



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 — 16 de enero  
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, 16 de enero de 2019

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. Daniel Sarabia Ortiz

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 conecataada 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. . .

### **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.



---

# Índice general

---

|   |     |
|---|-----|
| Índice general                                  | III |
| Índice de figuras                               | IV  |
| Índice de tablas                                | V   |
| Introducción                                    | 1   |
| Objetivos del proyecto                          | 5   |
| Conceptos teóricos                              | 7   |
| 3.1. Secciones . . . . .                        | 7   |
| 3.2. Referencias . . . . .                      | 7   |
| 3.3. Imágenes . . . . .                         | 8   |
| 3.4. Listas de ítems . . . . .                  | 8   |
| 3.5. Tablas . . . . .                           | 9   |
| Técnicas y herramientas                         | 11  |
| Aspectos relevantes del desarrollo del proyecto | 17  |
| Trabajos relacionados                           | 19  |
| Conclusiones y Líneas de trabajo futuras        | 21  |
| Bibliografía                                    | 23  |

---

# Índice de figuras

---

|  |    |
|--|----|
| 3.1. Autómata para una expresión vacía . . . . . | 8  |
| 4.2. ZenHub Overview . . . . .                   | 13 |
| 4.3. Codacy Dashboard . . . . .                  | 15 |

---

# Índice de tablas

---

|   |    |
|---|----|
| 3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto | 10 |
|---|----|



---

# 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 aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de  $\text{\LaTeX}$ <sup>1</sup>.

## 3.1. Secciones

Las secciones se incluyen con el comando `section`.

### Subsecciones

Además de secciones tenemos subsecciones.

### Subsubsecciones

Y subsecciones.

## 3.2. Referencias

Las referencias se incluyen en el texto usando `cite` [7]. Para citar webs, artículos o libros [1].

---

<sup>1</sup>Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

### 3.3. Imágenes

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



Figura 3.1: Autómata para una expresión vacía

### 3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

**Primer item** más información sobre el primer item.

**Segundo item** más información sobre el segundo item.

▪

## 3.5. Tablas

Igualmente se pueden usar los comandos específicos de  $\text{\LaTeX}$  o bien usar alguno de los comandos de la plantilla.

| 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

---

# 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 [5].

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, 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 [6].

## 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 [2].

## 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 [3].

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

**GitHub** es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git.[4]. 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.

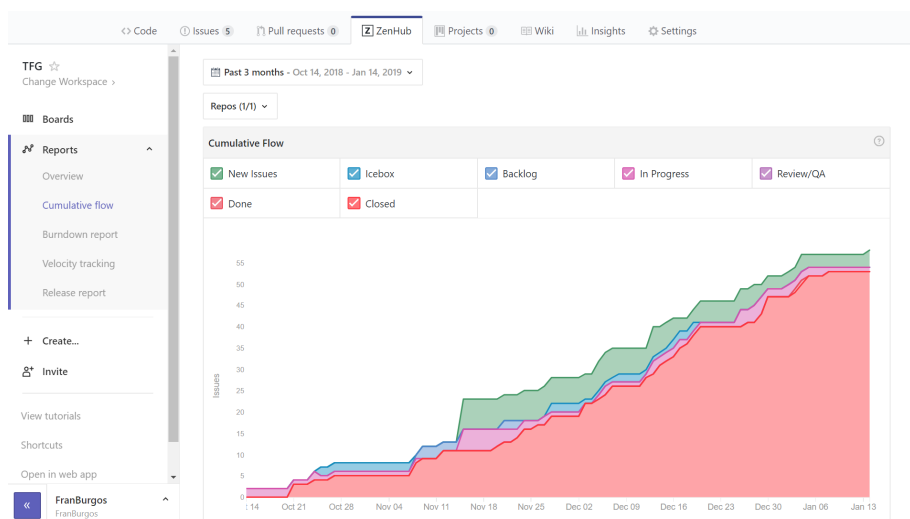


Figura 4.2: ZenHub Overview

En un principio también se barajó utilizar GitHub Project 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. Soporta múltiples lenguajes de programación, tales como F#, Python, C++, C#, Visual Basic .NET, Java, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades online bajo Windows Azure en forma del editor Monaco [8].

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#.

#### 4.6.2. Texmaker

**Texmaker** es un editor multiplataforma gratuito (licencia GNU GLP v3.0) para  $\text{\LaTeX}$  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  $\text{\LaTeX}$ .

### 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

$\text{\LaTeX}$  es 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 [7].

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.

## 4.9. 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 4.3: 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.



---

# 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. Inicio del proyecto

La elección de este proyecto se vio motivada tanto por el interés personal por desarrollar una aplicación que se comunicara con un dispositivo como por la posibilidad de ayudar a la Universidad de Burgos y a sus alumnos a tener una facilidad más con la que pudieran contar a la hora de llevar a cabo sus estudios y que no fuera un proyecto que queda olvidado.

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 C++ y por los conocimientos de los tutores en este lenguaje. 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, las herramientas y entornos de desarrollo que se iban a utilizar (ZenHub, Visual Studio 2017), y la periodicidad con la que se iban a realizar reuniones para tratar los avances del proyecto, siendo periodos de dos semanas.

### **5.1. Formación**

El proyecto requería de unos conocimientos de los que no se disponía originalmente, por lo que hubo que solventar estas carencias.

---

## 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] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [2] Microsoft. ¿qué es un patrón de diseño, 2018. [Internet; descargado 11-enero-2019].
- [3] Wikipedia. Git — wikipedia, la enciclopedia libre, 2018. [Internet; descargado 12-enero-2019].
- [4] Wikipedia. Github — wikipedia, la enciclopedia libre, 2018. [Internet; descargado 13-enero-2019].
- [5] Wikipedia. Scrum (desarrollo de software) — wikipedia, la enciclopedia libre, 2018. [Internet; descargado 11-enero-2019].
- [6] Wikipedia. Técnica pomodoro — wikipedia, la enciclopedia libre, 2018. [Internet; descargado 11-enero-2019].
- [7] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2019. [Internet; descargado 13-enero-2019].
- [8] Wikipedia. Microsoft visual studio — wikipedia, la enciclopedia libre, 2019. [Internet; descargado 13-enero-2019].