv2 Micro USBƨ el botón Reset button", centrándose de esta manera en la programación de la interfaz hombre-máquina.

3.5. Entorno de desarrollo integrado (IDE)

Un IDE (Integrated Developement Enviroment) 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.

3.6. SCADA

SCADA (Supervisory Control And Data Acquisition, Supercisión, 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.

Para todo tipo de negocio, la automatización con SCADA ofrece una maximización del rendimiento de sus activos a través de la excelencia operativa.

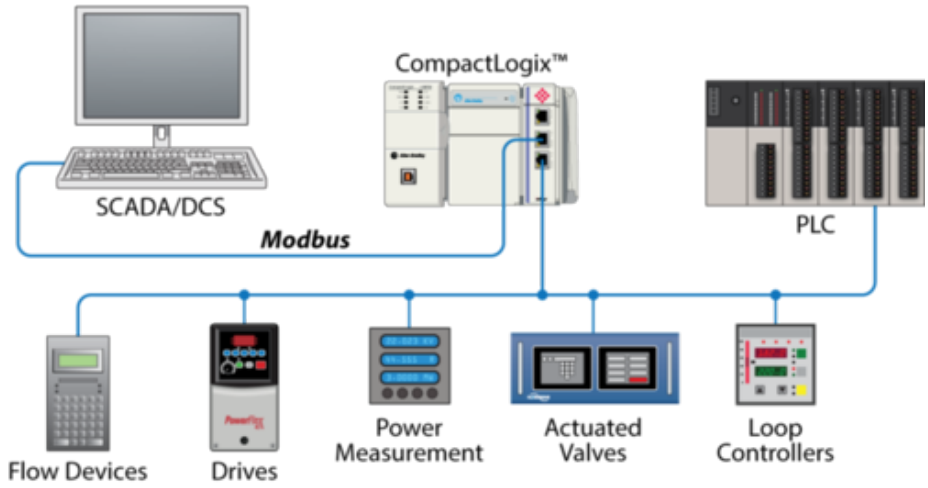

Figura 3.7: Ejemplo esquema implementación SCADA.

También añadir capítulo "Descripción de la herramiento": descripción, funcionalidad y características...

3.7. Secciones

Las secciones se incluyen con el comando section.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.8. Referencias

Las referencias se incluyen en el texto usando cite [10]. Para citar webs, artículos o libros [**?**].

3.9. Imágenes

Se pueden incluir imágenes con los comandos standard de LaTeX, pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:

Figura 3.8: Autómata para una expresión vacía

3.10. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.

Herramientas	App AngularJS	API REST	BD	Memoria
HTML5	X			
CSS3	X			
BOOTSTRAP	X			
JavaScript	X			
AngularJS	X			
Bower	X			
PHP		X		
Karma + Jasmine	X			
Slim framework		X		
Idiorm		X		
Composer		X		
JSON	X	X		
PhpStorm	X	X		
MySQL			X	
PhpMyAdmin			X	
Git + BitBucket	X	X	X	X
MikTEX				X
TEXMaker				X
Astah				X
Balsamiq Mockups	X			
VersionOne	X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

2. segundo item.

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

-

3.11. Tablas

Igualmente se pueden usar los comandos específicos de LATEXo bien usar alguno de los comandos de la plantilla.

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 [**?**].

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 [**?**].

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 [5].

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 [**?**].

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 proyectos web.

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 canvas 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.

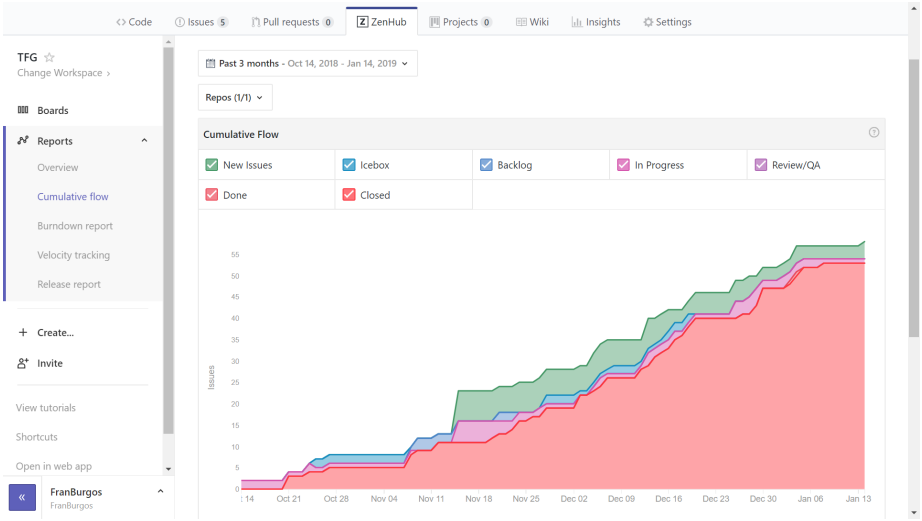

Figura 4.9: ZenHub Overview

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 sistemas operativos Windows y macOs con todas las características para Android, iOS, Windows, la Wev 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 [**?**].

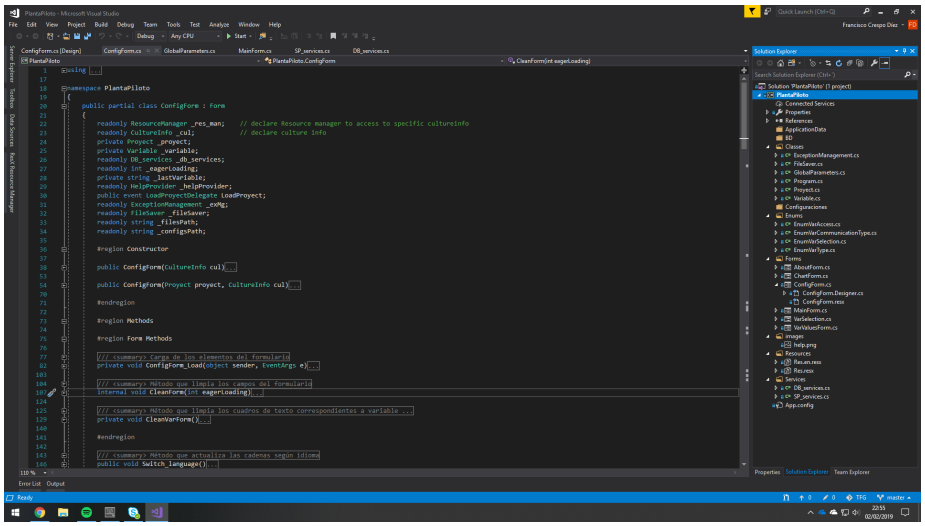

Figura 4.10: Ventana de desarrolo del proyecto en Visual Studio 2017

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 pero se descartaron porque se vieron innecesarias.

4.8. Documentación

4.8.1. LaTeX

LaTeXes un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Por sus

posibilidades y características, es usado de forma especialmente intensa en la generación de libros científicos y artículos que incluyen, entre otros elementos, expresiones matemáticas [10].

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 Writter 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 mostrar la ayuda presente en todas las ventanas de la aplicación.

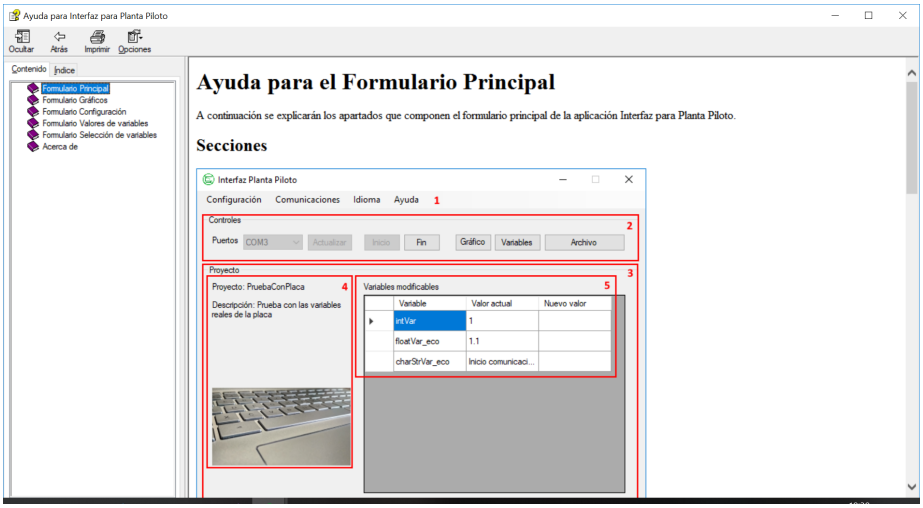

Figura 4.11: CHM File

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.

Figura 4.12: 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 un 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.

4.11. Base de datos

4.11.1. Microsoft SQL Server

Microsoft SQL Server es un sistema de manejo de bases del modelo relacional, diseñado por Microsoft [11].

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 LaTeXque 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 LaTeX.

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, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa Atom y en su lugar emplea el mismo componente editor ("Monaco") utilizado en Visual Studio Team Services (anteriormente llamado Visual Studio Online) [13].

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 periocidad 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 LaTeX.

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

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

Bibliografía

[1] *FRDMK64FUG, FRDM-K64F Freedom Module User's Guide, Rev. 0.1, 04/2014*, chapter 2 FRDM-K64F hardware overview. •, 2014.

[2] IBM. Comunicación serie, 2019. [Internet; descargado 31-enero-2019].

[3] María Isabel Revilla Izquierdo. "implementaciÓn de reguladores pids industriales y reguladores avanzados en microcontroladores". Master's thesis, UNIVERSIDAD DE BURGOS, 2018.

[4] Microsoft. Información general acerca de .net framework, 2018. [Internet; descargado 30-enero-2019].

[5] Microsoft. ¿qué es un patrón de diseño?, 2018. [Internet; descargado 11-enero-2019].

[6] Wikipedia. Git — wikipedia, la enciclopedia libre, 2018. [Internet; descargado 12-enero-2019].

[7] Wikipedia. Github — wikipedia, la enciclopedia libre, 2018. [Internet; descargado 13-enero-2019].

[8] Wikipedia. Scrum (desarrollo de software) — wikipedia, la enciclopedia libre, 2018. [Internet; descargado 11-enero-2019].

[9] Wikipedia. Técnica pomodoro — wikipedia, la enciclopedia libre, 2018. [Internet; descargado 11-enero-2019].

[10] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2019. [Internet; descargado 13-enero-2019].

[11] Wikipedia. Microsoft sql server — wikipedia, la enciclopedia libre, 2019. [Internet; descargado 28-enero-2019].

[12] Wikipedia. Microsoft visual studio — wikipedia, la enciclopedia libre, 2019. [Internet; descargado 13-enero-2019].

[13] Wikipedia. Visual studio code — wikipedia, la enciclopedia libre, 2019. [Internet; descargado 28-enero-2019].