

TFG del Grado en Ingenier´ıa Inform´atica

Mejora de interfaz para planta piloto y nueva funcionalidad como aplicación web



Presentado por Iván Cortés

en Universidad de Burgos — 18 de junio de 2022 Tutores: Dr. Alejandro Merino Gómez - Dr.

Daniel Sarabia Ortiz

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.

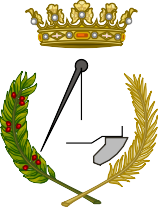
Exponen:

Que el alumno D. Iván Cortés Aliende, con DNI 71706785F, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado “Mejora de interfaz para planta piloto y nueva funcionalidad como aplicación web”.

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, 18 de junio de 2022

Vo. Bo. del Tutor:



D. Daniel Sarabia Ortiz

Vo. Bo. del co-tutor:

D. Alejandro Merino Gómez

**Resumen**

El objetivo de este proyecto es la mejora de la interfaz desarrollada en 2019 en el TFG “Desarrollo de una interfaz para planta piloto” por Francisco Crespo Diez, la cual permite al usuario manejar una placa Freescale FRDMK64F que, a su vez, controla una planta piloto.

Las mejoras se centran en el arreglo de errores detectados, la implementación de nuevas características solicitadas por los usuarios y la implementación de una versión de dicha interfaz en versión de aplicación web.

**Descriptores**

Aplicación web, solución de errores, .NET, ASP, C#, interfaz, SCADA

**Abstract**

The goal of this project is to improve the interface developed in 2019 in the TFG “Desarrollo de una interfaz para planta piloto” by Francisco Crespo Diez, which allows the user to manage a Freescale FRDMK64F board that controls a pilot plant.

The improvements focus on the fixing detected errors, the implementation of new features requested by the useres and the implementation of a web application based version.

**Keywords**

Web application, error fixing, .NET, ASP, C#, interface, SCADA.

# ´Indice general

|  |  |
| --- | --- |
| [**´****Indice general**](#_bookmark0)  [**´Indice de figuras**](#_bookmark1) | **III**  **IV** |
| [**Introducci´on**](#_bookmark2) | **1** |
| [**Objetivos del proyecto**](#_bookmark3) | **2** |
| [**Conceptos te´oricos**](#_bookmark4) | **3** |
| [3.1. Secciones](#_bookmark5) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 3 |
| [3.2. Referencias](#_bookmark6) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 3 |
| [3.3. Im´agenes](#_bookmark7) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 3 |
| [3.4. Listas de items](#_bookmark10) . . . . . . . . . . . . . . . . . . . . . . . . . . . | 4 |
| [3.5. Tablas](#_bookmark11) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 5 |
| [**T´ecnicas y herramientas**](#_bookmark12) | **6** |
| [**Aspectos relevantes del desarrollo del proyecto**](#_bookmark13) | **7** |
| [**Trabajos relacionados**](#_bookmark14) | **8** |
| [**Conclusiones y L´ıneas de trabajo futuras**](#_bookmark15) | **9** |
| [**Bibliograf´ıa**](#_bookmark16) | **10** |

# ´Indice de figuras

[3.1. Aut´omata para una expresi´on vac´ıa](#_bookmark9)

# Introducción

En 2019 el alumno de la Universidad de Burgos Francisco Crespo Diez desarrolló como Trabajo Fin de Grado una interfaz que permitía a los alumnos comunicarse con las placas NXP FRDM-K64F de una forma mucho más cómoda e intuitiva, permitiendo que estos se pudieran centran en aprender en vez de gastar tiempo con métodos de comunicación mucho más complicados [1].

Dicha interfaz está siendo utilizada hoy en día por alumnos de distintas titulaciones para poder comunicarse con las placas, cambiando las variables que estas manejan y estudiando como estos cambios afectan al caudal de aire que llega al sensor de la planta piloto. Esto a permitido tanto a alumnos como a profesores detectar posibles mejoras que la interfaz podría implementar y fallos que se podrían arreglar, todo con el fin de mejorar la experiencia de usuario.

La principal mejora de todas las implementadas ha sido el desarrollo de una aplicación web la cual permite a los usuarios comunicarse con la placa con una interfaz similar a la que ya se encontraba presente en el proyecto original, pero con todas las ventajas que supone desplegarlo en un entorno web.

* 1. Estructura de la memoria

Estructura principal de este documento.

* **Introducción:** Descripción y contexto del proyecto realizado, estructura de los documentos presentados y qué recursos han sido adjuntados.
* **Objetivos del proyecto:** Objetivos que se buscaban cumplir con el desarrollo de este proyecto
* **Conceptos teóricos**: Marco y contexto teórico del proyecto.
* **Técnicas y herramientas**: Utilidades y procedimientos que se han seguido para la realización del proyecto.
* **Aspectos relevantes del desarrollo del proyecto**: Cómo ha sido llevado a cabo el proyecto.
* **Trabajos relacionados**: Proyectos similares al presentado y el proyecto de partida.
* **Conclusiones y Líneas de trabajo futuras:** Deducciones a las que se han llegado tras la realización del proyecto y posibles futuras mejoras y/o funcionalidades a implementar.
* **Bibliografía:** Procedencia de los conocimientos retratados en este documento.
  1. Recursos adjuntados

# Objetivos del proyecto

En este apartado se detallarán, como ya hemos indicado, los objetivos que se planeaban cumplir con el desarrollo de este proyecto.

1. Corrección de los errores detectados por los usuarios tras el uso de la aplicación.
2. Implementación de nuevas funcionalidades sugeridas por usuarios de la aplicación.
3. Desarrollo de una nueva interfaz en forma de aplicación web mediante la cual los usuarios también puedan comunicarse con la placa.
4. Cambio del sistema gestor de bases de datos que utilizaba la aplicación por uno que se cree y se destruya con el uso de la aplicación.

2.1. Requisitos técnicos a tener en cuenta

Hay una serie de requisitos técnicos que se han debido tener en cuenta a la hora de cumplir los objetivos ya definidos:

1. Se debe partir de la aplicación inicial ya desarrollada.
2. Se deben mantener las compatibilidades con máquinas Windows y con el método de conexión en serie a la placa.
3. Se debe de utilizar un sistema de control de versiones.
4. Se debe mantener .NET Framework como base de la aplicación.
5. Las interfaces deben mantenerse lo más intuitivas posible, manteniendo así el objetivo principal del proyecto original, facilitar a los alumnos su aprendizaje.

**Conceptos teóricos**

* 1. .NET Framework

Es un entorno de ejecución de aplicaciones administrado para el sistema operativo Windows que proporciona servicios a las aplicaciones que se están ejecutando. Se basa en dos componentes principales:

* **Common Language Runtime (CLR):** Es el motor de ejecución de las aplicaciones. Se encarga de controlar las aplicaciones que se encuentran en ejecución en el sistema.
* **Biblioteca de clases:** Se encarga de dotar a los desarrolladores de una biblioteca compuesta por código reusable y altamente probado para que estos la puedan utilizar a la hora de desarrollar sus proyectos en .NET Framework.

.NET Framework ofrece los siguientes servicios a las aplicaciones en ejecución:

* **Administración de la memoria.** CLR se encarga de asignar y liberar la memoria, así como de gestionar la vida útil de los objetos instanciados en la aplicación, de manera que los desarrolladores no tengan que ocuparse de ello.
* **Sistema de tipos comunes:** .NET Framework se encarga de definir los tipos básicos. Esto nos permite que todos los lenguajes destinados a ser utilizados junto con .NET Framework tengan estos tipos comunes, haciéndolos interoperables.
* **Biblioteca de clases extensa:** Los desarrolladores pueden utilizar la biblioteca de clases de .NET Framework para ahorrarse el tener que programar grandes cantidades de código de bajo nivel.
* **Marcos y tecnologías de desarrollo:** .NET Framework dispone de una serie de bibliotecas especializadas en el desarrollo de determinados tipos de aplicaciones. Por ejemplo, ASP.NET para el desarrollo de aplicaciones web.
* **Interoperabilidad de lenguajes:** Continuando con lo explicado en el apartado de Sistemas de tipos comunes, los compiladores de los lenguajes de programación destinados a .NET Framework producen el denominado Lenguaje Intermedio Común (CIL). Este lenguaje es compilado en pleno tiempo de ejecución por el CLR, de manera que las rutinas programadas en un determinado lenguaje son accesibles para el resto de los lenguajes.
* **Compatibilidad de versiones:** Las aplicaciones desarrolladas para una determinada versión de .NET Framework serán compatibles para versiones posteriores. Esto se cumple en la mayoría de las situaciones, pero se pueden dar extrañas excepciones.
* **Ejecución en paralelo:** .NET Framework permite que coexistan varias versiones de CLR en el mismo equipo, de manera que también puedan coexistir varias versiones de las mismas aplicaciones, permitiendo que cada una se ejecute en la versión de .NET Framework en la que se compiló.
* **Compatibilidad con múltiples versiones:** Si los desarrolladores de bibliotecas establecían como destino .NET Stardard (que es la especificación formal de las API de .NET [2]) sus bibliotecas podían ser utilizadas en varias plataformas de .NET Framework.

[3]

Dichos servicios son proporcionados por .NET Framework con la intención de cumplir los siguientes objetivos:

* Dotar al desarrollador de un entorno de programación orientada a objetos (POO) coherente en el cual el código fuente de los objetos se pueda:
  + Almacenar y ejecutarse de forma remota.
  + Almacenar de forma distribuida en Internet, pero ejecutarse de forma local
  + Almacenar y ejecutar localmente.
* Dotar al usuario de un entorno de ejecución de código el cual:
  + No cuente con los problemas de rendimiento presentes en otros entornos, como los entornos con scripts o interpretados.
  + Cuente con los menores conflictos de software y de control de versiones posibles.
  + Promueva la ejecución segura de código
* Proporcionar al desarrollador de una experiencia muy similar entre todos los tipos de aplicaciones que pueden ser desarrollados con .NET Framework.
* Basar todos los protocolos de comunicaciones en estándares ya definidos, de tal forma que sea sencillo integrar el código desarrollado con .NET Framework con otro distinto.

[4]

.NET Framework fue el entorno de ejecución para el que Francisco Crespo Diez desarrolló la primera versión de la aplicación en su día. Por lo tanto, ha sido el empleado también durante el desarrollo de este proyecto de mejora e implementación de nuevas funcionalidades.

* 1. ASP.NET

Es un marco web pensado para la creación de páginas y aplicaciones web mediante la utilización de los lenguajes HTML, CSS y JavaScript. También es utilizado para la utilización de tecnologías de tiempo real y la creación de APIs web.

ASP.NET nos ofrece tres marcos distintos para la creación de aplicaciones web. Todos ellos cuentan con la totalidad de las ventajas y las características de ASP.NET:

* **Web Forms (ASP.NET Web Forms)**: Permite la creación de aplicaciones web dinámicas mediante un modelo controlado por eventos. Está pensado para la creación de aplicaciones web centradas en la utilización de datos y en el manejo mediante una interfaz de usuario.
* **MVC (ASP.NET MVC)**: Como su propio nombre indica, se basa en el patrón de arquitectura Modelo-Vista-Controlador. Permite compilar espacios web dinámicos basándose en modelos, de forma que se consigue una separación muy marcada de intereses y proporciona un control absoluto del marcado.
* **Web Pages (ASP.NET Web Pages)**: Permite la combinación del código backend del servidor con el lenguaje de marcado HTML para la creación de espacios web dinámicos utilizando la sintaxis de Razor.

Estos tres marcos web que proporciona ASP.NET se basan en .NET Framework, compartiendo la funcionalidad principal de .NET y ASP.NET. Estos marcos funcionan de manera independiente, pero pueden existir conjuntamente dentro de la misma aplicación, siendo frecuente encontrar su uso combinado en bastantes proyectos. [5]

ASP.NET ha sido el marco de desarrollo web empleado para la creación de la aplicación web de este proyecto, en concreto, el marco de ASP.NET Web Forms.

* 1. C#

Es un lenguaje de programación proveniente en el lenguaje C y basado la utilización de objetos y cuenta con seguridad de tipos. Está pensado para el desarrollo de aplicaciones pensadas para ser ejecutadas en .NET.

Se trata de un lenguaje orientado a objetos orientados a componentes. Permite a los desarrolladores crear y usar componentes software mediante las construcciones de lenguaje que el propio lenguaje nos proporciona.

Características principales del lenguaje:

* Recolección de elementos no utilizados. Se encarga de demandar de forma automática la memoria ocupada por objetos no utilizados o inaccesibles.
* Tipos que aceptan valores NULL.
* Control de excepciones.
* Admisión de expresiones lambda.
* Language Integrated Query (LINQ). Patrón común que permite al lenguaje operar con datos independientemente de cuál sea su origen.
* Compatible con operaciones asíncronas.
* Implementa un sistema de tipos unificado
* Admite tipos de referencia que el usuario haya definido y tipos de valor.

[6]

C# ha sido el lenguaje utilizado para programar la parte lógica tanto de la aplicación local pensada para entornos Windows con .NET Framework como para la versión web en ASP.NET.

* 1. IIS Express

IIS Express es una versión ligera y portable del servidor web de Microsoft Internet Information Server (IIS). Cuenta con todas las capacidades de IIS 7 y posteriores, así como las siguientes características propias:

* No requiere de permisos de administrador para realizar sus tareas.
* Es capaz de trabajar con aplicaciones ASP.NET y PHP sin necesidad de instalar ningún complemento del servidor ni realizar ninguna configuración.
* Soporta a varios usuarios trabajando en el mismo equipo de forma simultánea.

Además, se trata del servidor web que viene incluido con el entorno de desarrollo de Microsoft Visual Studio, y es el que incorpora para probar aplicaciones web antes de su lanzamiento a producción. [7]

IIS Express ha sido el servidor web para el que se ha ideado el panel de lanzamiento de la aplicación web.

* 1. Base de datos

Una base de datos es un conjunto de información estructurada que es almacenada en un sistema informático. Es controlada por un Sistema de Gestión de Bases de Datos (DBMS).

Lo más habitual en los sistemas de bases de datos actuales es que la información se estructure y se almacene en forma de tablas, mediante la utilización de filas y columnas (bases de datos relacionales), y que se lea, se escriba, se edite y se elimine dicha información mediante la utilización de un Lenguaje de Consulta Estructurada (SQL)

Existe un inmenso número de tipos de bases de datos. Algunos de ellos son los siguientes:

* **Bases de datos relacionales:** Sus elementos se organizan en un conjunto de tablas, mediante filas y columnas. Su tecnología aporta la manera más eficiente y flexible de poder acceder a la información estructurada. Este tipo de base de datos ha sido el empleado en el proyecto para almacenar los datos de la placa y de la aplicación web.
* **Bases de datos orientadas a objetos:** Como su propio nombre indica, la información es estructurada en forma de objetos.
* **Bases de datos distribuidas:** Las que se encuentran en sistemas distribuidos, es decir, divididas entre distintos equipos.
* **Almacenes de datos:** Diseñados para consultas y análisis rápidos.
* **Bases de datos no relacional:** Permite almacenar y manejar información que no ha sido estructurada o información semiestructurada.
* **Bases de datos orientadas a grafos:** Almacena y estructura la información en nodos y relaciones entre nodos.
* **Bases de datos OLTP:** Pensada para que un gran número de usuarios realicen un gran número de transacciones, debido a ser una base de datos veloz y analítica.

Sistema de Gestión de Bases de Datos (DBMS)

Software que sirve como interfaz entre la base de datos y los usuarios o aplicaciones finales, de forma que estos pueden gestionar cómo se organiza la información.

También permite un gran número de operaciones de índole administrativa, ya que nos facilita el control y la monitorización de las bases de datos.

[8]

* 1. Planta Piloto

Este proyecto parte de mejorar la interfaz que controla las plantas piloto empleadas en las asignaturas de los grados de Electrónica Industrial y Automática y el máster de Ingeniería Industrial. Estas plantas piloto fueron diseñadas en un principio por Rubén Zambrana Rodríguez para su Trabajo Fin de Grado [9] y mejoradas en una segunda versión por María Isabel Revilla Izquierdo [10].

La planta piloto está conformada por los siguientes componentes:

* Una planta, forma da a su vez por un ventilador, una resistencia, una sonda y un caudalímetro.
* Una fuente de alimentación.
* Un conjunto de etapas de acondicionamiento de señales.
* Una placa FRDM-K64F que controla el sistema y se encarga de servir de intermediario entre la interfaz y la plata mediante intercambio de mensajes.

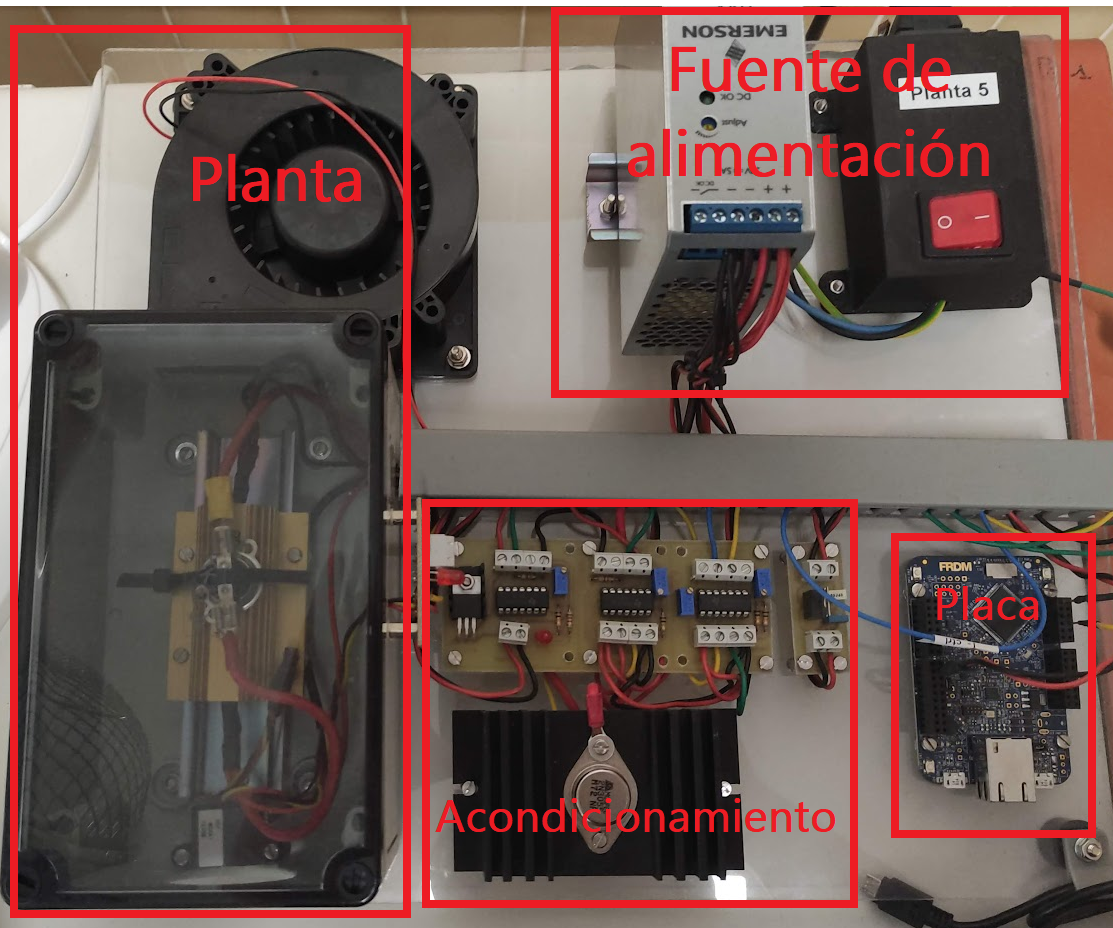


Figura 3.1: Partes de la planta piloto.

Lo más interesante de la planta piloto en cuanto a este proyecto se refiere es la placa FRDM-K64F, ya que es la que va a interactuar con la aplicación. La placa se conecta por USB a un equipo con el sistema operativo Windows y con la aplicación ejecutándose. Una vez se ha establecido la conexión, el programa es el encargado de ir recogiendo periódicamente los datos que envía la placa mediante el puerto USB y la conexión en serie para almacenarlos en una tabla de una base de datos. Además, desde la interfaz también se le pueden mandar mensajes a la placa para que cambie los valores de sus variables internas, de forma que podamos estudiar como afectan los cambios al flujo de aire.

* 1. Comunicación en Serie

En la comunicación en serie los datos se envían del dispositivo emisor al dispositivo receptor de forma secuencial bit a bit, por ello, solo necesitan un único cable entre ambos. Los pasos que se siguen son los siguientes:

1. Los datos se transforman de formato paralelo a formato secuencial.
2. Los bits se organizan uno tras otro, formando una serie.
3. Se envían los bits al dispositivo receptor, siendo el primero en ser enviado el bit menos significativo.
4. El dispositivo receptor recibe los datos y los vuelve a transformar a formato paralelo.

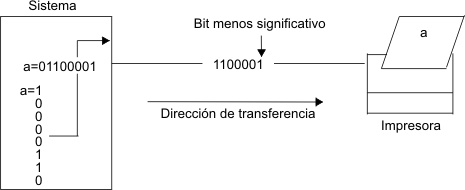


Figura 3.2: Comunicación en serie.

Pese a la simplicidad de este protocolo de comunicaciones, podemos encontrarnos con un importante problema, que el dispositivo receptor no sepa dónde finaliza un carácter y empieza el siguiente. Este problema surge cuando el mensaje está conformado por un gran número de caracteres. Para poder poner solución a este problema, tanto el dispositivo emisor como el dispositivo receptor deben estar temporizados o sincronizados. En la Figura 3.2 se puede observar este problema. [11]



Figura 3.3: Comunicación en serie con un gran número de bits.

La comunicación en serie se aplica en este proyecto para la comunicación entre la placa controladora de la planta piloto y el equipo Windows que recibe y le envía datos. La comunicación ha sido bidireccional.

**Técnicas y herramientas**

Estas son las técnicas y herramientas que se han empleado a la hora del desarrollo del proyecto, su documentación, y la comunicación con los tutores del proyecto:

* 1. GitHub

[GitHub](https://github.com) es la plataforma más conocida y utilizada para el control de versiones y el desarrollo de aplicaciones de forma colaborativa a día de hoy. Emplea el sistema Git para permitir a sus usuarios la creación de repositorios donde ellos y sus compañeros pueden participar.

Ha sido la herramienta empleada en cuanto a control de versiones se refiere. Como Francisco Crespo Diez ya creó un repositorio para el desarrollo de su primera versión en 2019, se ha partido de un fork de dicho repositorio.

Pantalla de computadora

Descripción generada automáticamente con confianza media

Figura 4.4: Repositorio del proyecto en GitHub.

* 1. GitKraken

[GitKraken](https://www.gitkraken.com) es un cliente multiplataforma que nos permite un manejo y una gestión de los repositorios de GitHub mediante el uso de una interfaz muy visual y sencilla.

Ha sido el cliente empleado para el control del repositorio en GitHub del proyecto debido a que, mediante su intuitiva y personalizable interfaz (Figura 4.5), la comunicación con GitHub y el registro de cambios han sido mucho más rápidos que mediante el uso de la consola de comandos.

Captura de pantalla de computadora

Descripción generada automáticamente

Figura 4.5: Interfaz de GitKraken con el repositorio GitHub del proyecto abierto.

* 1. Microsoft Visual Studio 2019

[Microsoft Visual Studio 2019](https://visualstudio.microsoft.com/es/) se trata de un entorno de desarrollo integrado multiplataforma desarrollado por Microsoft y enfocado en el desarrollo de aplicaciones .NET y C++, dotado de multitud de herramientas y documentación para ello.

Se trata del entorno de desarrollo integrado principal de todo el proyecto debido a que, al ser las principales bases del proyecto .NET Framework, ASP.NET y C++, era la opción más segura y recomendable.

Captura de pantalla de computadora

Descripción generada automáticamente

Figura 4.6: Microsoft Visual Studio con el proyecto abierto.

* 1. Microsoft Visual Studio Code

[Microsoft Visual Studio Code](https://code.visualstudio.com) es un editor de código fuente multiplataforma desarrollado por Microsoft muy versátil. Esto es debido a que mediante la instalación de distintas extensiones es posible trabajar con la mayoría de lenguajes de programación de una forma intuitiva, cómoda y rápida.

Ha sido empleado en momentos puntuales para edición de ficheros HTML, XML y TXT.

* 1. Microsoft Word

[Microsoft Word](https://www.microsoft.com/es-es/microsoft-365/word) es un software de procesamiento de textos desarrollado por Microsoft. Por defecto viene incluido en el paquete de software [Microsoft Office](https://www.office.com).

Se trata de la herramienta empleada para la redacción de la documentación del proyecto. Al principio se pensó en emplear el sistema de composición de textos [LATEX](https://www.latex-project.org), pero se acabó descartando esa opción por la poca experiencia que tengo en esa tecnología, lo cual iba a acabar afectando a la velocidad del desarrollo. La documentación ha sido desarrollada empleando la licencia de [Microsoft 365](https://www.microsoft.com/es-es/microsoft-365) (la cual incluye Microsoft Office y, por tanto, Microsoft Word) otorgada por la Universidad de Burgos a sus alumnos.

* 1. Microsoft SharePoint

[Microsoft SharePoint](https://www.microsoft.com/es-es/microsoft-365/sharepoint/collaboration) es una plataforma virtual de colaboración empresarial e institucional desarrollada por Microsoft e incluida en la licencia de Microsoft 365 proporcionada por la Universidad de Burgos.

Ha sido empleada como una plataforma de compartición de ficheros entre los tutores de este proyecto y yo mismo, de tal forma que todos pudiéramos estar al día de la situación del proyecto.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Figura 4.7: Carpeta del proyecto en el SharePoint de la UBU.

* 1. Microsoft Teams

[Microsoft Teams](https://www.microsoft.com/es-es/microsoft-teams/group-chat-software) es una plataforma virtual desarrollada por Microsoft y de índole empresarial e institucional con el objetivo de facilitar la colaboración del trabajo incorporando características como videollamadas, compartición de ficheros, chat de texto, etc.

Se trata de la plataforma a través de la cual han tenido lugar las distintas reuniones con los tutores del proyecto.

* 1. IIS Express

Como ya se explicó en el punto de Conceptos Teóricos, [IIS Express](https://www.microsoft.com/es-es/download/details.aspx?id=48264) es un servidor web desarrollado por Microsoft pensado para mantener en línea páginas y aplicaciones web desarrolladas en ASP.NET o PHP. Se caracteriza por ser portable, ligero y consumir muy pocos recursos.

Ha sido empleado como servidor de pruebas durante el desarrollo de la aplicación web en ASP.NET, ya que tiene integración directa con el entorno de desarrollo Microsoft Visual Studio 2019.

Además, también ha sido el servidor web para el que se ha desarrollado la integración y automatización del lanzamiento de la aplicación web en la interfaz original (Figura 4.7).

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Figura 4.8: Pantalla de lanzamiento del servidor web mediante la integración de IIS Express.

En un principio se pensó en lanzar la aplicación web en un contenedor de [Docker](https://www.docker.com) una vez su desarrollo hubiera finalizado. El problema fue que la [imagen oficial de Microsoft](https://hub.docker.com/_/microsoft-dotnet-framework-aspnet/) para aplicaciones web ASP.NET tenía un tamaño excesivo, de unos 8GB, lo cual ralentizaría enormemente el flujo del trabajo en los ordenadores de los laboratorios de la universidad, por lo que las prácticas de los estudiantes con la planta piloto se verían afectadas. Además, realizando pruebas se tuvieron problemas a la hora de instalar Docker en entornos Windows y automatizar el lanzamiento del contenedor (problemas de permisos, tiempos de espera muy elevados, alto consumo de recursos, etc.). Por todo ello, está opción se descartó y se optó por IIS Express por ser ligero, rápido y consumir muy pocos recursos.

* 1. Open Broadcaster Software

[Open Broadcaster Software (OBS Studio)](https://obsproject.com/es) es un software de grabación y retransmisión en directo de código abierto y multiplataforma. Se caracteriza por consumir pocos recursos, permitir la gestión de multitud de dispositivos de entrada de vídeo y audio, creación y manejo de varias escenas durante la grabación y por ser altamente personalizable a la hora de establecer cómo procesar y codificar el vídeo.

Este software ha sido empleado para la grabación de distintos vídeos mostrando el estado del desarrollo del proyecto, los problemas solucionados y las nuevas funcionalidades implementadas, con el objetivo de informar a los tutores del estado del proyecto. También ha sido utilizado para la grabación de los vídeos de muestra de la aplicación adjuntos.

* 1. Lightshot

Se trata de un sencillo software multiplataforma de realización de capturas de pantalla desarrollado por Skillbrains. Una vez presionado un determinado atajo de teclado, sobrepone en el sistema una interfaz mediante la cual se puede seleccionar un área de la pantalla donde realizar sencillas modificaciones (agregar textos, recuadros, flechas, etc.) y, al finalizar, guardar el área seleccionada y sus modificaciones como archivo de imagen.

Este software ha sido empleado a la hora de realizar capturas de la pantalla para el desarrollo de la documentación del proyecto, para la redacción del manual de usuario y para los archivos de ayuda de la aplicación.

* 1. Mockplus RP

[Mockplus RP](https://www.mockplus.com/mockplus-rp) es un editor visual que permite, mediante un sistema muy sencillo de arrastrar y soltar, diseñar interfaces para aplicaciones de cualquier ámbito (escritorio, web, móvil, etc.). Se trata de una aplicación web basada en la computación en la nube con un plan gratuito y otro de pago.

Ha sido utilizado para el diseño de las distintas interfaces de la aplicación web antes incluso de empezar su desarrollo, de tal forma que al empezar a desarrollarla ya estaban claros los distintos campos que la conformaban y su estructura, lo cual hizo más ágil este proceso. Pese a que también soporte exportar las interfaces diseñadas directamente a HTML y CSS para su maquetado, este fue realizado a mano en Microsoft Visual Studio 2019, por limitaciones del plan gratuito y para poder asegurarme de que el código fuera lo más limpio y óptimo posible.

Captura de pantalla de computadora

Descripción generada automáticamente

Figura 4.9: El editor de Mockplus RP con el proyecto de la interfaz principal de la aplicación web.

* 1. P&E Driver

[P&E Driver](https://www.pemicro.com/opensda/) es un software desarrollado por P&E Microcomputer Systems que otorga a un entorno Windows de la capacidad de emular una comunicación en serie [RS232](https://es.wikipedia.org/wiki/RS-232) en un puerto [UART](https://es.wikipedia.org/wiki/Universal_Asynchronous_Receiver-Transmitter).

Este software es el que otorga la capacidad de conectarse a la placa FRDM-K64F de la planta piloto a los entornos Windows en los que los estudiantes realizan las prácticas. Mediante este software, el sistema operativo es capaz de leer los valores que le llegan de la placa y de mandare mensajes él mismo. Por lo tanto, ha sido esencial a la hora del desarrollo del proyecto, pues sin él la comunicación con la placa no hubiera sido posible. Fue proporcionado por los tutores del proyecto al comenzar con el desarrollo de este.

* 1. Katalon Recorder

[Katalon Recorder](https://katalon.com/katalon-recorder-ide/) es una extensión de navegador pensada para la automatización de pruebas en aplicaciones web. Se compone de una interfaz muy simple que permite la grabación de acciones del usuario para más tarde replicarlas de forma automática. Es compatible con Selenium IDE.

Ha sido la herramienta empleada para la realización de pruebas automáticas de entrada de datos en la aplicación web.

* 1. HTML Help Workshop

HTML Help Workshop es una aplicación pensada para compilar archivos HTML en proyectos CHM (Archivo de Ayuda de HTML Compilado), que son los utilizados por aplicaciones desarrolladas para entornos Windows para mostrar los menús de ayuda.

Como es de suponer, ha sido la herramienta utilizada para la actualización de los menús de ayuda con las nuevas funcionalidades implementadas.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura 4.10: Menú de ayuda compilado en formato CHM.

* 1. Modelo Vista Controlador (MVC)

Se trata de un modelo de arquitectura software que separa en tres capas los distintos componentes de una aplicación:

* Modelo: la capa relativa al tratamiento de los datos.
* Vista: hace referencia a la interfaz del usuario.
* Controlador: intermediario entre la vista y el modelo.

[12]

Francisco Crespo Diez siguió este modelo arquitectónico durante el desarrollo de la primera versión de la interfaz para la planta piloto, por lo cual se ha seguido empleando durante esta revisión a fin de seguir un procedimiento coherente y óptimo.

* 1. Sonarqube

# Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos m´as interesantes del desarro- llo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposici´on del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de an´alisis, disen˜o e implementaci´on. Se busca que no sea una mera operaci´on de copiar y pegar diagramas y extractos del c´odigo fuente, sino que realmente se justifiquen los caminos de soluci´on que se han tomado, especial- mente aquellos que no sean triviales. Puede ser el lugar m´as adecuado para documentar los aspectos m´as interesantes del disen˜o y de la implementaci´on, con un mayor hincapi´e en aspectos tales como el tipo de arquitectura elegido, los ´ındices de las tablas de la base de datos, normalizaci´on y desnormaliza- ci´on, distribuci´on en ficheros3, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el [WWW...](http://WWW./) Este apartado, debe convertirse en el resumen de la experiencia pr´actica del proyecto, y por s´ı mismo justifica que la memoria se convierta en un documento u´til, fuente de referencia para los autores, los tutores y futuros alumnos.

7

# Trabajos relacionados

Este apartado ser´ıa 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 pequen˜o resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

8

# Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. E´stas pueden ser de diferente ´ındole, dependiendo de la tipolog´ıa del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacio- nadas con los resultados del proyecto y un conjunto de conclusiones t´ecnicas. Adem´as, resulta muy u´til realizar un informe cr´ıtico indicando c´omo se puede mejorar el proyecto, o c´omo se puede continuar trabajando en la l´ınea del proyecto realizado.

9

# Bibliografía

1. F.C. Diez, “Desarrollo de una interfaz para planta piloto,” tech.rep., Grado en Ingeniería Informática - Universidad de Burgos, 2019.
2. Microsoft, “.NET Standard,” 2022. [Internet; descargado 19-junio-2022].
3. Microsoft, “Introducción a .NET Framework,” 2022. [Internet; descargado 19-junio-2022].
4. Microsoft, “Información general acerca de .NET Framework,” 2022. [Internet; descargado 19-junio-2022].
5. Microsoft, “Información general de ASP.NET,” 2022. [Internet; descargado 19-junio-2022].
6. Microsoft, “Paseo por el lenguaje C#,” 2022. [Internet; descargado 19-junio-2022].
7. Microsoft, “IIS Express Overview,” 2022. [Internet; descargado 19-junio-2022].
8. Oracle, “¿Qué es una base de datos?,” 2019. [Internet; descargado 19-junio-2022].
9. 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.
10. 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.
11. IBM, “Comunicación serie,” 2021. [Internet; descargado 20-junio-2022].
12. Universitat d’Alacant, “Modelo Vista Controlador (MVC),” 2013. [Internet; descargado 24-junio-2022].