UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO

TÍTULO

Desarrollo de un prototipo de software HMI para control y monitoreo de sistemas de automatización industrial en tiempo real con PLCs SIEMENS, adaptable a los proyectos de la empresa INGEL-PRO

AUTOR

Naranjo Andrade, Edison Geovanny

TRABAJO DE TITULACIÓN

Previo a la obtención del grado académico en
MAGÍSTER EN ELECTRÓNICA Y AUTOMATIZACIÓN

TUTOR

Chuquimarca Jiménez, Luis Enrique

Santa Elena, Ecuador

Año 2026

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO

TRIBUNAL DE SUSTENTACIÓN

Firmado electrónicamente por:
ALICIA GERMANIA ANDRADE VERA
Validar únicamente con FirmaEC

Ing. Alicia Andrade Vera, Mgtr.
COORDINADORA DEL PROGRAMA

Firmado electrónicamente por:
LUIS ENRIQUE CHUQUIMARCA JIMENEZ
Validar únicamente con FirmaEC

Ing. Luis Chuquimarca Jiménez, Ph.D.
TUTOR

Firmado electrónicamente por:
OSCAR WLADIMIR GOMEZ MORALES
Validar únicamente con FirmaEC

Ing. Oscar Gómez Morales, Ph.D.
DOCENTE ESPECIALISTA

Firmado electrónicamente por:
JUNIOR RAFAEL FIGUEROA OLMEDO
Validar únicamente con FirmaEC

Ing. Junior Figueroa Olmedo, Mgtr.
DOCENTE ESPECIALISTA

Firmado electrónicamente por:
MARIA MARGARITA RIVERA GONZALEZ
Validar únicamente con FirmaEC

Abg. María Rivera González, Mgtr.
SECRETARIA GENERAL
UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por EDISON GEOVANNY NARANJO ANDRADE, como requerimiento para la obtención del título de Magíster en Electrónica y Automatización.

TUTOR

Firmado electrónicamente por:
LUIS ENRIQUE CHUQUIMARCA JIMENEZ
Validar únicamente con FirmaEC

Ing. Luis Chuquimarca Jiménez, Ph.D.

Santa Elena, 4 de febrero de 2026

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO
DECLARACIÓN DE RESPONSABILIDAD

Yo, **NARANJO ANDRADE EDISON GEOVANNY**

DECLARO QUE:

El trabajo de Titulación, Desarrollo de un prototipo de software HMI para control y monitoreo de sistemas de automatización industrial en tiempo real con PLCs SIEMENS, adaptable a los proyectos de la empresa INGEL-PRO; previo a la obtención del título en Magíster en Electrónica y Automatización, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Santa Elena, 4 de febrero de 2026

EL AUTOR

Firmado electrónicamente por:
EDISON GEOVANNY
NARANJO ANDRADE
Validar únicamente con FirmaEC

Edison Geovanny Naranjo Andrade

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO
CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Desarrollo de un prototipo de software HMI para control y monitoreo de sistemas de automatización industrial en tiempo real con PLCs SIEMENS, adaptable a los proyectos de la empresa INGEL-PRO, presentado por el estudiante, NARANJO ANDRADE EDISON GEOVANNY fue enviado al Sistema Antiplagio COMPILATIO, presentando un porcentaje de similitud correspondiente al 3%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.


CERTIFICADO DE ANÁLISIS
magister

Geovanny Naranjo rev 1 (1)

3% Textos sospechosos

3% Similitudes
< 1 % similitudes entre comillas
2 % entre las fuentes mencionadas
0% Idiomas no reconocidos
12% Textos potencialmente generados por la IA (ignorado)

Nombre del documento: Geovanny Naranjo rev 1 (1).pdf
ID del documento: 21960472834fc35d73693dd39bdd240d1e2b38dd
Tamaño del documento original: 3,88 MB

Depositante: LUIS ENRIQUE CHUQUIMARCA JIMENEZ
Fecha de depósito: 6/2/2026
Tipo de carga: interface
fecha de fin de análisis: 6/2/2026

Número de palabras: 13.800
Número de caracteres: 120.245

TUTOR

Firmado electrónicamente por:
LUIS ENRIQUE CHUQUIMARCA JIMENEZ
Validar únicamente con FirmaEC

Ing. Luis Chuquimarca Jiménez, Ph.D.

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO
AUTORIZACIÓN

Yo, **EDISON GEOVANNY NARANJO ANDRADE**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de este proyecto de titulación con componentes de investigación aplicada y/o de desarrollo con fines de difusión pública, además apruebo la reproducción de este proyecto de titulación con componentes de investigación aplicada y/o de desarrollo dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, 4 de febrero de 2026

EL AUTOR

Firmado electrónicamente por:
EDISON GEOVANNY NARANJO ANDRADE
Validar únicamente con FirmaEC

Edison Geovanny Naranjo Andrade

AGRADECIMIENTO

Agradezco profundamente a mi familia, pilar fundamental en cada etapa de este camino académico y personal.

A mi esposa, Cecilia, mi compañera de vida, cuya entrega, paciencia y confianza superan cualquier título y hacen posible que los objetivos trazados se conviertan en realidad.

A mi padre, Luis, eje central de nuestra familia, por su ejemplo de responsabilidad y constancia, valores que han guiado mi formación personal y profesional.

A la UPSE por brindarme la oportunidad de realizar una nueva titulación y todos quienes conforman la empresa INGEL-PRO por auspiciar el presente trabajo.

Edison Geovanny, Naranjo Andrade

DEDICATORIA

A mi madre, Tarcila Andrade, quien a lo largo de toda su vida se ha entregado incansablemente al trabajo, sacrificando su tiempo, su descanso y muchas veces sus propios sueños, con el firme propósito de brindarme una profesión y la estabilidad necesaria para salir adelante.

Desde hace 6 años, enfrenta una enfermedad incurable que ha marcado nuestras vidas dolorosamente.

Ver cada día como su vida se apaga poco a poco es un sufrimiento e impotencia indescriptible pero también verla luchar en ese día a día es una lección de valentía y amor que llevaré siempre conmigo.

Deseo desde lo más profundo de mi corazón, que Dios le conceda la oportunidad de verme culminar esta etapa como magíster. Quisiera que no me dejes, pero la voluntad de Dios está sobre nosotros.

Edison Geovanny, Naranjo Andrade

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente trabajo aborda el desarrollo de un prototipo de software HMI (Human Machine Interface) orientado al control y monitoreo de sistemas de automatización industrial basados en PLCs SIEMENS, con enfoque de aplicación a los proyectos que desarrolla la empresa INGEL-PRO.

Se diseña un prototipo de software HMI capaz de monitorear y controlar variables del PLC analógicas y digitales, tanto para lectura como para escritura; el mencionado HMI es una aplicación que funciona en el sistema operativo Windows (PC) con la capacidad de comunicación directa a los controladores sin requerir hardware o software adicional. El prototipo tiene la versatilidad de diseño en forma de plantilla, lo cual permite reutilizarlo para diferentes proyectos.

El desarrollo del HMI se realiza en Visual Studio con el lenguaje de programación C#, la versión utilizada de Visual Studio es gratuita y C# se utiliza como lenguaje por su disponibilidad de librerías existentes enfocadas a la comunicación industrial. La estructura del programa está realizada de forma tal que su código puede ser reutilizado para una aplicación diferente al estar debidamente organizado en clases y formularios.

El trabajo sigue una metodología que analiza los requerimientos de la empresa en base a sus clientes, con lo cual se define que el bus de campo industrial a usar es Ethernet, los controladores de la gama Simatic de Siemens S7-1200 y S7-1500 y el HMI en funcionamiento local en una PC, esto en base al tipo de cliente final usuario del sistema.

Como resultado se obtiene un prototipo de software que permite la comunicación con controladores para la visualización y control de variables de proceso, adaptable a cualquier proyecto que desarrolle la empresa, logrando la implementación de sistemas HMI sin costos de licencia o uso de software comercial al ser desarrollado con software de uso gratuito, siendo además una solución posible de implementar en pequeñas y medianas empresas.

Palabras claves: HMI, PLC, VISUAL STUDIO

ABSTRACT

This work is the development of a prototype HMI (Human Machine Interface) software oriented to the control and monitoring of industrial automation systems based on SIEMENS PLCs, with a focus on application to the projects developed by the company INGEL-PRO.

A prototype HMI software is designed to monitor and control analog and digital PLC variables, both for reading and writing. This HMI is an application that runs on the Windows operating system (PC) and can communicate directly with the controllers without requiring additional hardware or software. The prototype's versatile design, presented as a template, allows for its reuse in different projects.

The HMI was developed in Visual Studio using the C# programming language. The version of Visual Studio used is free, and C# was chosen due to the availability of existing libraries focused on industrial communication. The program's structure is designed so that its code can be reused for different applications, as it is properly organized into classes and forms.

The work follows a methodology that analyzes the company's requirements, which defines that the industrial fieldbus to be used is Ethernet, the controllers are from the Siemens Simatic S7-1200 and S7-1500 range, and the HMI operates locally on a PC, based on the type of end customer using the system.

The result is a software prototype that allows communication with controllers for the visualization and control of process variables, adaptable to any project developed by the company, achieving the implementation of HMI systems without license costs or use of commercial software as it is developed with free software, and is also a possible solution to implement in small and medium-sized enterprises.

Keywords: HMI, PLC, VISUAL STUDIO

INTRODUCCIÓN

En el contexto de la automatización industrial, los sistemas de control basados en Controladores Lógicos Programables (PLCs) y las Interfaces Hombre-Máquina (HMI) desempeñan un rol esencial, ya que permiten la supervisión, control y visualización de variables de proceso. Sin embargo, el uso de software HMI comercial asociado a estos controladores por lo general implica altos costos de licenciamiento y dependencia tecnológica de proveedores específicos.

Este escenario representa una limitación significativa para empresas de ingeniería y automatización, como INGEL-PRO, que se dedican al desarrollo e implementación de proyectos de automatización con requerimientos diversos y buscan soluciones flexibles, escalables y económicamente sostenibles.

Ante esta problemática, surge la necesidad de investigar y desarrollar alternativas de software HMI basadas en tecnologías de código abierto como Visual Studio y el lenguaje de programación C#. Estas tecnologías permiten la creación de interfaces gráficas personalizadas, con capacidades de comunicación a PLCs, reduciendo costos y facilitando la adaptación del software a distintos proyectos industriales.

El HMI desarrollado tiene como alcance la comunicación con un controlador lógico programable, para lectura y escritura de variables tanto digitales como analógicas, adicionalmente dispone de conexión de a base de datos local la cual gráficamente desde el HMI puede consultarse.

La comunicación, velocidad de transmisión y respuesta del sistema en general para leer o escribir las variables se prueban con la conexión en línea al PLC, comprobándose de esta manera el óptimo funcionamiento del HMI.

El presente trabajo se justifica por su aporte práctico, ya que implementa un prototipo de software HMI que pueda reemplazar, parcial o totalmente, el uso de software comercial de pago, ofreciendo una solución confiable y reutilizable para los proyectos de la empresa INGEL-PRO y del sector industrial en general.

Planteamiento de la investigación

El presente trabajo describe el desarrollo de software HMI con interfaz gráfico, capaz de comunicarse con Controladores Lógicos Programables, como opción de sustitución Software de pago que actualmente usa la empresa beneficiaria, con lo cual se puede reducir costos de implementación de nuevos proyectos de automatización industrial que la empresa ejecuta con regularidad.

Objetivo General

Desarrollar soft HMI con Visual Studio y el lenguaje C# para crear interfaces de control/monitoreo en tiempo real de sistemas industriales comandados por Controladores Lógicos Programables de la marca Siemens para el reemplazo de Software de pago.

Objetivos Específicos

- Desarrollar una plantilla con código de programación lista para comunicarse con Controladores Lógicos Programables, y que puedan ser usadas para diferentes aplicaciones.
- Crear interfaces gráficos capaces de mostrar y controlar las principales variables de un PLC.
- Comunicar variables tipo digital y analógicas del PLC hacia el HMI.
- Obtener variables de más PLCs vía Red Industrial.

Planteamiento hipotético

Un Soft HMI de código abierto con disponibilidad de plantillas de programa predefinidas representa una alternativa viable y económica frente a los softwares comerciales tradicionales, permitiendo un enlace confiable con PLCs Siemens en entornos industriales.

CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL

1.1. Estado del arte

En lo referente al desarrollo de sistemas HMI, en los últimos 5 años se tiene un auge debido a la nueva tendencia de la Industria 4.0; estos desarrollos se basan en la utilización de tecnologías Web o sistemas realizados con software de programación como Python o Visual Studio. Como un objetivo común de todos estos trabajos es disponer un sistema propietario sin dependencia de una licencia o soporte de una marca comercial.

Corral González (2021) argumenta que la motivación para realizar el Trabajo enfocado al "Diseño y Desarrollo de un sistema HMI para una Aplicación de Industria 4.0" radica en la necesidad de desarrollar un sistema autónomo, independiente de software específico, que resulte innovador, sostenible y de fácil ejecución. En este sentido, los sistemas HMI (Human Machine Interface) representan una solución ventajosa para distintos sectores industriales, ya que permiten un acceso global a la información sin requerir configuraciones particulares de hardware o software.

Asimismo, Corral González (2021) señala que el uso de aplicaciones web contribuye a la reducción de costos económicos y operativos, al facilitar la toma de decisiones en momentos clave y disminuir la necesidad de realizar tareas administrativas o desplazamientos físicos, centralizando la información en un único entorno.

El uso específico de Visual Studio con el lenguaje C# para la creación de interfaces gráficos de control y monitoreo por lo general se orienta a proyectos basados en microcontroladores.

En el trabajo realizado por Udari Bhagya Liyanage (2021), el desarrolló un sistema de HMI basado en aplicaciones de Microsoft Windows con el propósito de controlar un microvehículo el uso de Microsoft Visual Studio empleando el lenguaje C#, optimiza el interfaz gráfico mediante el uso de Windows Forms.

Visual Studio posee una gama amplia para desarrollo de sistemas HMI, por ejemplo, puede ser empleado para tecnologías Web. Cabe indicar que este tipo de soluciones además personalizan o se enfocan en un control en específico del proyecto.

Según Shyr-Long Jeng (2021), los avances en los controladores industriales convencionales han impulsado la aparición de nuevas tecnologías, como el uso de

múltiples lenguajes, las aplicaciones multiplataforma y las capacidades de monitoreo y control remoto. No obstante, las interfaces hombre-máquina (HMI) de estos controladores tradicionales y las de los dispositivos móviles presentan limitaciones, ya que no permiten la transmisión directa de información en tiempo real entre sí. Por tanto, plantea un método sencillo para evolucionar las HMI de los controladores industriales convencionales hacia sistemas capaces de ofrecer acceso remoto a través de la web. Para ello usa una arquitectura basada en el modelo vista-controlador, empleando tecnologías web.

El requerimiento de la industria de obtener datos de proceso es común, tradicionalmente se lo hace con soluciones de pago, en este contexto con C# es posible enlazar la aplicación HMI con una base de datos, desde el punto de vista de una solución SCADA las posibilidades se amplían.

Según Tükez y Kaya (2022), la producción industrial actual se caracteriza por estar distribuida en áreas extensas y por presentar una elevada complejidad operativa, lo que exige sistemas de supervisión y control más eficientes para optimizar el rendimiento de los procesos. Es por eso que los sistemas SCADA desempeñan un papel fundamental, ya que permiten la supervisión, el control y la adquisición de datos en tiempo real, así como el registro histórico de la información del proceso. No obstante, el alto costo de las soluciones SCADA comerciales limita su adopción en pequeñas y medianas industrias. Ante esta problemática, el autor propone el desarrollo de un sistema SCADA implementado mediante el lenguaje de programación C#, el cual ofrece una alternativa de menor costo capaz de proporcionar funcionalidades equivalentes. En el estudio se diseñó un sistema SCADA orientado a la adquisición, visualización y almacenamiento de datos del proceso, aplicado al monitoreo y control de un modelo de fábrica inteligente.

La popularidad de Python crece exponencialmente por su versatilidad de aplicación en varias áreas, en el desarrollo HMI también ha sido usado como lenguaje de programación.

Dikme (2021) sostiene que una aplicación de uso industrial debe caracterizarse por su estabilidad y eficiencia en el consumo de memoria, garantizando al usuario una interacción fluida, sin interrupciones ni ralentizaciones perceptibles. En el desarrollo de interfaces hombre–máquina (HMI) para entornos industriales, estas exigencias resultan críticas, ya que la confiabilidad y el rendimiento del sistema impactan directamente en la operación. En este contexto, el empleo de lenguajes cercanos al hardware suele ofrecer

mejores resultados en términos de estabilidad y eficiencia que los lenguajes altamente declarativos.

1.2.Desarrollo teórico y conceptual

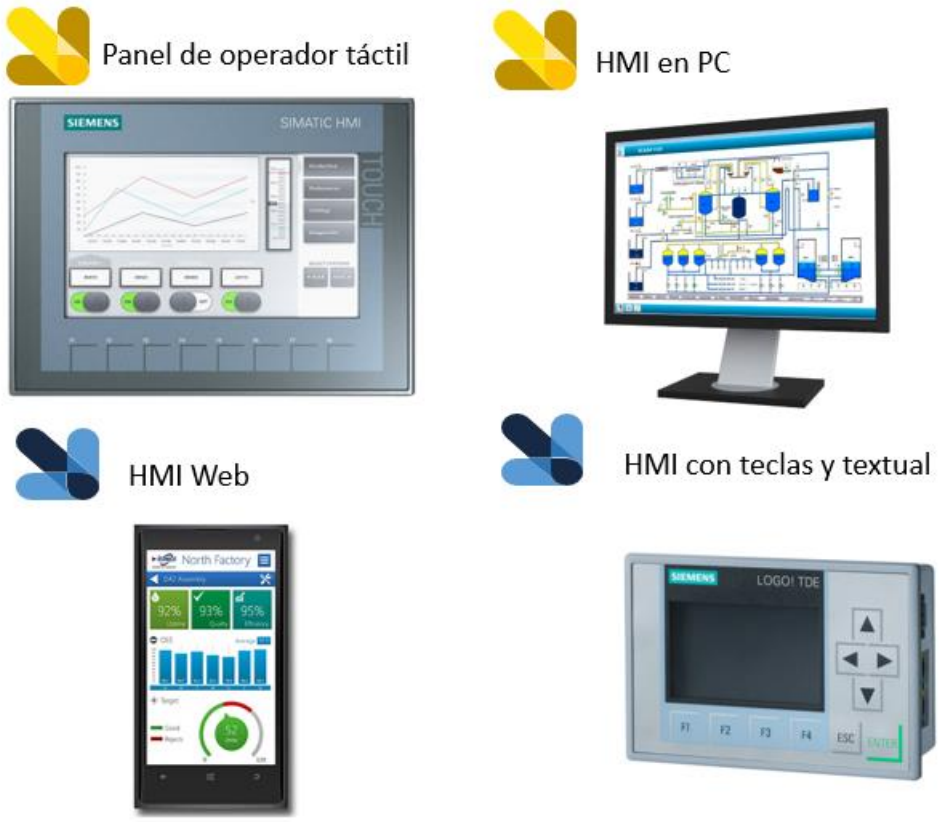
1.2.1. Tipos de HMI

Una interfaz hombre-máquina (HMI, por sus siglas en inglés) es una plataforma que permite la interacción entre un usuario y una máquina.

En la figura 1 se ilustra los HMI de nivel industrial de uso común en el medio.

Figura 1

Tipos de HMI



Euroinnova (2024) indica que en lo referente a tipos de HMI entre las más destacadas se encuentran las HMI táctiles, las cuales han revolucionado la forma en que los usuarios interactúan con los sistemas al reemplazar los controles físicos por pantallas sensibles al

tacto, las basadas en texto continúan siendo relevantes en determinados entornos técnicos ya que requieren la introducción de comandos a través del teclado, ofrecen un alto nivel de precisión y control, siendo especialmente valoradas en sistemas donde la simplicidad, la rapidez y la eficiencia operativa son prioritarias.

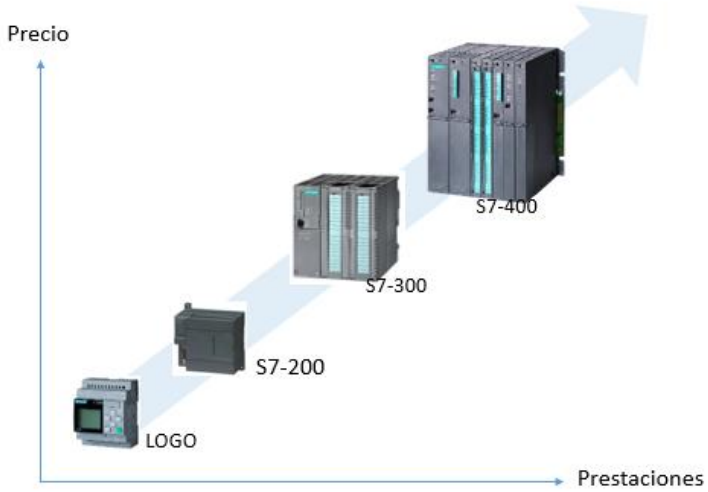
En cuanto a la accesibilidad remota, las HMI web han adquirido un papel fundamental al permitir la supervisión y el control de sistemas a través de navegadores web. Gracias al uso de internet, estas interfaces eliminan las limitaciones geográficas, facilitando la interacción con los sistemas desde cualquier ubicación con conexión disponible.

1.2.2. Controladores lógicos programables de la gama Simatic

Los PLCs S7-300, S7-1200 y S7-1500 son una marca registrada de SIEMENS. La evolución de los diferentes autómatas de esta marca ha ido a la par con el incremento del uso de las tecnologías de la información y de las redes de comunicaciones. Hoy en día se requieren características diferentes a las exigidas hace 20 años. A nivel industrial se dispone de las series 300/1200/1500. Cada serie se utiliza para aplicaciones y necesidades concretas. En la figura 2 se puede apreciar el posicionamiento de los diferentes PLC de Siemens antes de la aparición del S7-1200. (Peciña, 2018, p. 20).

Figura 2

Evolución de los autómatas Siemens



Nota. Tomada de Programación de controladores avanzados SIMATIC S7 1500 con TIA Portal AWL, KOP y SLC por . L Peciña, 2019, p. 19. Marcombo.

El micro PLC LOGO es un programador utilizado para aplicaciones de tipo domótico. El S7-200 es un PLC que se dejó de fabricar hace algunos años. Dicho autómata fue sustituido por el S7-1200, destinado a aplicaciones industriales de control de pequeño tamaño o automatización de máquinas.

Las series S7-300 y S7-400 son autómatas utilizados para grandes aplicaciones industriales de control. Hoy día conviven junto al nuevo PLC S7-1500. Es bien seguro que el S7-1500 será el que, en un tiempo no muy lejano, sustituirá al S7-300/400.

En la nueva concepción de Siemens, cabe destacar que todos los PLC nuevos pueden ser programados en el mismo lenguaje que el S7-300.

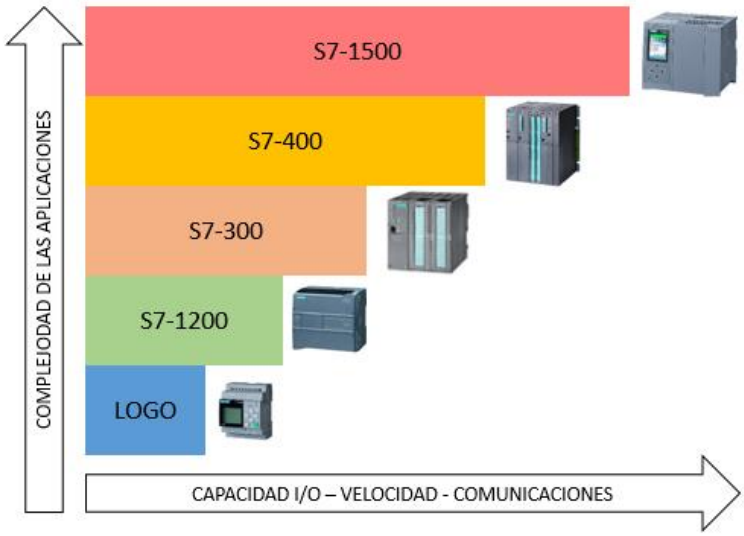
Con la aparición de S7-1200, Siemens lanzó una plataforma de programación nueva denominada TIA Portal.

El PLC S7-1500 ha llegado para sustituir al S7-300. El S7-1500 dispone de un mejor tratamiento de las instrucciones, sobre todo en la norma de estandarización IEC 61131. (Peciña, 2018, p. 21).

La figura 3 muestra de forma gráfica la capacidad de cada PLC según la gama.

Figura 3

Clasificación de los autómatas Siemens



Nota. Tomada de Programación de controladores avanzados SIMATIC S7 1500 con TIA Portal AWL, KOP y SLC por . L Peciña, 2019, p. 21. Marcombo.

Existen muchos tipos diferentes de autómatas dentro de cada una de las series. Las diferencias son de memoria, número de temporizadores y contadores, de comunicaciones, etc. (Peciña, 2018, p. 21).

1.2.3. Fundamentos de redes industriales

Los sistemas de comunicación industrial hacen posible el intercambio de información entre distintos equipos en fábricas, plantas productivas y otros entornos industriales. Existen buses de comunicación basados en serial y ethernet que permiten la integración eficiente y confiable de sensores, actuadores y controladores, asegurando la transmisión de datos en tiempo real (García Jiménez, 2025).

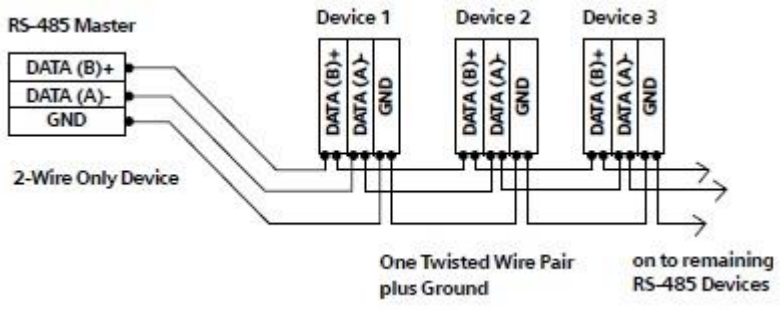
1.2.3.1. Buses de campo basados en serial RS-485

La comunicación serial RS-485 es uno de los estándares más utilizados en entornos industriales debido a su robustez, simplicidad y bajo costo de implementación. Este estándar permite la transmisión de datos en largas distancias y en ambientes con altos niveles de ruido electromagnético, características comunes en aplicaciones industriales. RS-485 utiliza una transmisión diferencial balanceada, lo que mejora la inmunidad al ruido y garantiza una comunicación más confiable en comparación con otros estándares seriales (Come-Star, 2025).

La figura 4 muestra el cableado típico del bus RS-485.

Figura 4

Comunicación serial RS-485



Nota. Tomada de InGel-Pro (2023). https://ingel-pro.com/blog1.html

Los buses de campo basados en RS-485 emplean comunicación serial diferencial, lo que les permite operar en entornos con elevados niveles de ruido electromagnético y alcanzar

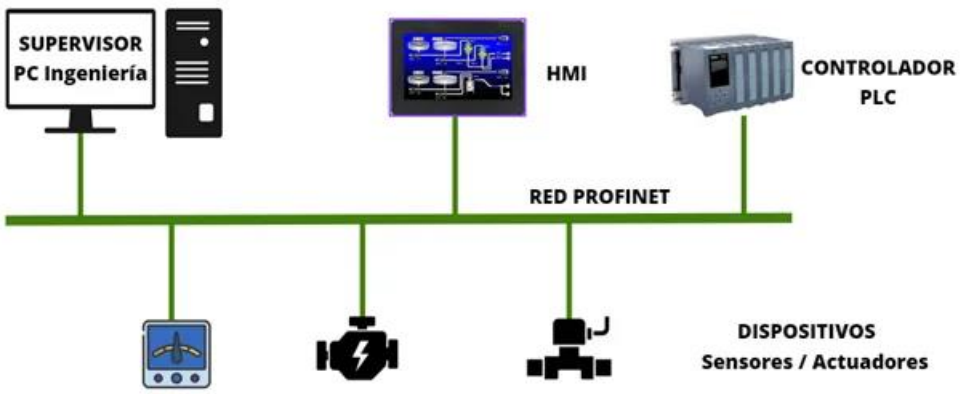
largas distancias de transmisión. Protocolos ampliamente utilizados como Modbus RTU y Profibus-DP usan en RS-485 como capa física (Come-Star, 2025).

1.2.3.2.Buses de campo basados en ethernet

Linares Gutiérrez (2025) señala que el nivel de control está conformado por equipos como PLC y controladores industriales, encargados de procesar los datos obtenidos desde el nivel de campo. En esta capa, la comunicación se lleva a cabo principalmente a través de buses de campo de alta velocidad y redes industriales basadas en Ethernet, tales como Profinet, Modbus TCP y EtherNet/IP, lo que contribuye a una mayor eficiencia en la automatización de los procesos. La figura 5 muestra la arquitectura de conexión basada en ethernet.

Figura 5

Bus de campo ethernet industrial



Nota. Tomada de Wit Automatización (2023). https://witautomatizacion.es/que-es-profinet-y-para-que-sirve/

A menudo se confunde el bus de campo con el protocolo, por ejemplo Profinet (Process Field Network), no es un bus, es una red Industrial Ethernet, por lo tanto, la comunicación está basada en Ethernet estándar (TCP/IP y UDP). Este sistema dispone de dos variantes de comunicación: PROFINET RT, orientado a aplicaciones de tiempo real, y PROFINET IRT, diseñado para comunicaciones isócronas de alta precisión. Su fuerte presencia en el ámbito industrial ha consolidado su reconocimiento como una tecnología clave para la

evolución hacia fábricas inteligentes y entornos de automatización avanzada. Además, permite tasas de transmisión que alcanzan o superan los 100 Mbps (Linares Gutiérrez, 2025).

1.2.4. Microsoft Visual Studio

Visual Studio es una herramienta eficaz para desarrolladores que puede usar para completar todo el ciclo de desarrollo en un solo lugar. Es un entorno de desarrollo integrado (IDE) completo que puede usar para escribir, editar, depurar y compilar código. A continuación, implemente la aplicación. Visual Studio incluye compiladores, herramientas de finalización de código, control de código fuente, extensiones y muchas otras características para mejorar todas las fases del proceso de desarrollo de software. (Microsoft, 2020).

Está destinado a ser utilizado, entre otras cosas, como base para el desarrollo de sitios web, programas informáticos y aplicaciones. Dado que este no es un IDE vinculado al lenguaje, Visual Studio admite más de 30 lenguajes de programación. Puede escribir código en Python, HTML, JavaScript y C++, entre otros. Visual Studio viene en tres ediciones: Community, Professional y Enterprise, disponibles para Windows y macOS. (Digi License, 2022).

Visual Studio está disponible en varias versiones, el presente proyecto será desarrollado en su versión gratuita, la cual permite todo lo necesario para este fin. Esta versión es Visual Studio Community.

A diferencia de todas las demás versiones, Visual Studio Community es una licencia gratuita que se puede descargar desde el sitio web de Microsoft. Esta edición es utilizada principalmente por estudiantes y desarrolladores independientes para crear aplicaciones. Hasta 5 usuarios en una organización no corporativa pueden usar Visual Studio Community. Para las empresas clasificadas como organizaciones empresariales, existen restricciones en la cantidad de PC y los ingresos anuales. Según el fabricante, las grandes empresas con menos de 250 PC o un ingreso anual de menos de 1 millón de dólares estadounidenses aún pueden usar Community Edition, pero solo si la naturaleza del uso es investigación académica, enseñanza o contribución a proyectos de código abierto. (Digi License, 2022).

1.2.5. El lenguaje C#

Se trata de un lenguaje de programación que está orientado a objetos y ha sido diseñado con el fin de compilar varias aplicaciones que se ejecutan en .NET framework. Por otra parte, hay que saber que este es un lenguaje de propósito general, lo cual permite crear sitios y aplicaciones para Windows, Linux, Mac, Android, iOS, etc.

En muchos sentidos, se puede decir que se trata del lenguaje C de Microsoft. De hecho, es inevitable notar que C# se deriva de C y C++. Su sintaxis es muy similar a la de Java, por lo que simplifica la tarea del desarrollador.

Además, hace uso de objetos de la plataforma .NET, pero a su vez incluye mejoras de otros lenguajes. Antes de comenzar con C#, es recomendable estar familiarizado con Javascript, Python y, obviamente, C. (Certus, 2020).

Existen múltiples lenguajes de programación entre los que elegir en lugar de C#. Todos ellos cuentan con diferentes características y son idóneos para distintos productos. En la siguiente tabla puedes encontrar una comparación entro los más importantes. (Tokio School, 2024).

Tabla 1

Comparación entre lenguajes de programación

Características	C#	Java	Python	JavaScript	Swift
Paradigma	Orientada a objetos	Orientada a objetos	Multiparadigma	Multiparadigma	Orientada a objetos
Tipado	Estático	Estático	Dinámico	Dinámico	Estático
Plataforma	.NET	JVM	Multiplataforma	Web, acepta multiplataforma	Ecosistema OS
Sintaxis	Concisa, similar a C++	Verborrágica	Muy legible	Flexible	Concisa y moderna
Usos principales	Desarrollo web y aplicaciones de escritorio	Desarrollo empresarial y Android	Ciencias de datos	Desarrollo web	Aplicaciones para ecosistema OS
Rendimiento	Alto	Alto	Moderado	Varía	Alto

Nota. Tomada de Tokio School. https://www.tokioschool.com/noticias/c-que-es/

De la información de la tabal 1, en vista de que el lenguaje C# permite desarrollar aplicaciones de escritorio, se elige para el presente proyecto.

1.2.6. Descripción de la empresa

INGEL-PRO es una empresa privada categorizada en el conjunto de pequeñas y medianas empresas (PYMES) la cual tiene como giro comercial principal las actividades relacionadas a los servicios de Ingeniería Electrónica Industrial, de esta actividad principal se especializa en 3 ejes:

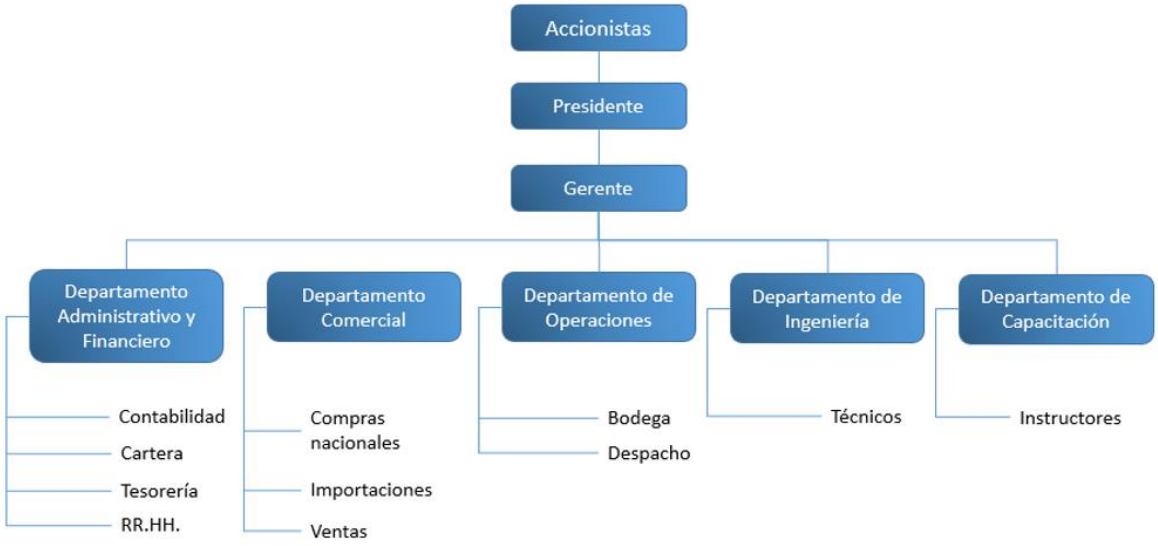
- Comercialización de material eléctrico.
- Desarrollo e implementación de proyectos de automatización industrial.
- Servicios de capacitación técnica y consultoría.

La sede se encuentra ubicada en la ciudad de Quito, Ecuador, iniciando sus actividades comerciales en julio de 2011.

La organización de los diferentes departamentos de la empresa se muestra en la figura 6.

Figura 6

Estructura organizacional de la empresa INGEL-PRO



CAPÍTULO 2. METODOLOGÍA

2.1. Contexto de la investigación

Se plantea el desarrollo de un software HMI (Interfaz Hombre–Máquina) con interfaz gráfica, orientado a la supervisión y control de procesos industriales mediante su comunicación directa con Controladores Lógicos Programables (PLC) como una alternativa tecnológica frente a los sistemas HMI comerciales de pago que actualmente utiliza la empresa InGel-Pro para la ejecución de sus proyectos de automatización industrial, los cuales implican un costo significativo asociado a la adquisición de licencias de software propietario. En este contexto, la propuesta busca reducir los costos de implementación de nuevos proyectos, sin comprometer la funcionalidad y confiabilidad de uso requeridos en los sistemas HMI.

La empresa en mención se dedica a la implementación de proyectos de ingeniería en las áreas de electrónica y automatización industrial, ofreciendo soluciones de control y supervisión a diversos sectores productivos.

No obstante, el uso de plataformas HMI comerciales limita la competitividad de la empresa, especialmente en proyectos que son para pequeñas y medianas empresas, donde los costos asociados a licencias de software representan un limitante para decidirse por la implementación.

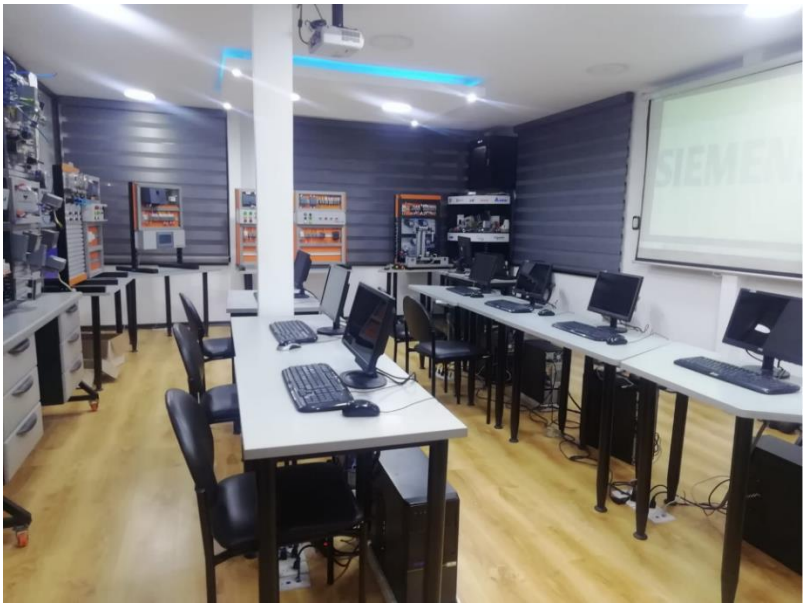
2.1.1. Laboratorio de pruebas

Con la finalidad de una posterior implementación del sistema HMI en un ambiente industrial, previamente las pruebas se realizan en el laboratorio de la empresa en donde se cuenta con los mismos controladores lógicos programables y una red ethernet industrial lo cual es propicio para las pruebas de funcionamiento.

En la figura 7 se muestra una imagen del laboratorio.

Figura 7

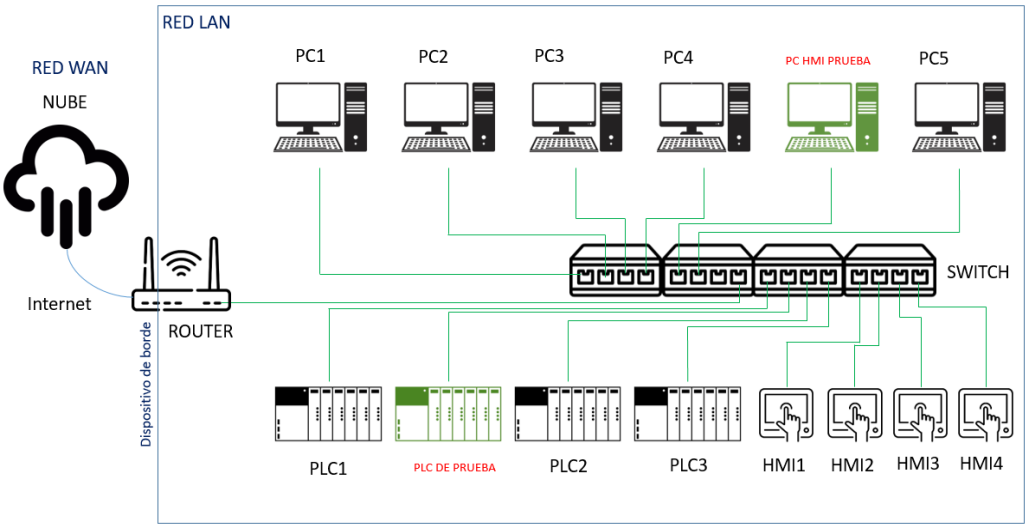
Laboratorio de pruebas



En la figura 8 se esquematiza la arquitectura de red del laboratorio en donde se realizan las pruebas, se dispone de varias PCs, varios PLCs y HMIs, todos conectados entre sí con un bus de campo ethernet industrial mediante un switch.

Figura 8

Arquitectura de red del laboratorio de pruebas



En la figura 8 se aprecia que la topología de red es del tipo estrella en donde todos los dispositivos se concentran en el switch. Esta topología de red es la más común a nivel de pequeña y mediana empresa.

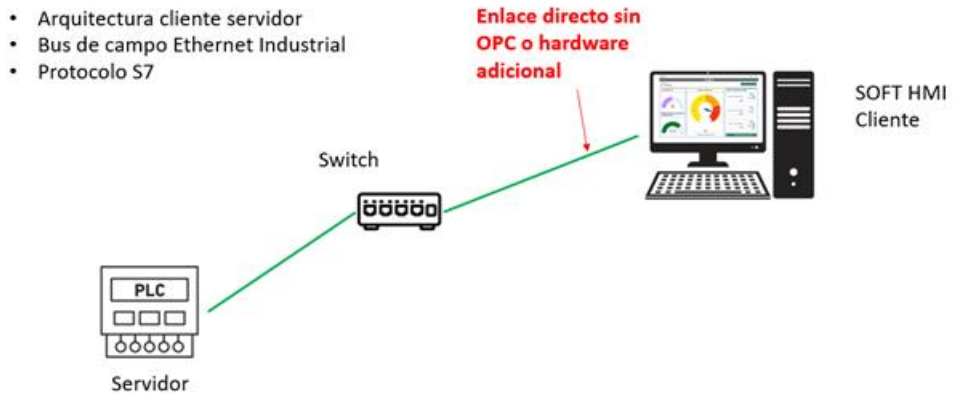
2.2. Diseño y alcance de la investigación

2.2.1. Diseño experimental

Se adopta un diseño experimental para validar la capacidad del Soft HMI para control y monitoreo en tiempo real de sistemas de automatización industrial con PLCs Siemens, la implementación en términos generales permite la comunicación directa del HMI con el PLC sin la intervención de hardware o software adicional, como por ejemplo Gateways o servicios OPC respectivamente. La figura 9 grafica una arquitectura básica a realizar.

Figura 9

Arquitectura de comunicación PLC y HMI



Se utiliza PLCs simulados y reales para comprobar la comunicación con el HMI, así mismo redes Ethernet simuladas y físicas. Gracias al interfaz desarrollado de forma gráfica y en tiempo real se comprueba si las diferentes variables comunicadas al PLC están operando adecuadamente, sea para lectura o escritura y con un tiempo de respuesta acorde a los HMIs comerciales

En lo referente al alcance de la investigación, la misma es del tipo explicativo para determinar los factores técnicos de funcionamiento del HMI y de esta manera obtener las ventajas y limitaciones frente a productos comerciales.

Mhetraskar et al. (2020) analizan la automatización industrial basada en PLC y HMI mediante el uso de protocolos de comunicación en tiempo real, estableciendo la comunicación mediante Ethernet y enlaces seriales, lo que permite la lectura y escritura de registros del PLC desde el HMI. A través de esta comunicación se verifica el monitoreo en tiempo real de variables digitales y analógicas, así como la ejecución de órdenes de control enviadas desde la interfaz. La prueba confirma la correcta interacción entre ambos dispositivos, tanto en operación local como remota, garantizando la supervisión y el control confiable del sistema.

2.3. Tipo y métodos de investigación

En lo referente al enfoque se adopta el tipo mixto en razón de que se requiere técnicas tanto cuantitativas como cualitativas.

Desde el enfoque cuantitativo lo que se busca obtener es la cantidad y tipo de variables que pueden leer/escribirse simultáneamente con un tiempo de respuesta máximo de 500ms. Para este fin se necesita la recopilación de datos mediante pruebas de funcionamiento con PLCs simulados y PLCs físicos, con esto se podrá obtener los datos necesarios para evaluar la efectividad del Soft HMI.

En lo referente al enfoque cualitativo se recoge experiencias de usuario, como por ejemplo la percepción del operador durante la interacción con el Soft HMI, si tuvo problemas de “congelamiento” de las variables o fluyó sin problemas. Esto se realiza con el cliente interno de la empresa antes de su uso para el cliente final mediante entrevistas u observación directa.

Descritos los dos enfoques anteriores se justifica el uso del enfoque tipo mixto lo cual permite evaluar el Soft HMI tanto en funcionamiento técnico como en la percepción que tiene el usuario durante su uso.

En lo referente al método de investigación se opta por el método Hipotético-Deductivo, esto en razón de que se parte de una hipótesis: “Un Soft HMI de código abierto con disponibilidad de plantillas de programa predefinidas representa una alternativa viable y económica frente a los softwares comerciales tradicionales, permitiendo un enlace confiable con PLCs Siemens en entornos industriales”. Luego mediante la experimentación y pruebas se validará esta hipótesis.

Adicionalmente, al trabajar con tema de índole tecnológico es posible realizar este esquema: pruebas, mediciones y observaciones.

2.4. Población y muestra

La población objetiva del presente trabajo lo constituyen las empresas integradoras de ingeniería en Automatización y el cliente final de las industrias por medio de los técnicos de mantenimiento y operadores.

Por tanto, el tipo de población es finita considerando que se iniciará con un grupo determinado, y en cuanto a la muestra se realiza un muestreo no probabilístico Por Conveniencia tomando como nodo central a una empresa integradora de ingeniería.

Para asegurar que el personal escogido son sujetos de prueba, se aplica el siguiente criterio de prueba:

- Las empresas integradoras de ingeniería disponen de las condiciones de prueba para el Soft HMI así como las plantas industriales en donde finalmente funcionará el software.

La primera muestra consiste en una comunicación plc-hmi de 4 variables digitales y 4 variables analógicas tanto para lectura y escritura.

La segunda muestra consististe en evaluar la comunicación punto a punto PLC-HMI, conectando y desconectando súbitamente el PLC para observar el comportamiento del HMI.

Se incluye finalmente una muestra de 2 operadores de procesos industriales, quienes participarán en pruebas de usabilidad y evaluación cualitativa del HMI Soft; en la fase de pruebas los “operadores” son los clientes internos de la empresa.

2.5. Técnicas e instrumentos de recolección de datos

Partiendo de que el proyecto tiene un enfoque de investigación mixto, se utilizarán los siguientes instrumentos de medición:

- Pruebas de rendimiento del software.
- Registro de errores.
- Observación directa en pruebas de usuario.
- Software de monitoreo de red.

- Simuladores de PLC
- Prueba de latencia y tiempo de respuesta.

Los ítems descritos anteriormente se enmarcan dentro de la observación directa.

Sin embargo, también es posible aplicar entrevistas a los usuarios del HMI Soft:

- Encuesta de satisfacción del usuario.

Este instrumento permite cuantificar la satisfacción del usuario.

2.6. Procesamiento de la evaluación: Validez y confiabilidad de los instrumentos aplicados para el levantamiento de información.

Al tratarse de un proyecto tecnológico que necesita de pruebas para evaluar se confiabilidad se recurre a una opinión experta.

Antes del funcionamiento en campo del Soft HMI se realizarán validaciones con personal técnico de empresas integradoras de ingeniería en automatización, quienes revisarán la velocidad de respuesta en lectura /escritura de las variables y reacción ante una desconexión súbita de la red.

CAPÍTULO 3. RESULTADOS Y DISCUSIÓN

3.1. Metodología de desarrollo

Considerando que el producto final a desarrollar es un software, para conseguir los objetivos se realizan fases de trabajo que se indican en la figura 10.

Figura 10

Fases del desarrollo

En la planificación se recoge información interna y externa, en la fase de diseño se parte de una idea inicial, en el desarrollo se emplean las herramientas de software y hardware para la creación práctica, y finalmente en la fase de pruebas se evalúan los resultados obtenidos.

3.2. Planificación

La necesidad del desarrollo del producto en cuestión comienza por recolectar las opiniones de los clientes de la empresa, en las tablas 2, 3 y 4 se muestran lo que la empresa denomina “historia de usuario”, seleccionadas de un conjunto de historias.

Tabla 2

Historia de usuario 1

Historia de usuario - 01	
Cliente:	Ing. Patricio Cunalata
Empresa:	Tesalia CBC Planta Machachi
Requerimiento:	El sistema HMI debe permitir obtener datos de producción de diferentes variables. Los reportes de preferencia en formato Excel.

Tabla 3

Historia de usuario 2

Historia de Usuario - 02	
Cliente:	Ing. Daniel Guerrón
Empresa:	Pronavit
Requerimiento:	Monitoreo de variables tipo analógicas como presión, nivel y caudal. Visualización de las mismas en forma gráfica amigable para el operador.

Tabla 4

Historia de usuario 3

Historia de Usuario - 03	
Cliente:	Ing. Miguel Yanez
Empresa:	Solemeec
Requerimiento:	HMI centralizado en una PC capaz de monitorear al menos un área entera de una planta, pero sin costo de licenciamiento

Al revisar estas y otras "historias de usuario" se tiene que los principales requerimientos que buscan los clientes del sistema HMI son:

- Entorno gráfico amigable.
- Posibilidad de monitoreo/control de diferentes tipos de variables PLC.
- Registradores de datos.

Por tanto, se determina que una primera versión del HMI a desarrollar es de tipo Escritorio ya que se usará localmente en las instalaciones de la planta industrial, por tanto, al ser Windows el sistema operativo dominante se elige a Visual Studio como la plataforma de programación por ser nativa para el desarrollo de aplicaciones de escritorio para Microsoft.

El lenguaje de programación a usar es C# por su variedad de Frameworks disponibles para conexión a PLCs.

La Base de Datos a utilizar es Microsoft SQL Server debido a que usa .NET como "motor" de funcionamiento y las aplicaciones de escritorio se basan en este "motor". La versión de SQL Server a utilizar es la Express ya que es gratuita y proporciona un espacio de 10GB, lo cual es suficiente para aplicaciones de PLCs.

A nivel interno de la empresa se requiere el HMI pueda ser adaptable a cualquier otro proyecto y fundamentalmente pueda ser desarrollado fácilmente por otro ingeniero quien no es el creador del proyecto original, la tabla 5 resume este requerimiento.

Tabla 5

Requerimiento del cliente interno

Requerimiento interno	
Cliente:	Ingeniero de automatización de la empresa.
Requerimiento:	El software HMI debe ser de fácil modificación, para que pueda ser implementado en otros proyectos y por cualquier compañero del área de ingeniería de la empresa.

Se resume en una palabra lo mostrado en la tabla 5, el objetivo es lograr una **plantilla,** por tanto, el desarrollo hace énfasis en una estructura ordenada y con código comentado, que, con una inducción sencilla pueda ser utilizada por cualquier ingeniero de la empresa.

3.3. Diseño

3.3.1. Componentes de hardware y software para el desarrollo HMI

El desarrollo del sistema HMI industrial propuesto se fundamenta en la adecuada selección e integración de componentes de hardware y software que permitan garantizar un correcto funcionamiento acorde a los requerimientos del cliente de la empresa.

3.3.1.1. Controlador lógico programable para pruebas

Específicamente para las pruebas de funcionamiento se usa un PLC físico Simatic S7-1500 modelo 1512C el cual está conectado y en funcionamiento en el laboratorio de la empresa, la figura 11 hace referencia a este módulo.

Figura 11

Módulo de pruebas

En la figura 11 se aprecia el PLC S7-1500 conectado a la PC, no se encuentra en una conexión punto a punto sino a traves de la arquitectura de red del laboratorio (figura 8), justamente para emular también las condiciones de una red física industrial.

Las características técnicas del PLC para la realización de pruebas de detalla en la tabla 6.

Tabla 6

Características del PLC utilizado para pruebas

Característica	Descripción
Modelo	SIMATIC S7-1500 CPU 1512C-1 PN (6ES7512-1CK01-0BA0)
Firmware (FW)	2.1
Memoria de programa	250 KB integrada versión compatible 2.1 (soporta características con versiones TIA Portal apropiadas)
Memoria de datos	1 MB integrada
Velocidad de procesamiento	48 ns por operación de bit
Dirección IP configurable desde la CPU	Si, mediante display integrado
Interfaz de red	PROFINET IRT con switch de 2 puertos integrado
Fuente de alimentación	24VDC (19.2 – 28.8 VDC)
Entradas digitales integradas	32 canales
Salidas digitales integradas	32 canales
Salidas analógicas integradas	2 canales
Contadores de alta velocidad	6 contadores rápidos
Salidas rápidas (PTO/PWM/frecuencia)	4 salidas para señal de frecuencia/PWM/PTO

La CPU 1512C-1 PN forma parte de las CPUs compactas de la familia SIMATIC S7-1500, las cuales están diseñadas para aplicaciones de automatización de complejidad media a alta, integrando entradas y salidas digitales y analógicas en un solo equipo. Este controlador incorpora comunicación PROFINET IRT, lo que permite una transmisión de datos mediante Ethernet industrial con alto rendimiento y comportamiento determinista, adecuada para sistemas de supervisión, interfaces HMI y redes de campo. Además, dispone de un display frontal integrado que facilita el diagnóstico local del estado de la CPU y la identificación de fallas sin necesidad de herramientas externas. (Siemens AG, 2016).

3.3.1.2. Software TIA Portal

A través de su capacidad para abordar una amplia gama de aspectos en la ingeniería, el TIA Portal de SIEMENS se destaca como un actor clave en el panorama de la automatización industrial. En el ámbito de la programación, el TIA Portal ofrece un entorno unificado que abarca desde la configuración de controladores periféricos hasta la gestión de la energía, lo que facilita enormemente la labor de los ingenieros

El Totally Integrated Automation Portal (TIA Portal) integra diferentes productos SIMATIC, por ejemplo: STEP 7, software de programación de PLC; WinCC: software de visualización. Dentro del TIA Portal, los productos TIA interactúan entre sí, ofreciendo soporte en todas las áreas implicadas en la creación de una solución de automatización (Siemens AG, 2023).

3.3.1.3. Simulador avanzado de PLCs Simatic

Para la demostración de funcionamiento tipo exposición de venta al cliente final se usa un simulador de PLC debido a la facilidad que este representa y sobretodo que cumple con los requerimientos técnicos para el enlace al HMI, destacando su característica de emulación de una red ethernet física.

PLCSim Advanced es una herramienta de software que permite la simulación y verificación del funcionamiento de controladores lógicos programables de Siemens, así como de los programas de automatización asociados a estos dispositivos. Esta aplicación proporciona un entorno virtual en el que es posible emular el comportamiento de PLC de las series SIMATIC S7-1200 y S7-1500, lo que facilita la validación de los programas de control sin la necesidad de disponer de hardware físico. La herramienta se integra de forma directa con el software de ingeniería de Siemens, como TIA Portal, permitiendo cargar y ejecutar los programas desarrollados dentro del entorno de simulación. Asimismo, ofrece capacidades de simulación en tiempo real, lo que posibilita observar el comportamiento dinámico del sistema de control de manera similar a un escenario industrial real. PLCSim Advanced incorpora funciones avanzadas de depuración y monitoreo, tales como la ejecución paso a paso, el uso de puntos de interrupción y la supervisión de variables, lo que contribuye a la detección y corrección eficiente de errores en los programas (Bacis, 2024).

Adicionalmente, la herramienta admite simulaciones de tipo Hardware-in-the-Loop, permitiendo la integración de dispositivos físicos, como módulos de entrada y salida o interfaces HMI, con el PLC virtual para evaluar el funcionamiento completo del sistema de automatización. También soporta diversos protocolos de comunicación industrial, entre ellos PROFINET, PROFIBUS y OPC UA, facilitando la simulación de la interacción del PLC con otros componentes y sistemas. Gracias a su escalabilidad y flexibilidad, PLCSim Advanced permite simular múltiples PLC de forma simultánea y

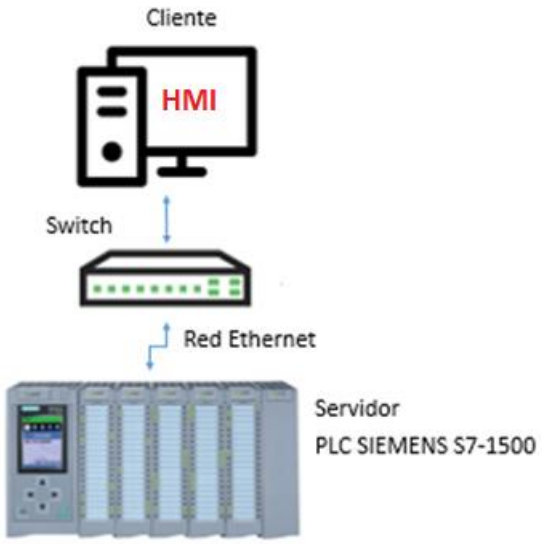
adaptar el entorno de simulación a los requerimientos específicos de proyectos de automatización distribuidos y de mayor complejidad (Bacis, 2024).

3.3.2. Conexión de red

Se define la arquitectura de red, la cual está basada en el bus de campo Ethernet, protocolo S7. El protocolo S7 es un protocolo propietario de Siemens utilizado para la comunicación directa con los PLC SIMATIC, especialmente para acceso a datos internos del controlador, permite leer y escribir variables, marcas, entradas, salidas y bloques de datos, es muy usado en software HMI desarrollado a medida (como el presente proyecto en C#). A diferencia de Profinet, S7 no está orientado a control determinista de E/S, sino al acceso a datos del PLC. La figura 12 ilustra una primera conexión de red entre el PC (HMI) y el PLC para posterior iniciar el desarrollo.

Figura 12

Conexión inicial de red



La arquitectura de la figura 12 se define como cliente–servidor en un bus de campo Ethernet el cual es un modelo de comunicación utilizado en sistemas de automatización industrial, aquí los dispositivos se organizan según roles según el intercambio de información. En esta arquitectura, uno o varios equipos actúan como servidores, proporcionando datos y servicios del proceso, mientras que los clientes acceden a dicha

información para tareas de supervisión, control o análisis. El PLC es el servidor y el HMI desarrollado es un cliente.

3.3.3. Modelo Vista-Controlador (MVC)

Este modelo se aplica en el ámbito informático como herramienta de desarrollo de software.

El enfoque MVC en el desarrollo de aplicaciones propone la división del código en componentes de modelo, vista y controlador, donde cada uno cumple funciones específicas y claramente delimitadas. Esta separación favorece un proceso de desarrollo más ordenado, facilita las pruebas del sistema y mejora el trabajo colaborativo, ya que distintos desarrolladores pueden intervenir de manera independiente en cada parte de la aplicación sin generar interferencias. Asimismo, el manejo de aplicaciones de gran tamaño se simplifica al estructurar el código en clases más compactas, con responsabilidades bien definidas, lo que contribuye a una mayor claridad, mantenibilidad y escalabilidad del software (Dempsey & Deeley, 2023).

El patrón de arquitectura de software modelo-vista-controlador (MVC) se utiliza para crear aplicaciones de escritorio y basadas en web en una amplia variedad de lenguajes de programación. MVC separa el código de la aplicación en tres áreas principales de responsabilidad:

Modelo. Los modelos organizan y gestionan los datos de la aplicación, almacenan el estado del sistema y se comunican con otras partes de la aplicación cuando se producen cambios de datos relevantes.

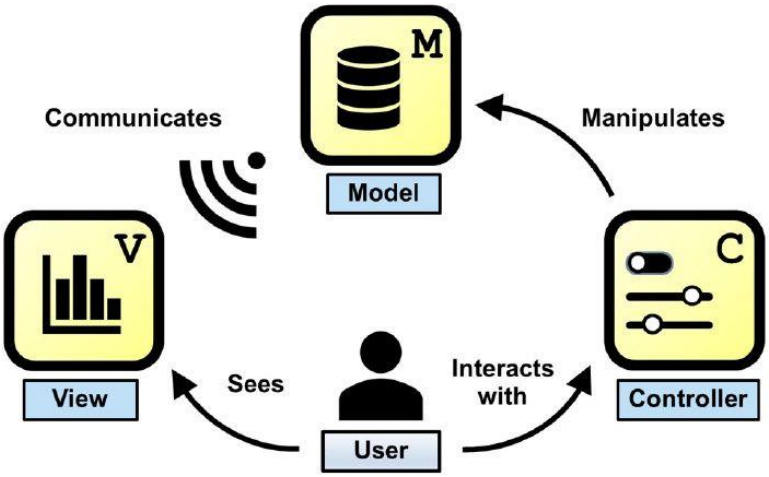
Vista. Las vistas proporcionan a los usuarios de la aplicación visualizaciones de los datos de la aplicación y el estado del sistema.

Controlador. Los controladores proporcionan mecanismos para que los usuarios de la aplicación interactúen y modifiquen los datos de la aplicación y el estado del sistema (Dempsey & Deeley, 2023).

La figura 13 ilustra el patrón de arquitectura MVC.

Figura 13

El patrón de arquitectura de software modelo-vista-controlador (MVC)



Nota: obtenido de: https://la.mathworks.com/company/technical-articles/developing-matlab-apps-using-the-model-view-controller-pattern.html

En lo referente al sistema HMI, basándose en una arquitectura: Modelo-Vista-Controlador. La **Vista** consiste en el desarrollo de interfaz de usuario, se crearán tres pantallas, la figura 14 corresponde al interfaz de control analógico, la figura 15 al control booleano y la figura 16 al registrador de datos.

Figura 14

Pantalla de control y monitoreo de variables analógicas

Figura 15

Pantalla de control y monitoreo de variables booleanas

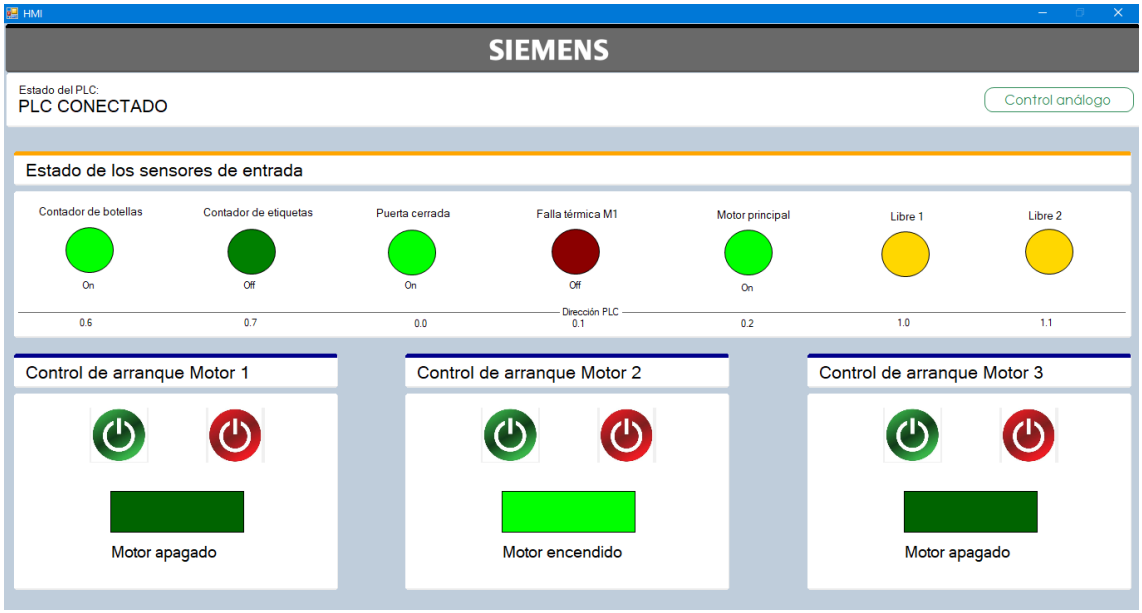
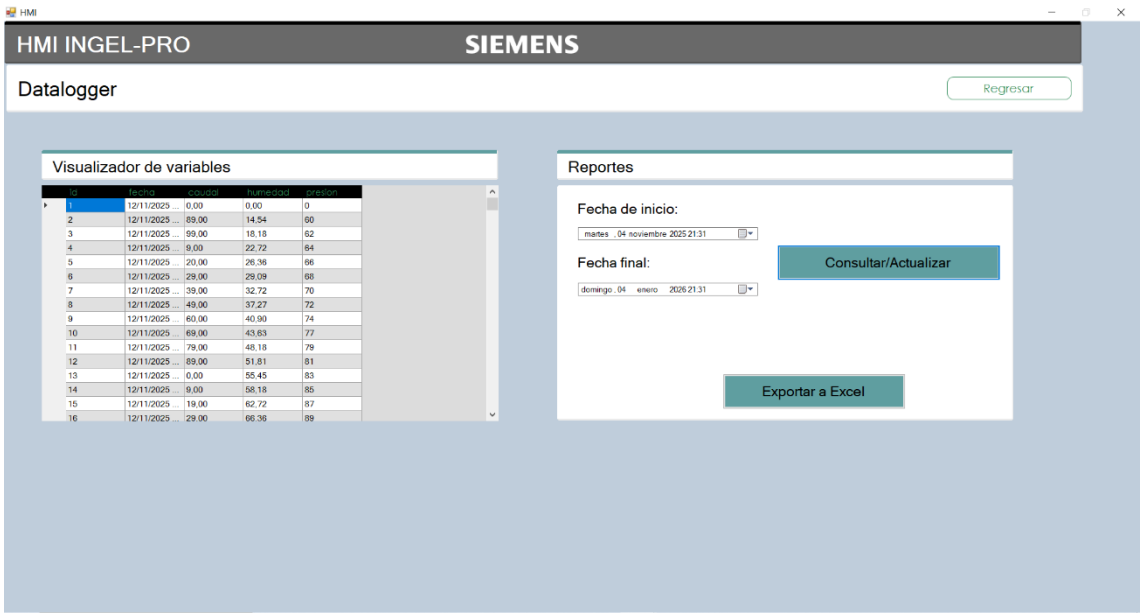


Figura 16

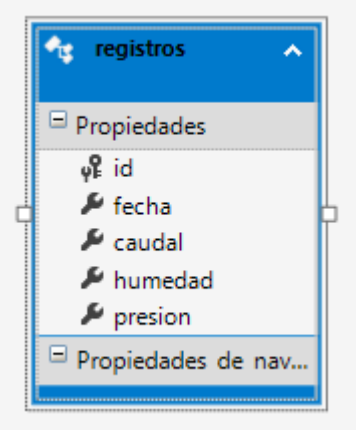
Datalogger



El **Modelo** equivale a la Base de Datos, las tablas de la Base de Datos para los sistemas de producción por lo general no requieren de llaves foráneas, por tanto, la tabla para el datalogger es simple, como lo muestra la figura 17.

Figura 17

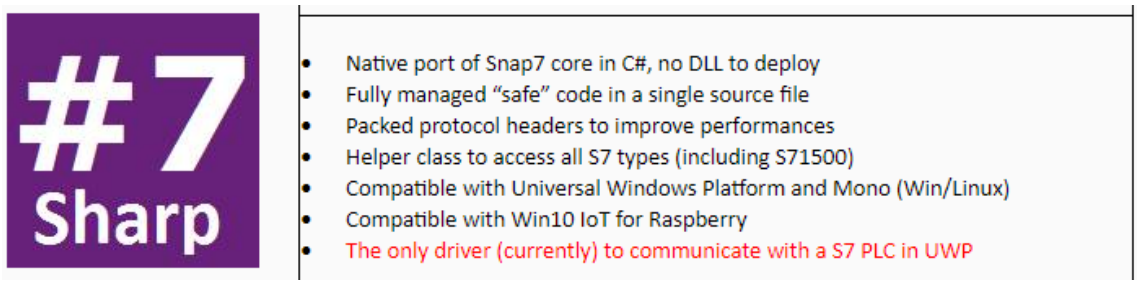
Tabla del Datalogger



El **Controlador** es una Clase de C# que administra los enlaces hacia el PLC, el componente principal es un paquete Nuget o Framework de nombre SHARP7, el cual también se puede obtener de la página oficial: http://snap7.sourceforge.net/. La figura 18 corresponde al logo de la librería y sus características.

Figura 18

Librería Sharp7 para comunicación con el PLC



La librería Sharp7 es de uso gratuito.

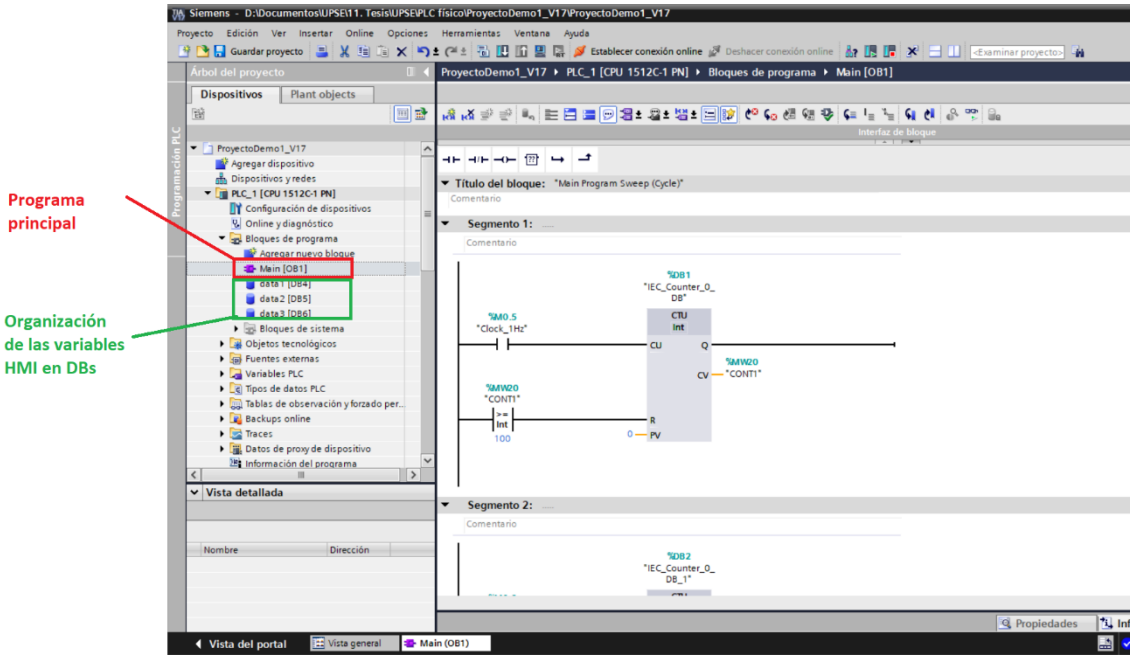
3.4. Desarrollo

3.4.1. Programación del PLC

La programación del PLC puede ser realizada en cualquier lenguaje para fines de comunicación al HMI; en lo referente a la forma de programar es más óptimo usar programación estructurada, como lo indica la figura 19, para facilitar la organización de las variables que tendrán comunicación al HMI.

Figura 19

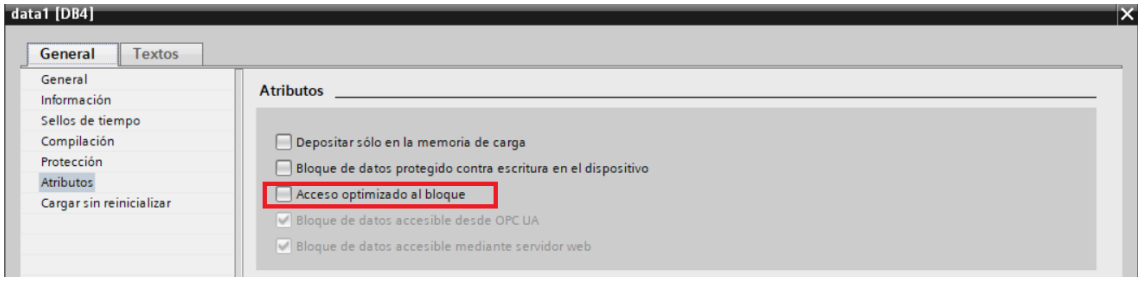
Estructura general del programa del PLC



Los DBs (bloque de datos) organizan las variables a comunicarse con el HMI, ya que la comunicación se realiza con una aplicación ajena a Siemens es obligatorio retirar el acceso optimizado al bloque, como lo muestra la figura 20.

Figura 20

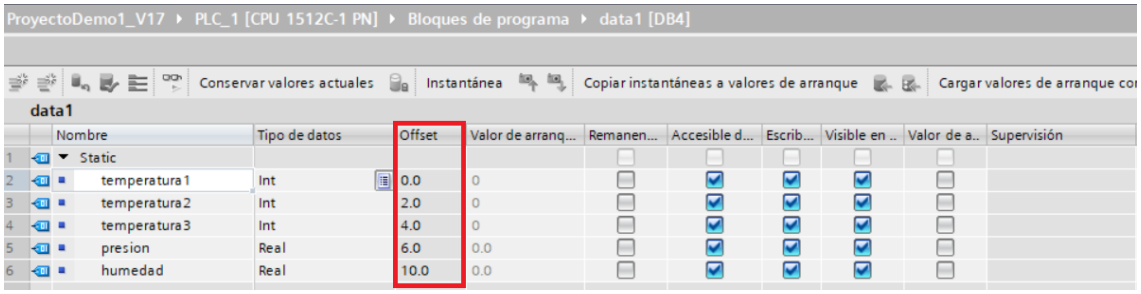
Acceso optimizado al bloque desactivado



Al retirar el acceso optimizado al bloque de cada DB se puede tener acceso a la dirección de red, requisito necesario para el enlace con el HMI desarrollado en Visual Studio, en la figura 21 se muestra esta dirección en la columna Offset.

Figura 21

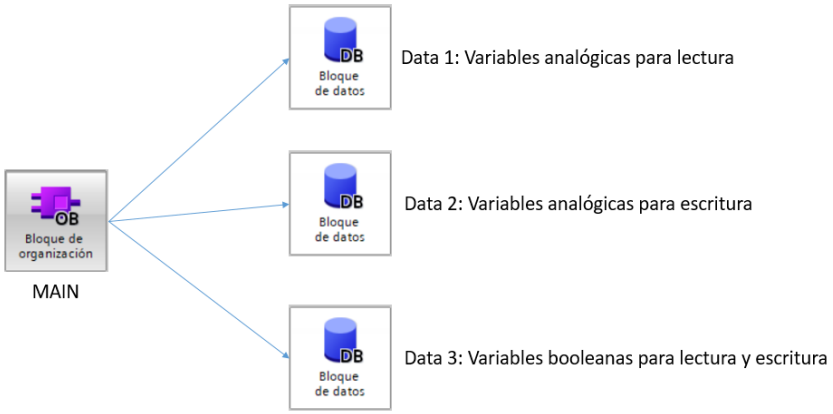
Direcciones de comunicación al HMI



La estructuración del programa se resume en la figura 22.

Figura 22

Estructuración del programa del PLC

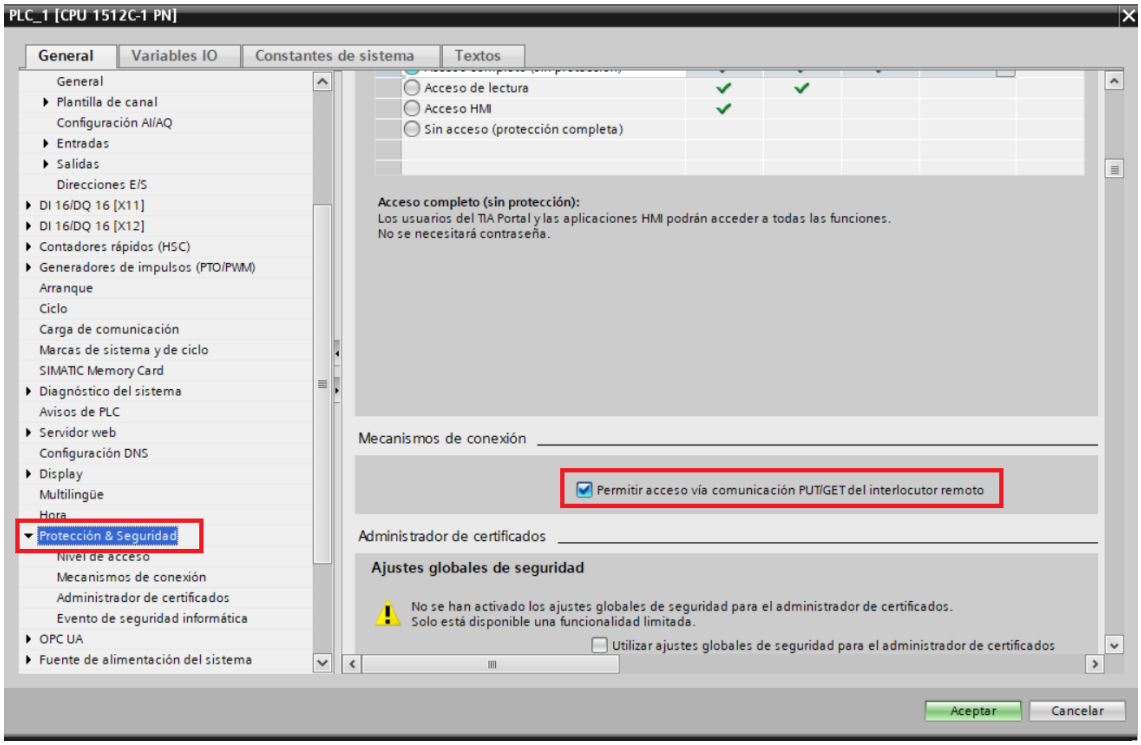


Esta estructuración es útil no solamente para el programa de prueba sino también para futuros proyectos de la empresa.

Como se está usando el protocolo S7 para comunicarse con el HMI, es obligatorio en el PLC habilitar dicha comunicación como se ilustra en la figura 23.

Figura 23

Habilitación de la comunicación S7en el PLC



Esta habilitación se realiza en la ventana de propiedades del PLC.

3.4.2. Desarrollo HMI en Visual Studio con el lenguaje C#

El "motor" base de la aplicación HMI es .NET., la cual es una plataforma de código abierto para crear aplicaciones de escritorio, web y móviles que se pueden ejecutar de forma nativa en cualquier sistema operativo. El sistema .NET incluye herramientas, bibliotecas y lenguajes que admiten el desarrollo de software moderno, escalable y de alto rendimiento. Una comunidad de desarrolladores activa mantiene y apoya la plataforma .NET. (Amazon Web Services, Inc., 2025).

En lo referente a implementaciones con .NET se tiene:

- .NET Framework.
- .NET Core.
- .NET Estándar.

Para el desarrollo HMI se usa .NET Framework debido a que según Amazon Web Services, Inc. (2025) es la implementación de .NET original. Admite la ejecución de sitios

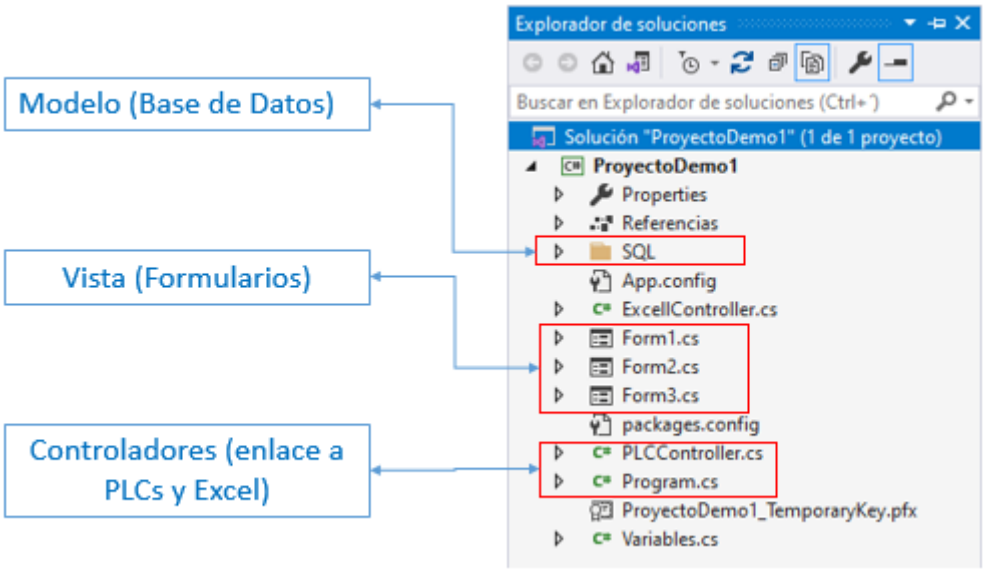
web, servicios, aplicaciones de escritorio y más en Windows. Microsoft lanzó .NET Framework a principios de la década de 1990.

3.4.2.1. Estructura del programa en Visual Studio

En Visual Studio, el desarrollo de aplicaciones se organiza de manera jerárquica mediante el uso de soluciones y proyectos, lo que permite una gestión estructurada y ordenada del código fuente. Una solución actúa como un contenedor principal que agrupa uno o varios proyectos relacionados, facilitando la administración conjunta de archivos, configuraciones y dependencias. Por su parte, cada proyecto corresponde a una aplicación o componente específico dentro de la solución, donde se definen el tipo de aplicación, el lenguaje de programación utilizado y los recursos necesarios para su compilación y ejecución. Esta estructura favorece la modularidad del software, simplifica el mantenimiento del código y permite el desarrollo colaborativo, ya que distintos proyectos pueden interactuar entre sí dentro de una misma solución, optimizando así el proceso de desarrollo de aplicaciones en Visual Studio. La figura 24 muestra la estructura del programa realizado para el HMI.

Figura 24

Estructura de la solución en Visual Studio



La figura 24 hace alusión a la estructura final de la solución, en los siguientes apartados se describe el proceso para llegar a la mencionada estructura.

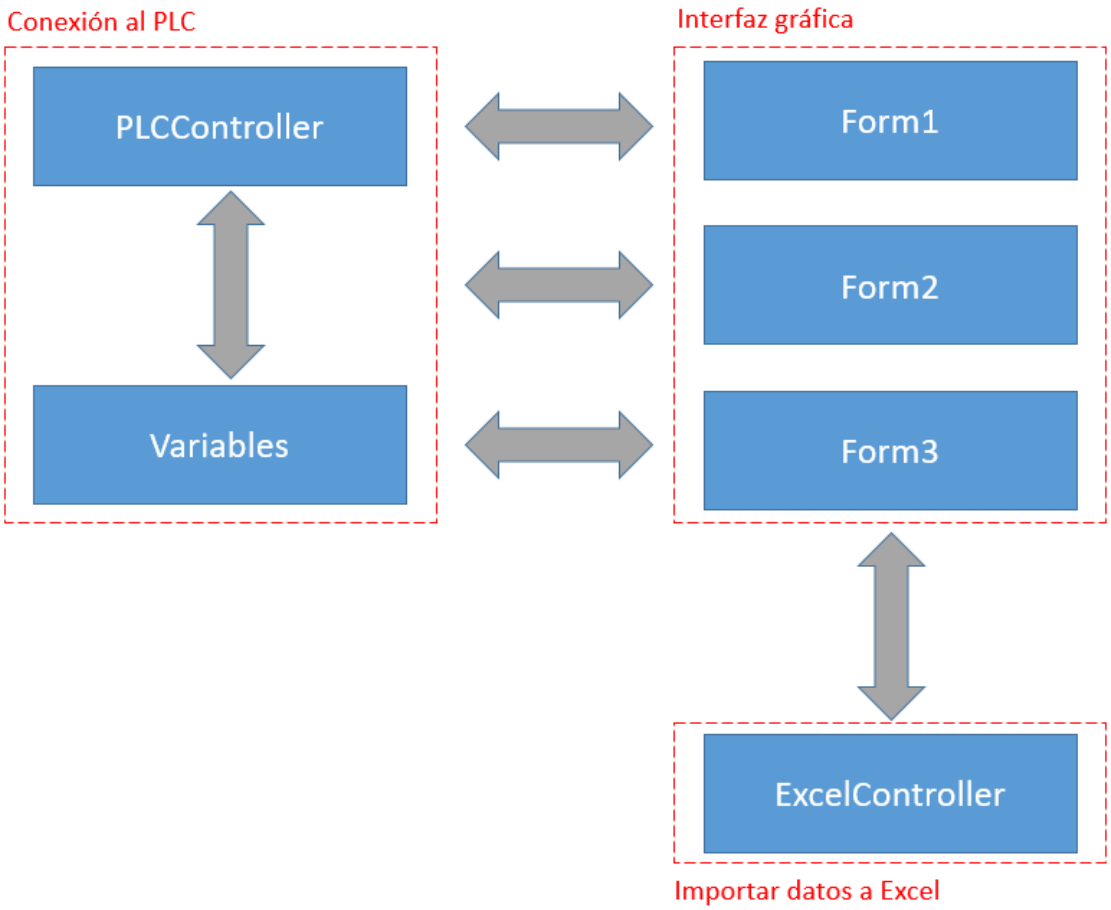
3.4.2.2. Creación de clases

Una clase en C# es un tipo de dato que define un conjunto de campos, propiedades, métodos y eventos que representan un concepto o entidad en el mundo real. Las clases nos permiten encapsular datos y funcionalidades relacionadas en una sola entidad, lo que facilita la reutilización de código y el mantenimiento de nuestras aplicaciones. Son la base de la programación orientada a objetos (Llamas, 2026, párr. 1).

Las clases que usa el proyecto siempre serán las mismas considerando que el enfoque es una plantilla reutilizable, en la figura 25 se ilustra las clases necesarias.

Figura 25

Clases creadas y su aplicación



Las clases se intercomunican entre sí, por ejemplo, en el interfaz gráfico que corresponde a los Formularios o “Forms” es posible visualizar o escribir una determinada variable gracias a que hay un contenedor de variables (clase variables) y sobre todo, a que existe una clase encargada de la comunicación (PLCController).

Lo que varía de proyecto a proyecto es la cantidad de clases, pero siempre será la misma estructura. A manera de ejemplo, si en otro proyecto hay seis pantallas se deberá aumentar de tres a seis Forms.

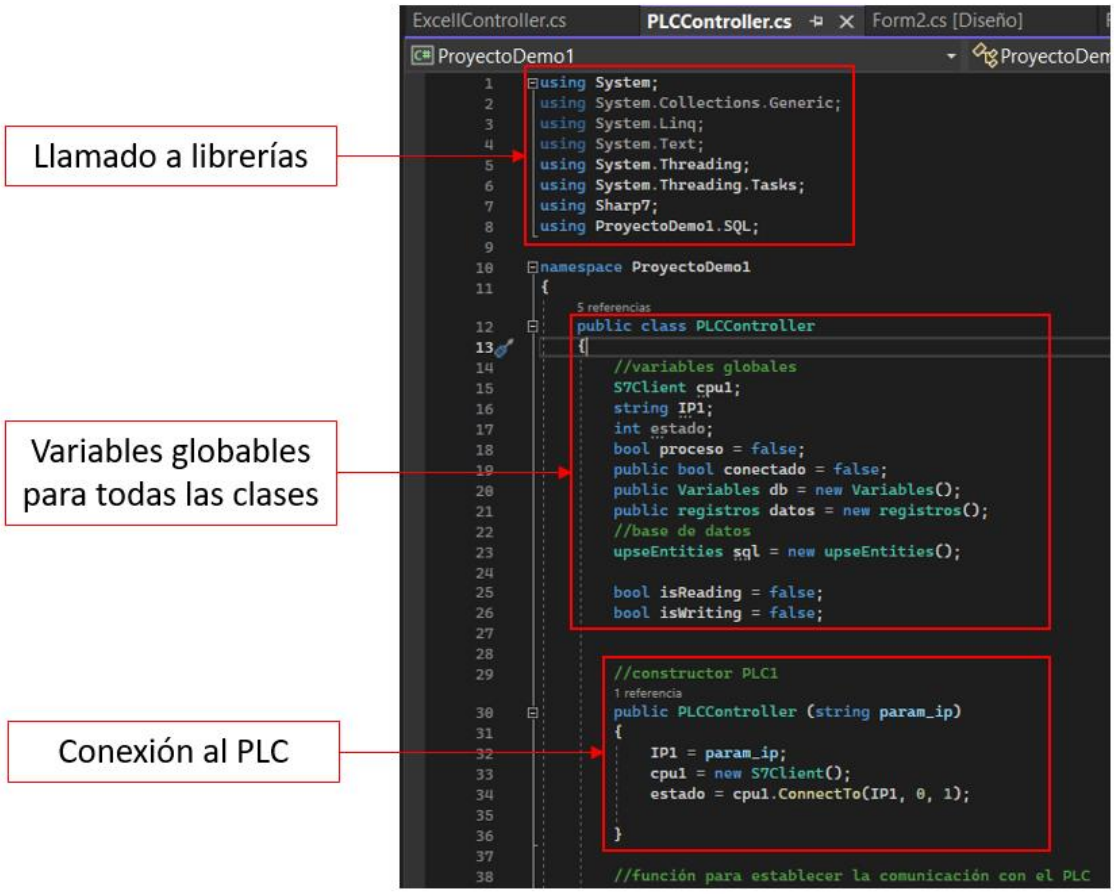
Los nombres de las clases son dadas por el programador, al ser una plantilla se recomienda usar los mostrados en la figura 25.

3.4.2.3. Clase para comunicación y administración de lectura/escritura

La clase PLCController tiene la estructura que indica la figura 26.

Figura 26

Clase para comunicación con el PLC



Inicia con el llamado a librerías, en donde se encuentra la librería de comunicación Sharp7, posterior se crea las variables encargadas de monitorear constantemente la comunicación, luego se realiza conexión al PLC indicando su IP, número de rack y slot.

Esta misma clase también es la encargada de la lectura/escritura de variables, el llamado a la base de datos y la función para trabajo multitarea.

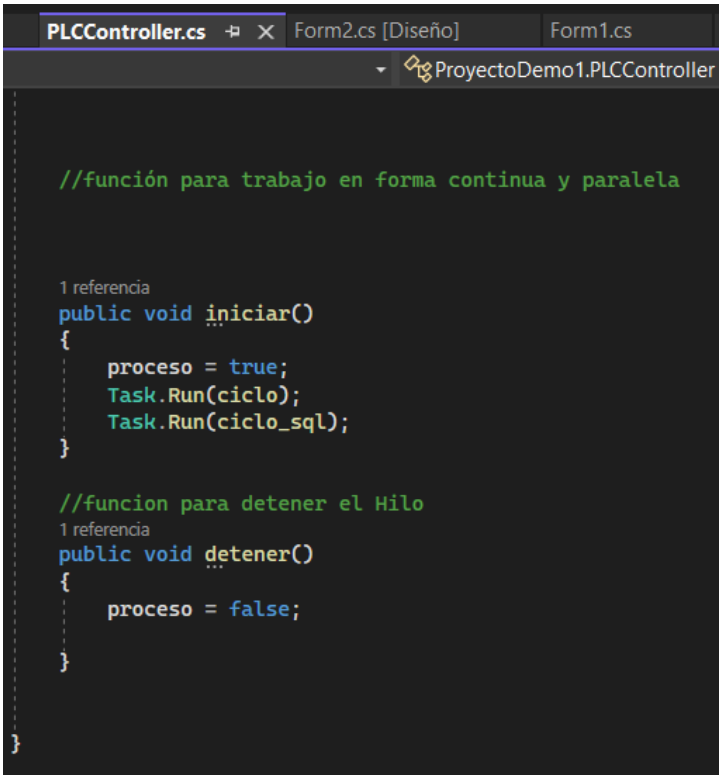
3.4.2.4. Hilos en C#

En C#, los threads (hilos) son una forma de conseguir la programación concurrente y paralela. Nos permiten dividir la ejecución de un programa en múltiples tareas que pueden correr simultáneamente (Llamas, 2026, párr. 1).

La clase PLCController administra los hilos del programa como se muestra en la figura 27.

Figura 27

Hilos del programa



El funcionamiento multitarea que administra esta clase es fundamental para lograr que la aplicación finalmente no tenga errores cuando por ejemplo se está leyendo una variable

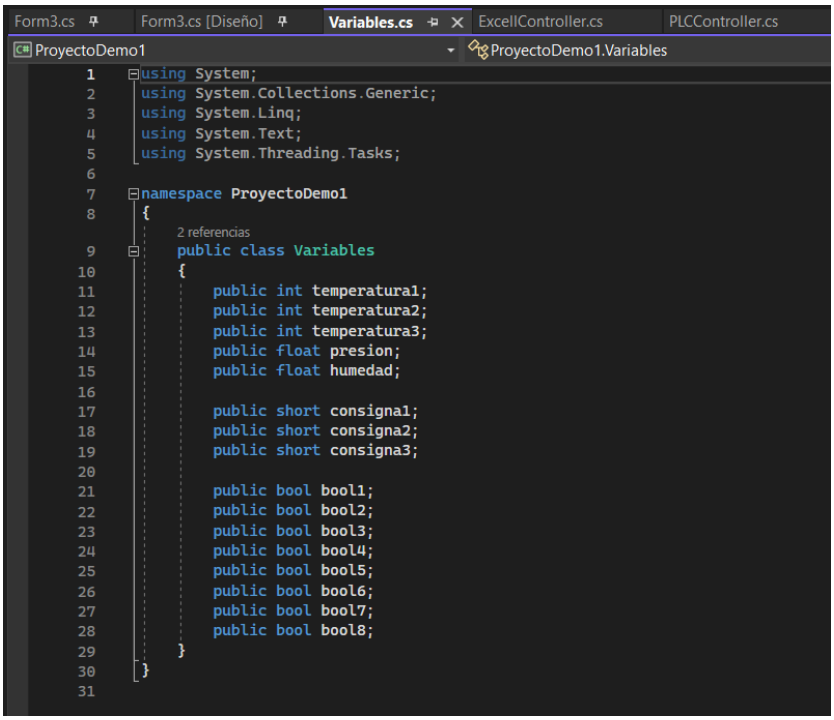
y súbitamente se desconecta el PLC, esto es dos tareas simultáneas, sin los hilos el programa falla, comúnmente conocido con el término “se cuelga el programa”.

3.4.2.5. Clase contenedora de variables

En esta clase se colocan las variables a comunicar con el PLC (figura 28), para un proyecto diferente en esta clase se deberá modificar según lo que se requiera.

Figura 28

Variables del PLC en C#



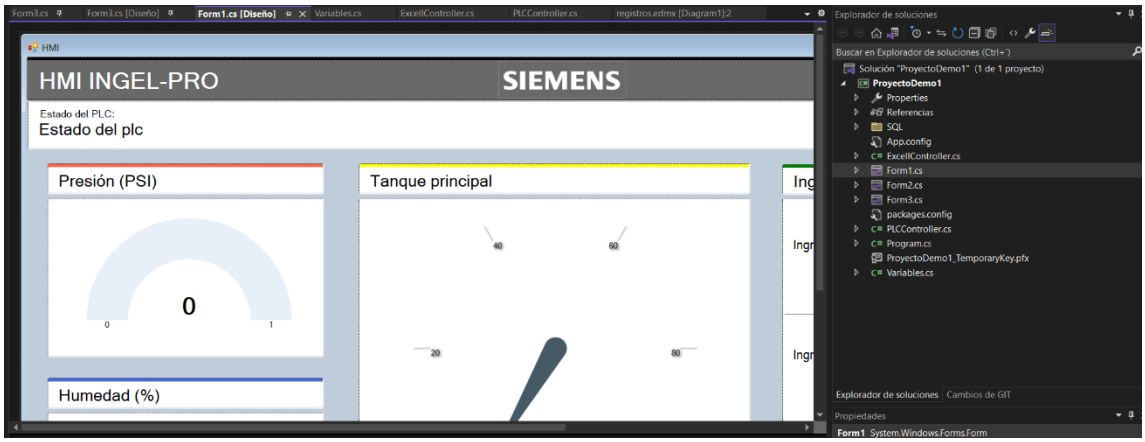
Esta clase es del tipo pública para que las demás clases puedan acceder a ella y sus variables. El tipo de variable tiene que coincidir con el tipo de variable del PLC mientras que su nombre puede ser independiente.

3.4.2.5. Clase formulario

El formulario es una ventana interactiva gráfica para aplicaciones de escritorio de Windows, sin embargo, sigue siendo una clase y en este tipo de clases se realiza el interfaz HMI (figura 29).

Figura 29

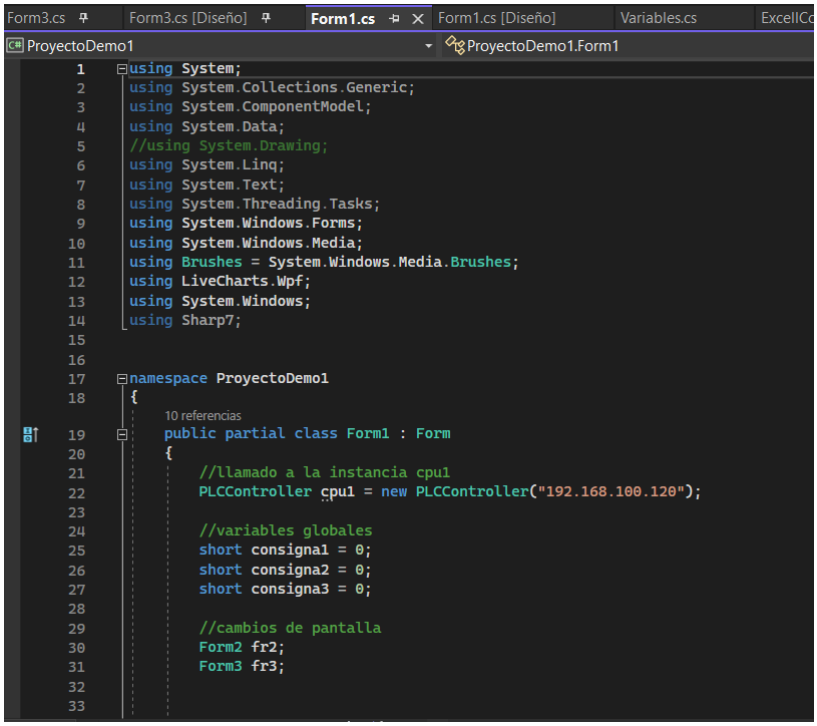
Vista de formulario en el proyecto



La figura 30 muestra como es un formulario en el ambiente de desarrollo de Visual Studio. Los formularios disponen también se su vista en código como lo muestra la figura 31.

Figura 30

Vista de código del formulario



Por tanto, se puede decir que el formulario tiene un “front-end” que corresponde al interfaz gráfico y un “back-end” que corresponde a la programación del mismo.

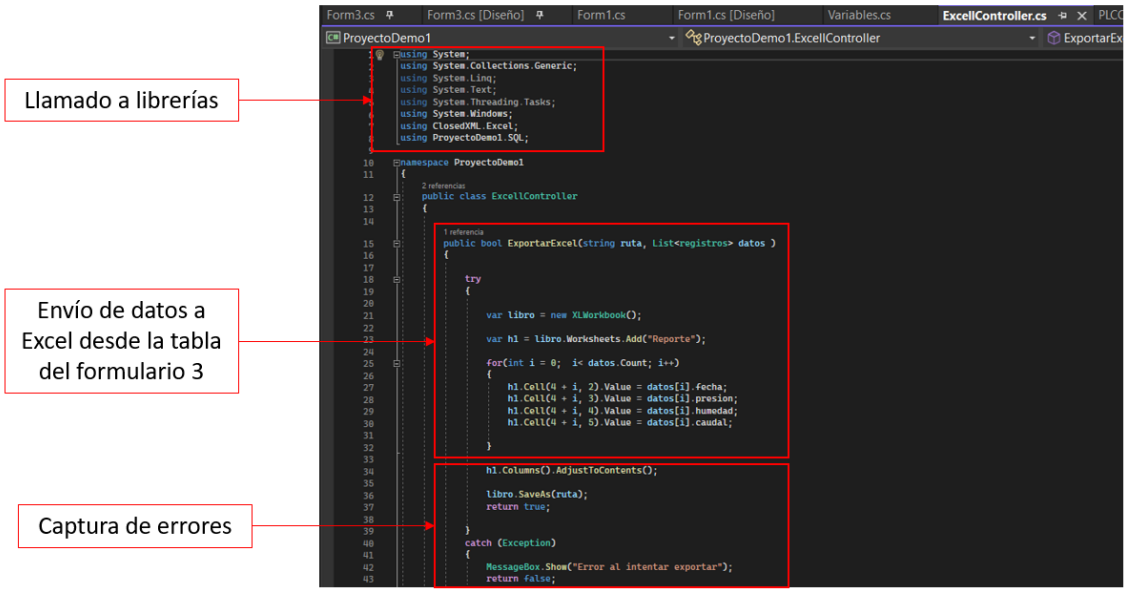
El programa desarrollado consta de tres pantallas HMI, por tanto, tiene tres clases de formulario.

3.4.2.6. Clase para enlace a Excel

Esta clase lee las filas y columnas de la tabla de consulta de la base de datos y los exporta directamente en formato Excel. La figura 31 muestra su estructura.

Figura 31

Estructura de la clase de enlace a Excel



Cuando se requiera aplicar esta clase a un proyecto diferente simplemente se tiene que modificar el contenido del ciclo for por las nuevas variables.

3.4.3. Vinculación de base de datos al HMI

La inclusión de una base de datos en un sistema HMI industrial es fundamental para disponer de la información generada durante la operación de los procesos industriales. Mediante el almacenamiento de valores de variables, la base de datos permite el análisis de tendencias, la trazabilidad de la producción y la detección temprana de fallos,

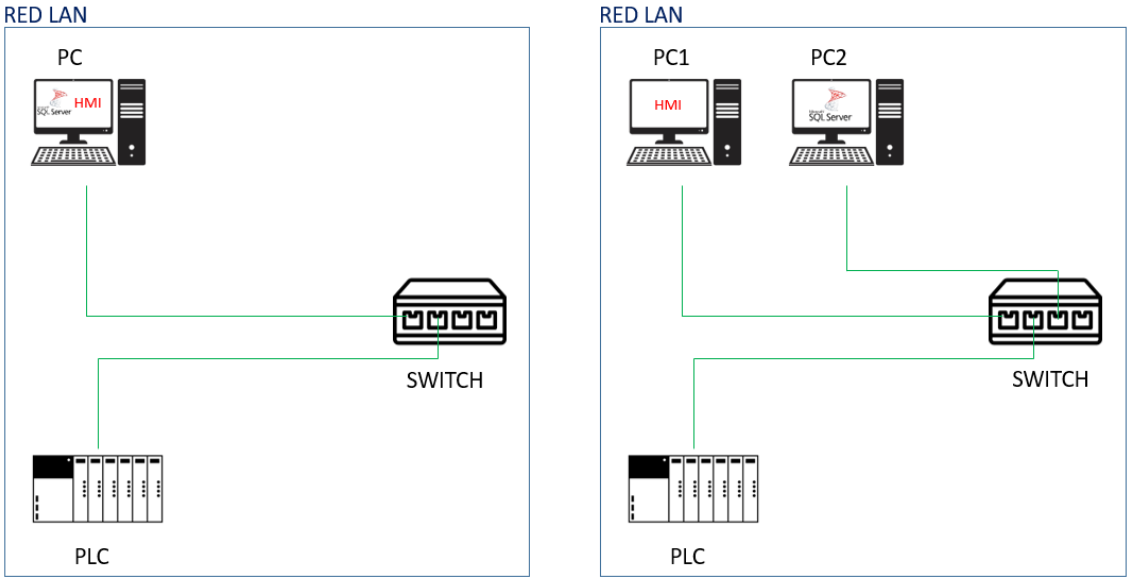
facilitando la toma de decisiones basada en datos reales. Además, posibilita la generación de reportes, el cumplimiento de requisitos de calidad y auditorías, así como la integración del HMI con otros sistemas de nivel superior, como MES o ERP.

Se escoge a Microsoft SQL Server como motor de base de datos del sistema HMI, esto en virtud de que la marca Siemens trabaja tradicionalmente con Microsoft para integración a base datos.

La arquitectura de red de la base de datos se muestra en la figura 32, en donde se indica que la base puede estar en la misma PC en la que corre el HMI o puede estar en otra PC que se encuentre en la misma red.

Figura 32

Arquitectura de red para la base de datos



Ya que el desarrollo se enfoca en la creación de una plantilla, al momento de implementar uno u otro proyecto se decidirá la ubicación de la base de datos.

Para realizar la integración de la base de datos al HMI se realiza los siguientes pasos, partiendo de que la base de datos y su administrador ya se encuentra instalado en la PC deseada.

- Crear la base de datos, sus tablas y variables.
- Conectar la base de datos a C#.
- Diseño del interfaz gráfico en el HMI para la base de datos.

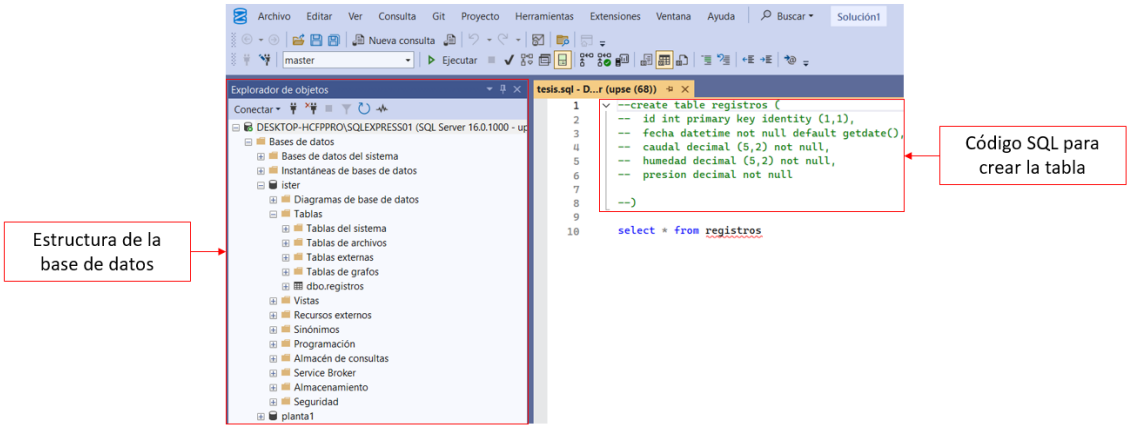
3.4.3.1. Creación de la base de datos

La base de datos se crea en función del proyecto, es decir, según el proyecto se determina la cantidad de variables a almacenar, así como su tipo. Para esta plantilla base se almacenarán tres variables provenientes del PLC que son caudal, humedad, presión.

La figura 33 muestra el software SQL Server Management Studio, con el código SQL que se usa para crear la tabla del proyecto HMI.

Figura 33

Creación de la tabla de la base de datos

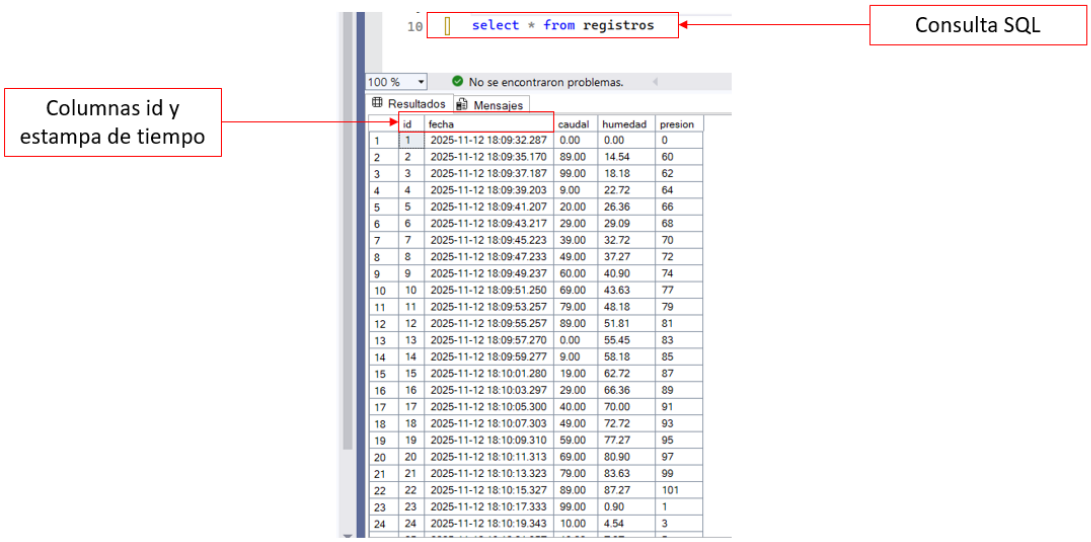


En sistemas industriales las bases de datos son más sencillas, por lo general no se crean varias tablas y se usan llaves primarias o foráneas para vincularlas.

Al momento de crear las columnas para cualquier caso siempre es importante crear dos columnas adicionales que corresponden al id y a la estampa de tiempo como se muestra en la figura 34.

Figura 34

Consulta de la base de datos



	id	fecha	caudal	humedad	presion
1	1	2025-11-12 18:09:32.287	0.00	0.00	0
2	2	2025-11-12 18:09:35.170	89.00	14.54	60
3	3	2025-11-12 18:09:37.187	99.00	18.18	62
4	4	2025-11-12 18:09:39.203	9.00	22.72	64
5	5	2025-11-12 18:09:41.207	20.00	26.36	66
6	6	2025-11-12 18:09:43.217	29.00	29.09	68
7	7	2025-11-12 18:09:45.223	39.00	32.72	70
8	8	2025-11-12 18:09:47.233	49.00	37.27	72
9	9	2025-11-12 18:09:49.237	60.00	40.90	74
10	10	2025-11-12 18:09:51.250	69.00	43.63	77
11	11	2025-11-12 18:09:53.257	79.00	48.18	79
12	12	2025-11-12 18:09:55.257	89.00	51.81	81
13	13	2025-11-12 18:09:57.270	0.00	55.45	83
14	14	2025-11-12 18:09:59.277	9.00	58.18	85
15	15	2025-11-12 18:10:01.280	19.00	62.72	87
16	16	2025-11-12 18:10:03.297	29.00	66.36	89
17	17	2025-11-12 18:10:05.300	40.00	70.00	91
18	18	2025-11-12 18:10:07.303	49.00	72.72	93
19	19	2025-11-12 18:10:09.310	59.00	77.27	95
20	20	2025-11-12 18:10:11.313	69.00	80.90	97
21	21	2025-11-12 18:10:13.323	79.00	83.63	99
22	22	2025-11-12 18:10:15.327	89.00	87.27	101
23	23	2025-11-12 18:10:17.333	99.00	0.90	1
24	24	2025-11-12 18:10:19.343	10.00	4.54	3

La figura 34 además muestra datos de las variables caudal, humedad y presión las cuales vienen desde el PLC, su escritura de muestra a continuación.

3.4.3.2. Conexión de la base de datos con C#

Tradicionalmente, la integración entre aplicaciones desarrolladas en C# y bases de datos SQL Server se realizaba mediante el uso directo de ADO.NET, esto implicaba la escritura manual de consultas SQL, la realización correcta de conexiones y consultas SQL. Esto incrementaba la complejidad del código y el tiempo de desarrollo.

En la actualidad, este proceso ha evolucionado con la adopción de frameworks de mapeo objeto-relacional como Entity Framework, el cual permite interactuar con la base de datos a través de clases y objetos del lenguaje C#, simplificado en forma general todo el proceso.

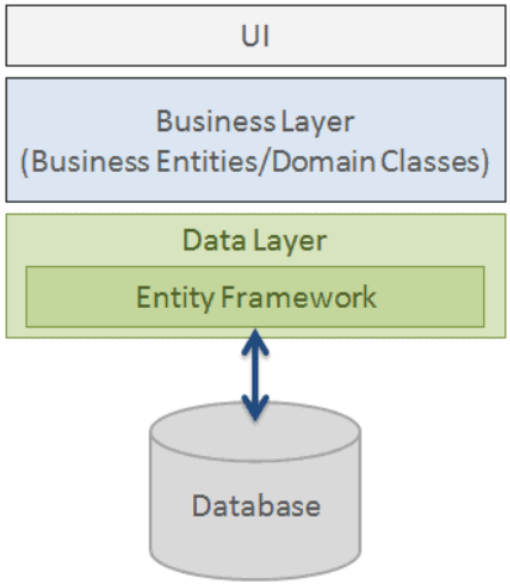
Entity Framework es un framework ORM de código abierto para aplicaciones .NET, compatible con Microsoft. Permite a los desarrolladores trabajar con datos mediante objetos de clases específicas del dominio, sin centrarse en las tablas y columnas de la base de datos subyacente donde se almacenan dichos datos. Con Entity Framework, los desarrolladores pueden trabajar con un mayor nivel de abstracción al gestionar datos y

crear y mantener aplicaciones orientadas a datos con menos código que las aplicaciones tradicionales (¿Qué es Entity Framework?, 2025).

La figura 35 ilustra dónde encaja Entity Framework en la aplicación.

Figura 35

Estructura Entity Framework



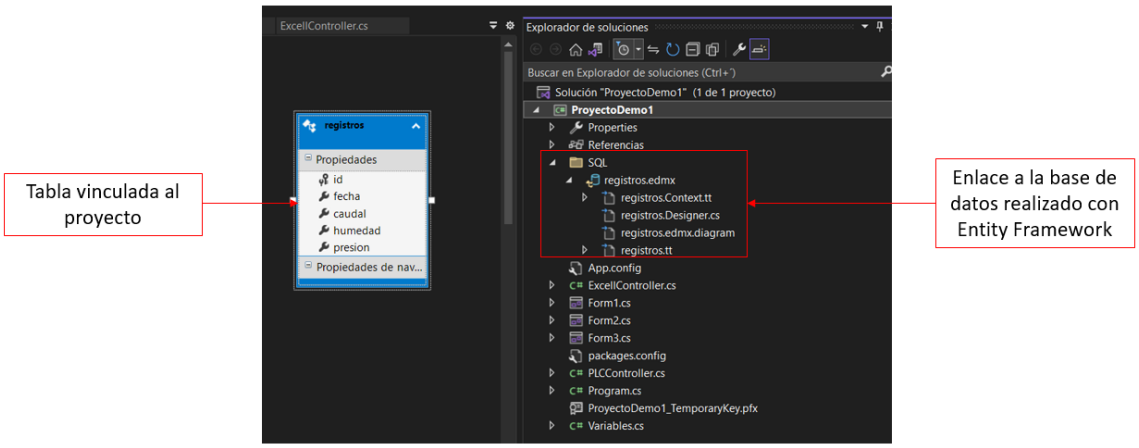
Nota: información obtenida de: https://www.entityframeworktutorial.net/entityframework6/what-is-entityframework.aspx

Según la figura anterior, Entity Framework se integra entre las entidades de negocio (clases de dominio) y la base de datos. Guarda los datos almacenados en las propiedades de las entidades de negocio y, además, recupera datos de la base de datos y los convierte automáticamente en objetos de entidad de negocio (¿Qué es Entity Framework?, 2025).

En el presente proyecto la tabla perteneciente a la base de datos que se muestra en la figura 34 se vincula utilizado Entity Framework, obteniendo el resultado que se indica en la figura 36.

Figura 36

Entity Framework en el proyecto HMI



La vinculación de C# con la base de datos finalmente es un proceso muy conocido, sobre todo si se usa el Framework, lo relevante es disponer de un interfaz gráfico que permita realizar consultas a la base de datos, para ello la clase “Form3” se encarga de realizar este proceso.

El interfaz gráfico de este formulario dispone de una tabla para visualizar los datos de las variables y una sección de reportes para colocar la fecha de inicio y fin de la consulta a realizar. La figura 18 muestra la realización de una consulta en forma gráfica. La sección Reportes también incluye el botón para exportar los datos consultados a formato Excel.

3.5. Pruebas

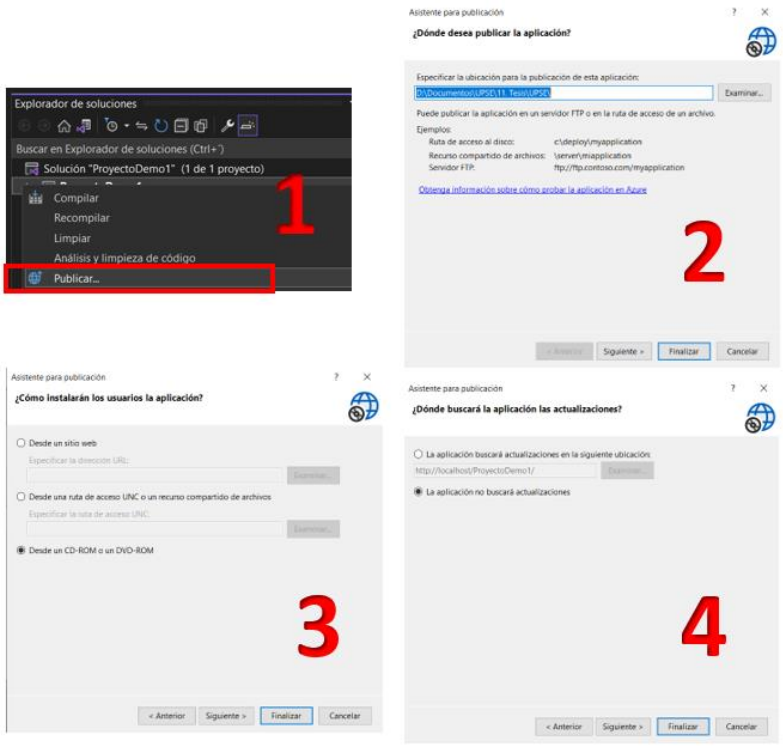
El producto final consiste en entregar al cliente el instalador del software HMI, o en términos de HMI SCADA tradicional el “Runtime” de la aplicación. Para este fin Visual Studio dispone de la opción “Publicar” que genera un instalador de la aplicación

3.5.1. Creación del instalador “Runtime”

Las pruebas por tanto se realizan con el ejecutable de la aplicación más no en el entorno de desarrollo de Visual Studio. La figura 37 muestra los pasos para crear el ejecutable de la aplicación.

Figura 37

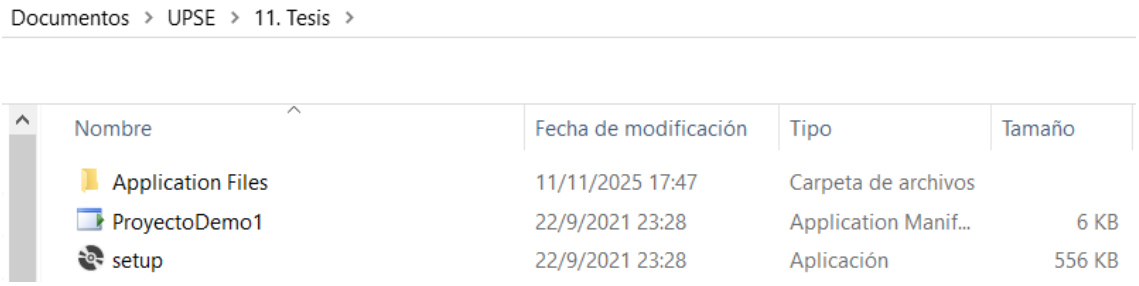
Proceso de creación del ejecutable de la aplicación



Una vez que concluye el asistente para la creación del ejecutable se crea la estructura de carpetas que se indica en la figura 38.

Figura 38

Ejecutable generado



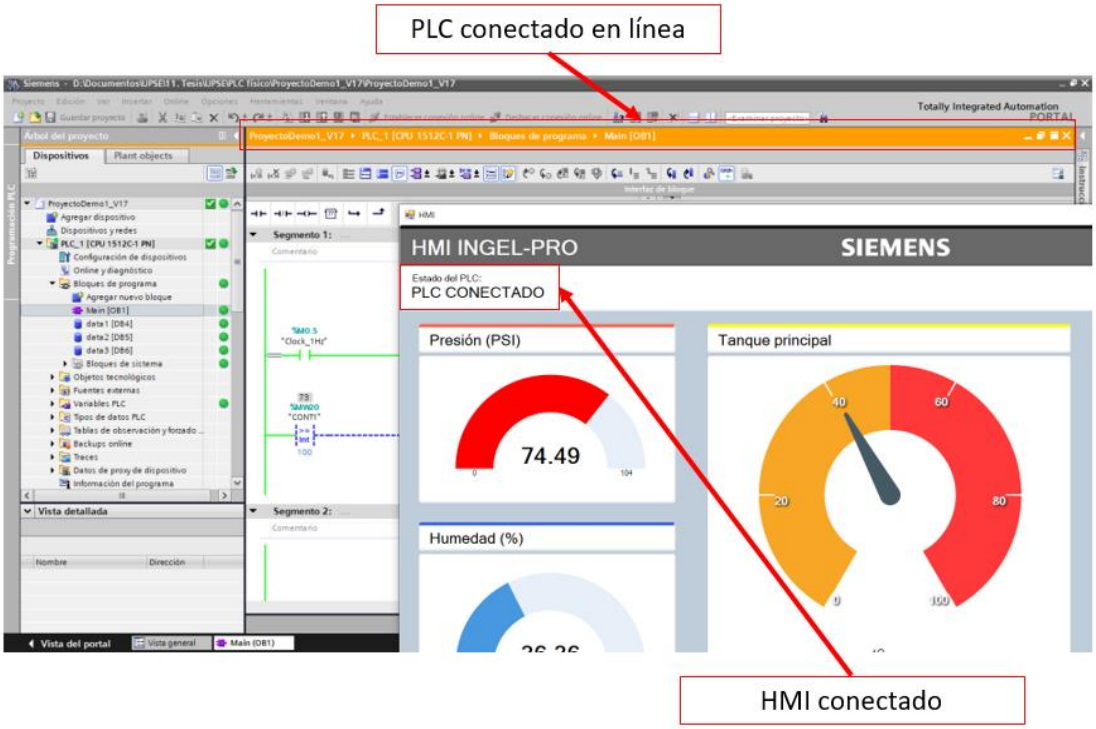
Para instalar la aplicación de Windows basta con dar doble clic en el archivo “setup”, se abrirá una ventana de instalación en donde basta con presionar el botón “Instalar”.

3.5.2. Prueba de conexión

Para iniciar las pruebas del software HMI, lo primero es garantizar que se conecte al PLC, para ello el interfaz del HMI está diseñado con un aviso textual que indica si el PLC está conectado o desconectado, la figura 39 ilustra el caso en que el PLC está encendido, se encuentra conectado por red ethernet a la PC.

Figura 39

Aviso de PLC y HMI conectados



En la figura 39 se aprecia que el PLC se encuentra conectado en línea (la franja de color naranja indica que está en línea) y como ventana sobrepuesta se encuentra el HMI desarrollado en donde se aprecia el mensaje “PLC CONECTADO”.

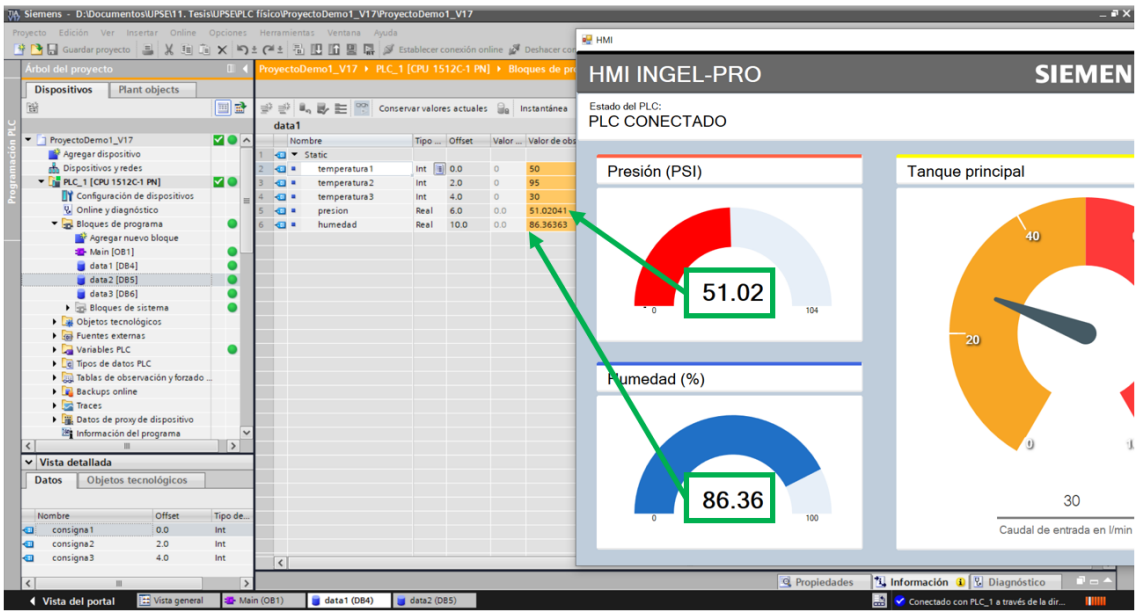
Se comprueba también que sucede si se apaga el PLC o se desconecta el cable de red, al realizar cualquiera de estas dos acciones el mensaje cambia a “PLC DESCONECTADO” dando al usuario información de primera mano del estado de la conexión.

3.5.3. Prueba de lectura y escritura de variables

Utilizando la conexión en línea al PLC es posible comprobar en tiempo real la lectura de variables del PLC en el HMI y la escritura de variables desde el HMI hacia el PLC. La figura 40 muestra que el valor del HMI es el mismo de las variables del PLC.

Figura 40

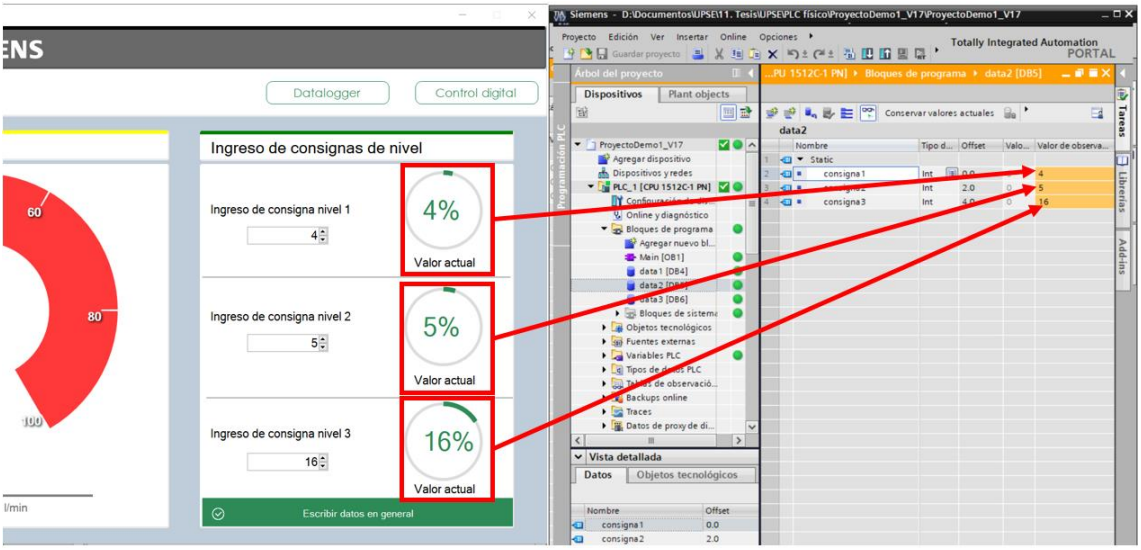
Prueba de lectura de variables analógicas



La prueba de escritura de variables analógicas se realiza de manera análoga a la lectura, es decir con una conexión en línea al PLC y la vista del interfaz del HMI. La figura 41 grafica el resultado de la escritura.

Figura 41

Prueba de escritura de variables analógicas

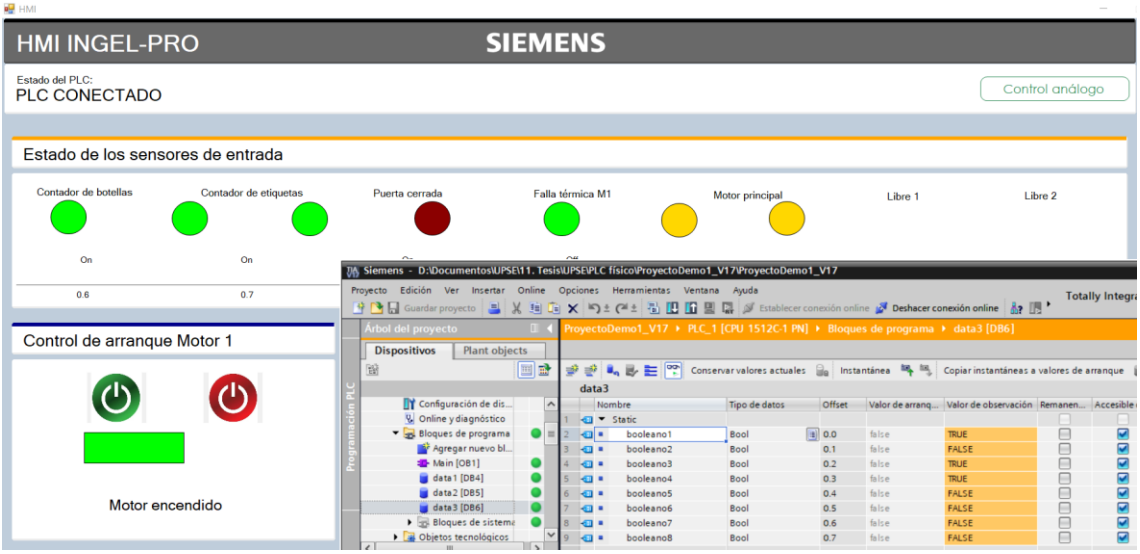


La figura 41 se enfoca en la prueba de escritura de variables analógicas tipo enteras.

La prueba de lectura y escritura de variables tipo booleanas se realiza de la misma forma, realizando una conexión en línea al PLC y observando el interfaz gráfico del HMI. La figura 42 evidencia la conexión de estas variables.

Figura 42

Prueba de lectura y escritura de variables booleanas



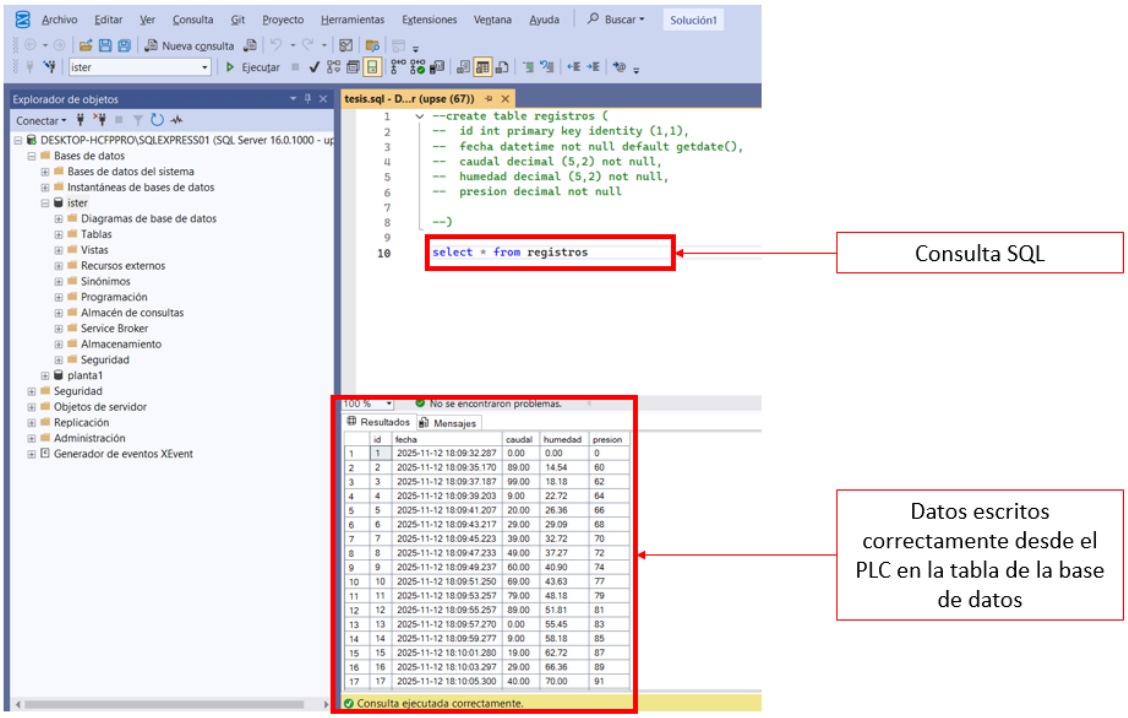
En el proyecto el Form2 contiene el interfaz gráfico para lectura y escritura de variables tipo booleanas.

3.5.4. Prueba en la base de datos

Utilizando el software de administración de la base de datos, SQL Server Management Studio, y con instrucciones SQL se realiza una consulta a la tabla creada anteriormente como se muestra en la figura 43.

Figura 43

Prueba de escritura de variables del PLC en la tabla de la base de datos

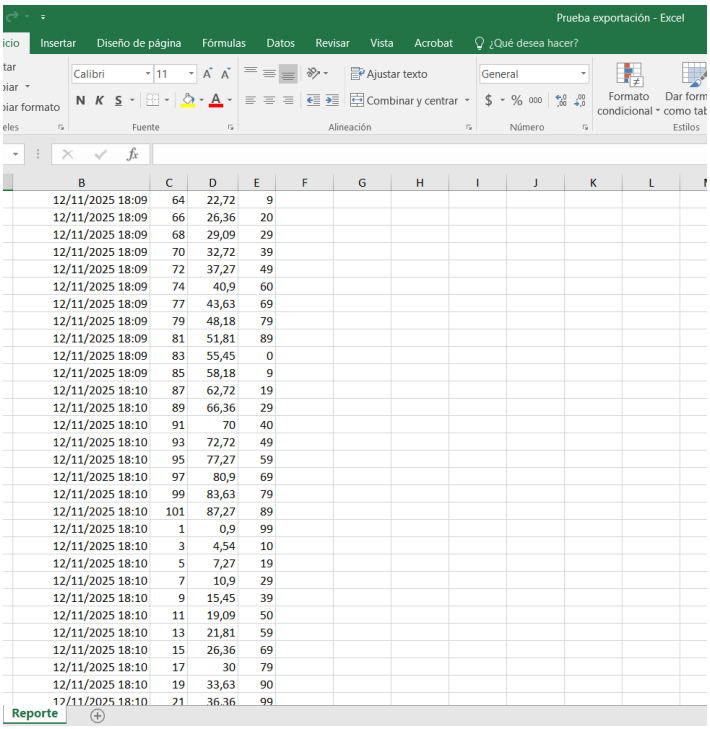


En el diseño HMI el Form3 es aquel que contiene el interfaz gráfico para realizar la consulta de los datos que se encuentran en la base de datos.

Como complemento de la consulta de la base de datos, el Form3 dispone de la opción para exportar los datos a Excel. La figura 44 muestra el resultado de la exportación de datos.

Figura 44

Prueba de exportación de datos consultados a Excel



B	C	D	E
12/11/2025 18:09	64	22,72	9
12/11/2025 18:09	66	26,36	20
12/11/2025 18:09	68	29,09	29
12/11/2025 18:09	70	32,72	39
12/11/2025 18:09	72	37,27	49
12/11/2025 18:09	74	40,9	60
12/11/2025 18:09	77	43,63	69
12/11/2025 18:09	79	48,18	79
12/11/2025 18:09	81	51,81	89
12/11/2025 18:09	83	55,45	0
12/11/2025 18:09	85	58,18	9
12/11/2025 18:10	87	62,72	19
12/11/2025 18:10	89	66,36	29
12/11/2025 18:10	91	70	40
12/11/2025 18:10	93	72,72	49
12/11/2025 18:10	95	77,27	59
12/11/2025 18:10	97	80,9	69
12/11/2025 18:10	99	83,63	79
12/11/2025 18:10	101	87,27	89
12/11/2025 18:10	1	0,9	99
12/11/2025 18:10	3	4,54	10
12/11/2025 18:10	5	7,27	19
12/11/2025 18:10	7	10,9	29
12/11/2025 18:10	9	15,45	39
12/11/2025 18:10	11	19,09	50
12/11/2025 18:10	13	21,81	59
12/11/2025 18:10	15	26,36	69
12/11/2025 18:10	17	30	79
12/11/2025 18:10	19	33,63	90
12/11/2025 18:10	21	36,36	99

Desde la izquierda, la primera columna contiene la estampa de tiempo, y las restantes columnas los valores de las variables del PLC. Al disponer de los datos base en Excel será posible crear reportes con gráficas, añadir columnas, etc., de tal manera que se tenga la opción de realizar análisis de datos.

3.6. Resultados

Una vez realizadas las pruebas al sistema se evalúan los resultados obtenidos.

3.6.1. Plantilla

La estructura de la Solución en Visual Studio está organizada de tal manera que la hace reutilizable para cualquier otro proyecto. La figura 45 ilustra la estructura.

Figura 45

Estructura de la plantilla

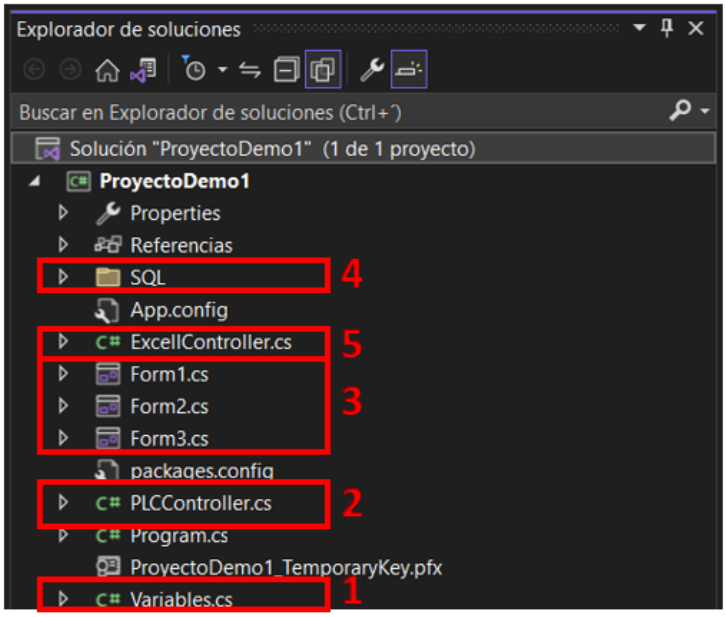
1. Contenedor de variables

2. Comunicación con el PLC, lectura y escritura de variables.

3. Interfaz gráfico y adquisición de variables

4. Enlace con la base de datos

5. Exportación de datos a Excel

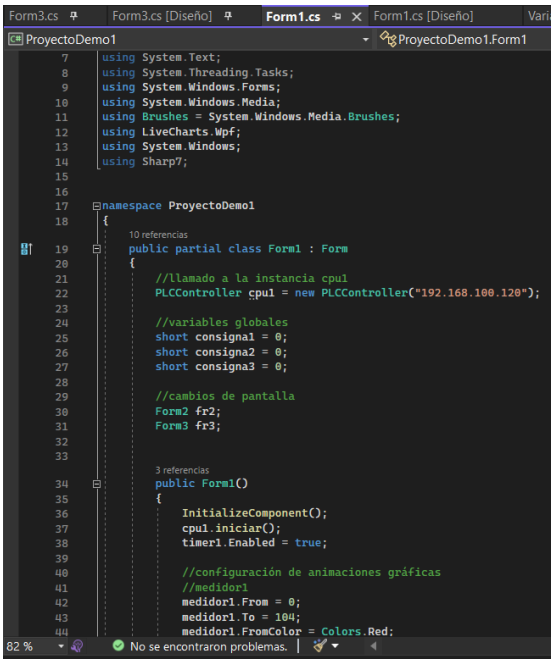


Para cualquier otro proyecto la estructura será la misma, únicamente puede variar la cantidad de Forms ya que corresponden a las pantallas de interface HMI.

Así mismo cada clase se encuentra con comentarios en cada sección del programa lo que facilita la modificación puntal sin necesidad de rehacer todo el código. Varios ejemplos de los comentarios se ven en el programa se ven en la figura 46.

Figura 46

Estructura de comentarios en el programa



En C# los comentarios inician con // y como se ve en la figura 46 cada Clase se encuentra debidamente organizada por secciones mediante las líneas de comentario.

3.6.2. Interfaz gráfico

Respecto a trabajos anteriores similares al presente, la interfaz gráfica es un punto sobresaliente ya que permite una interacción con el usuario amigable mostrando las variables del PLC de una forma clara.

El diseño puede variar e incluso usar normativas como la ISA 101, el objetivo de disponer de una plantilla es lograr flexibilidad en el diseño.

El interfaz que se desarrolla en C# en el artículo “Single Phase Auto-Reclosing Relay of Three-Phase Transmission Line with IoT and SCADA” *(Ahmed Jasim et al., 2025)*, es básico de una sola pantalla; y la gran mayoría de artículos similares mantienen la misma línea de uso básico del interfaz gráfico.

3.6.3. Adquisición de variables del PLC

Mediante las pruebas realizadas, integrando la comunicación con la visualización del HMI, la lectura de las variables se realiza de forma satisfactoria y de igual manera la escritura de las variables.

3.6.4. Integración de más PLCs al HMI

El desarrollo del HMI está enfocado para comunicarse a un solo PLC, sin embargo, por programación puede modificarse para que el HMI pueda comunicarse con más PLCs. Al igual que una arquitectura tradicional de comunicación PLC-HMI es recomendable que un solo PLC sea el encargado de comunicarse con el HMI, y, si se desea más variables provenientes de otro PLC estas deben provenir de una red industrial.

3.7. Discusión

Los resultados obtenidos a partir de las pruebas realizadas al sistema HMI desarrollado en C# evidencian el cumplimiento de los objetivos específicos. La estructura modular de la Solución implementada en Visual Studio muestra reutilización y escalabilidad, ya que permite adaptar el software a nuevos proyectos de automatización industrial con mínimas modificaciones. Esta característica resulta especialmente relevante para la empresa beneficiaria, al reducir los tiempos de desarrollo del software HMI.

En relación con la plantilla, la organización del código fuente en clases y secciones claramente comentadas facilita la comprensión, modificación y ampliación del sistema sin necesidad de rehacer el proyecto completo. Esta práctica mejora la mantenibilidad del software y reduce la dependencia de desarrolladores específicos, lo cual representa una ventaja frente a soluciones cerradas o poco documentadas.

Respecto a la interfaz gráfica, los resultados muestran que el HMI desarrollado ofrece una interacción clara y amigable con el usuario, permitiendo una visualización eficiente de las variables del PLC. Al comparar este desarrollo con trabajos previos, se observa que muchas soluciones similares presentan interfaces gráficas básicas, limitadas a una sola pantalla y con escaso diseño, como se evidencia en el trabajo de Ahmed Jasim et al. (2025). En contraste, el enfoque basado en una plantilla permite adaptar el diseño del HMI a distintas normativas y requerimientos industriales, como la ISA-101, lo que incrementa su aplicabilidad en entornos reales.

En cuanto a la adquisición de variables, las pruebas de lectura y escritura realizadas demuestran que la comunicación entre el HMI y el PLC es confiable y estable, permitiendo el intercambio de datos digitales y analógicos de forma correcta. Esto confirma que el software desarrollado es capaz de integrarse eficazmente en un sistema de control industrial, cumpliendo con los requisitos funcionales de monitoreo y control en tiempo real.

Finalmente, aunque el sistema fue diseñado inicialmente para comunicarse con un único PLC, los resultados indican que su arquitectura permite la expansión hacia la integración de múltiples controladores mediante modificaciones a nivel de programación y arquitectura de red. Esta posibilidad coincide con las buenas prácticas en automatización industrial, donde se recomienda centralizar la comunicación HMI en un PLC principal y obtener información adicional a través de redes industriales. De esta manera, el desarrollo propuesto no solo cumple su función actual, sino que también ofrece potencial de crecimiento y adaptación a sistemas más complejos.

CONCLUSIONES

- El desarrollo del prototipo de software HMI permitió demostrar la viabilidad técnica de implementar una solución propia de supervisión y control industrial utilizando Visual Studio y el lenguaje de programación C#, capaz de comunicarse en tiempo real con Controladores Lógicos Programables Siemens.
- El sistema desarrollado cumple con el objetivo general de constituirse como una alternativa funcional frente a los softwares HMI comerciales de pago, ofreciendo las principales prestaciones requeridas en proyectos de automatización industrial y contribuyendo a la reducción de costos de licenciamiento para la empresa InGel-Pro.
- Se logró desarrollar una plantilla de programación estructurada y reutilizable, preparada para establecer comunicación con PLC Siemens, lo que facilita su adaptación a distintos proyectos de automatización. Esta plantilla permite acelerar el desarrollo de nuevas aplicaciones HMI, estandarizando la estructura del código y simplificando las tareas de configuración y mantenimiento del sistema.
- Las interfaces gráficas implementadas demostraron ser capaces de visualizar y controlar de forma eficiente las principales variables del PLC, ofreciendo al operador un entorno gráfico claro, intuitivo y adecuado para la supervisión de procesos industriales. Asimismo, se verificó la correcta comunicación de variables digitales y analógicas entre el PLC y el HMI, permitiendo tanto la lectura de estados del proceso como la escritura de comandos de control en tiempo real.
- El prototipo desarrollado permitió la obtención y gestión de variables provenientes de múltiples PLC interconectados mediante una red industrial, evidenciando la escalabilidad del sistema y su aplicabilidad en arquitecturas de automatización distribuidas.
- Los resultados obtenidos confirman que el desarrollo de un HMI propio constituye una alternativa técnicamente viable y estratégicamente favorable para la empresa, al reducir la dependencia de software propietario, fortalecer el conocimiento interno y ampliar las posibilidades de implementación de sistemas de automatización industrial en tiempo real.

RECOMENDACIONES

- Se recomienda continuar con el proceso de validación del software HMI desarrollado mediante pruebas en entornos industriales reales, con el fin de evaluar su desempeño, estabilidad y confiabilidad bajo condiciones de operación continua y cargas de trabajo variables. Estas pruebas permitirán identificar oportunidades de mejora antes de su implementación definitiva en proyectos productivos de la empresa InGel-Pro.
- Es conveniente ampliar la compatibilidad del prototipo para soportar diferentes modelos de PLC Siemens y otros protocolos de comunicación industrial basados en Ethernet, lo que incrementaría la flexibilidad del sistema y su aplicabilidad en una mayor variedad de proyectos de automatización industrial.
- Se sugiere implementar mecanismos avanzados de seguridad informática, tales como autenticación de usuarios, gestión de permisos por niveles y protección de las comunicaciones, con el objetivo de garantizar la integridad de los datos y prevenir accesos no autorizados al sistema de control y monitoreo.
- Asimismo, se recomienda incorporar funcionalidades adicionales propias de los sistemas HMI comerciales, como registro histórico de datos, generación de reportes, alarmas configurables y tendencias gráficas, lo que permitiría mejorar la capacidad de análisis del proceso y la toma de decisiones operativas.
- Se aconseja documentar y estandarizar la plantilla de desarrollo creada, acompañándola de manuales técnicos y guías de uso, para facilitar su mantenimiento, actualización y transferencia de conocimiento dentro de la empresa, asegurando la sostenibilidad y evolución del software HMI en futuros proyectos.

REFERENCIAS

Ahmed Jasim, E., Allu, A. A., & Suliman, M. Y. (2025). Single phase auto-reclosing relay of three-phase transmission line with IoT and SCADA. *NTU Journal of Engineering and Technology*. https://journals.ntu.edu.iq/index.php/NTU-JET/article/view/486/575

Amazon Web Services, Inc. (2025). ¿Qué es .NET? https://aws.amazon.com/es/what-is/net/

Bacis, A. (2024). *Exploring the role of industrial edge computing in simulated environments using DriveSim Engineer and PLCSim Advanced* [Tesis de maestría, Politecnico di Milano]. https://www.politesi.polimi.it/retrieve/a28d620f-2b26-4ea6-81a4-36f767d479b8/_Tesi_2025_03_AnnaBacis.pdf

Certus. (2020). ¿Qué es Visual Studio .NET y cuáles son sus beneficios? https://www.certus.edu.pe/blog/que-es-visual-studio-net/

Corral González, M. (2021). *Diseño y desarrollo de un sistema HMI para una aplicación de Industria 4.0* [Trabajo de Fin de Grado, Universidad de Valladolid]. UVaDoc. https://uvadoc.uva.es/bitstream/handle/10324/50032/TFG-G5213.pdf?sequence=1&isAllowed=y

Dempsey, L., & Deeley, K. (2023). Developing MATLAB apps using the model-view-controller pattern. MathWorks. https://la.mathworks.com/company/technical-articles/developing-matlab-apps-using-the-model-view-controller-pattern.html

Digi License. (2022, julio 11). Visual Studio 2022: comparación de ediciones y versiones. https://www.digilicense.com/es/blogs/blog/visual-studio-2022-comparacion-de-ediciones-y-vers/

Dikme, U. (2021). Industrial user interface software design for visual Python AI applications using embedded Linux-based systems. *Journal of Applied and Physical Sciences, 7*(1), 1–10.

Euroinnova. (2024). Tipos de HMI: descubre la interacción hombre-máquina. https://www.euroinnova.com/nuevas-tecnologias/articulos/tipos-hmi

García Jiménez, M. (2025). Buses de comunicación en entornos industriales: tipos, topologías y recomendaciones de instalación. https://www.intarcon.com/buses-de-comunicacion-en-entornos-industriales-tipos-topologias-y-recomendaciones-de-instalacion/

Ingel-Pro. (2023). Redes de comunicación industrial: panorama general. https://ingel-pro.com/blog1.html

Jeng, S.-L. (2021). Web-based human-machine interfaces of industrial controllers in single-page applications. *Journal of Sensors, 2021*, Article 6668843. https://doi.org/10.1155/2021/6668843

Joskowicz, J. (2008). Reglas y prácticas en extreme programming (XP) [Documento de curso, Universidad de Vigo]. https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf

Linares Gutiérrez, V. M. (2025). *Implementación de redes seguras en entornos críticos: instrumentación, buses de campo y medios de transmisión* [Trabajo de fin de grado, Universidad de Cantabria]. https://repositorio.unican.es/xmlui/bitstream/handle/10902/36854/456313.pdf?sequence=1&isAllowed=y

Lista de protocolos de comunicación en serie: una guía completa. (2025, 20 de septiembre). Come-Star. https://www.come-star.com/es/blog/serial-communication-protocols/

Liyanage, U. B. (2021). *Final project* [Trabajo académico no publicado, University of Sunderland]. https://d1wqtxts1xzle7.cloudfront.net/51695525/159132966_Udari_Bhagya_Liyanage-libre.pdf

Llamas, L. (2026). Qué son y cómo usar clases en C#. https://www.luisllamas.es/csharp-clases/

Llamas, L. (2026). Qué son y cómo usar threads en C#. https://www.luisllamas.es/csharp-threads/

Mhetraskar, S. S., Namekar, S. A., Holmukhe, R. M., & Tamke, S. M. (2020). Industrial automation using PLC, HMI and its protocols based on real-time data for analysis. *International Journal of Advanced Research in Engineering and Technology (IJARET), 11*(10), 1353–1363.

Microsoft. (2020, junio 12). ¿Qué es Visual Studio? https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022

Peciña, L. (2018). *Programación de controles avanzados SIMATIC S7-1500 con TIA Portal AWL y SCL*. Marcombo.

¿Qué es Entity Framework? (2025). EntityFramework Tutorial. https://www.entityframeworktutorial.net/entityframework6/what-is-entityframework.aspx

Siemens. (2023). El núcleo de la automatización eficaz: la ingeniería integrada. https://www.siemens.com/mx/es/productos/automatizacion/industry-software/automation-software/tia-portal/integrated-engineering.html

Siemens AG. (2022). *SIMATIC S7-1500 CPU 1512C-1 PN (6ES7512-1CK01-0AB0): manual del producto*. https://cache.industry.siemens.com/dl/files/676/109478676/att_898615/v1/s71500_cpu1512c_1_pn_manual_es-ES_es-ES.pdf

Tokio School. (2024). C#, ¿qué es? ¿Quién lo inventó? ¡Conoce este lenguaje de programación! https://www.tokioschool.com/noticias/c-que-es/

Tükez, E. T., & Kaya, A. (2022). Scada system for next-generation smart factory environments. *Icontech International Journal, 6*(2), 1–12. http://icontechjournal.com/index.php/iij/article/view/211/64

Witautomatización. (2023). ¿Qué es PROFINET y para qué sirve? https://witautomatizacion.es/que-es-profinet-y-para-que-sirve/