

**Universidad Tecnológica de La Habana**

**“José Antonio Echeverría”**

**CUJAE**

**Facultad de Ingeniería Informática**

**Módulo de detección de anomalías en un sistema de control industrial**

**Informe de las prácticas profesionales II**

**Autor: Fabio Ford Campbell (fford@ceis.cuaje.edu.cu)**

**Tutor: Dra.C Nayma Cepero Pérez (no me sé el correo)**

**La Habana,**

**enero de 2025**

**Resumen**

En el contexto de la Industria 4.0, la detección temprana de anomalías en sistemas industriales es crucial para garantizar la continuidad y eficiencia operativa. Este estudio se enfoca en el desarrollo de un módulo de detección de anomalías que supervisa en tiempo real los datos operativos y de red de un sistema industrial experimental. Utilizando algoritmos de minería de datos y aprendizaje automático, el sistema es capaz de identificar desviaciones significativas que podrían indicar fallos operacionales. El objetivo principal es desarrollar un módulo capaz de detectar y alertar sobre anomalías en tiempo real, mejorando la resiliencia y eficiencia de los procesos industriales.

**Palabras Claves:** detección de anomalías, sistemas industriales, minería de datos, aprendizaje automático, Industria 4.0

**Abstract**

In the context of Industry 4.0, early detection of anomalies in industrial systems is crucial to ensuring operational continuity and efficiency. This study focuses on developing an anomaly detection module that monitors operational and network data from an experimental industrial system in real-time. Using data mining and machine learning algorithms, the system can identify significant deviations that may indicate operational failures. The main goal is to develop a module capable of detecting and alerting anomalies in real-time, thereby improving the resilience and efficiency of industrial processes.

**Keywords:** anomaly detection, industrial systems, data mining, machine learning, Industry 4.0

# 

**Índice de contenido**

[Introducción 1](#_Toc29071)

[Capítulo 1: Fundamentos Teóricos 4](#_Toc17556)

[1.1 Detección de Anomalías en Sistemas Industriales 4](#_Toc8011)

[1.1.1 Tipos de Anomalías en Sistemas Industriales 4](#_Toc5889)

[1.1.2 Principios de la Detección de Anomalías 5](#_Toc20160)

[1.1.3 Algoritmos para la Detección de Anomalías 5](#_Toc26516)

[1.1.4 Sistemas Industriales y su Monitoreo 6](#_Toc18129)

[1.1.5 Minería de Datos y Aprendizaje Automático en la Detección de Anomalías 6](#_Toc11803)

[1.1.6 Alertas en Tiempo Real 7](#_Toc8879)

[1.1.7 Desafíos en la Detección de Anomalías 7](#_Toc22863)

[1.1.8 Impacto de la Detección de Anomalías en la Industria 4.0 8](#_Toc32073)

[1.2 Fundamentación de la Tecnología a Usar 8](#_Toc27388)

[1.3 Conclusiones Parciales 11](#_Toc16754)

[Capítulo 2: Solución a la Problemática 13](#_Toc6674)

[2.1 Entorno de ICS simulado 13](#_Toc3096)

[2.1.1 Arquitectura del Sistema 14](#_Toc21125)

[2.1.2 Componentes Principales 14](#_Toc14646)

[Referencias Bibliográficas 18](#_Toc26855)

**Índice de figura**

*[Figura](#_Toc29256)* [1](#_Toc29256)  *[Arquitectura del entorno ICS simulado](#_Toc29256)* [14](#_Toc29256)

# Introducción

La Industria 4.0 ha transformado el panorama de los sistemas industriales con la integración de tecnologías conectadas y sistemas ciberfísicos. Estos cambios han traído consigo la necesidad de implementar mecanismos de monitoreo avanzados para asegurar el correcto funcionamiento de los procesos industriales. En este contexto, garantizar la fiabilidad de los datos que generan y procesan estos sistemas es esencial para optimizar las operaciones y evitar interrupciones inesperadas. Para ello, es fundamental contar con sistemas capaces de identificar y responder a comportamientos anómalos que puedan comprometer la estabilidad del entorno industrial [1].

Un enfoque ampliamente utilizado para la protección de la continuidad operativa en sistemas industriales es la implementación de módulos de detección de anomalías. Estos módulos supervisan en tiempo real los datos generados por sensores y controladores, buscando desviaciones significativas que puedan ser indicadores de fallos operativos. Al detectar estas anomalías de manera temprana, es posible minimizar el impacto en los sistemas, garantizando su disponibilidad, integridad y eficiencia operativa [1-3].

Uno de los mayores desafíos en la detección de anomalías en sistemas industriales es la capacidad de procesar grandes volúmenes de datos provenientes de diversas fuentes en tiempo real. A esto se suma la necesidad de que los algoritmos de detección sean lo suficientemente precisos como para diferenciar entre fluctuaciones normales del sistema y comportamientos que podrían derivar en fallos o problemas operativos, evitando tanto falsos positivos como falsos negativos que puedan generar costos innecesarios o fallos no detectados [4].

El problema de investigación en este campo surge ante la creciente complejidad de los sistemas industriales y la necesidad de contar con mecanismos automatizados que puedan adaptarse a las condiciones cambiantes del entorno operativo [3, 5]. ¿Cómo diseñar un módulo de detección de anomalías que se adapte dinámicamente a las características de los datos y sea capaz de identificar anomalías en un entorno industrial simulado?

El objeto de estudio de esta investigación es un sistema industrial experimental que incluye sensores, microcontroladores STM32 y un PLC flexible basado en la plataforma OpenPLC. Este sistema simula un entorno industrial con capacidad para capturar y procesar datos operativos, ofreciendo un marco adecuado para implementar y probar el módulo de detección de anomalías [1].

Si bien existen tecnologías para el monitoreo de procesos industriales, la complejidad de los entornos y la cantidad de datos que deben analizarse en tiempo real siguen siendo un reto. Los fallos operativos pueden comprometer seriamente la producción, generando pérdidas económicas y disminuyendo la eficiencia de los sistemas. Por lo tanto, contar con un sistema que permita la detección temprana de fallos es crucial para evitar estos contratiempos [1, 4].

El uso de módulos de detección de anomalías en este contexto es fundamental, ya que permiten identificar patrones inusuales de comportamiento en los sistemas antes de que el daño sea irreversible. El campo de acción de esta investigación se centra en la implementación de algoritmos de detección de anomalías que puedan supervisar tanto las señales operativas de los sensores como el comportamiento general del sistema industrial experimental [4, 5].

El objetivo general de esta investigación es desarrollar un módulo de detección de anomalías que permita identificar comportamientos anómalos en tiempo real en un sistema industrial simulado, utilizando técnicas de minería de datos y aprendizaje automático aplicadas a los datos operativos recolectados por el sistema [2].

Para alcanzar este objetivo, se realizará la implementación de algoritmos de detección que se integrarán con los controladores STM32 y el PLC del sistema industrial. Además, se llevarán a cabo pruebas exhaustivas para validar la efectividad del módulo en la detección de anomalías, garantizando su capacidad de respuesta ante posibles fallos operacionales [2, 3].

El alcance de esta investigación incluye la detección de anomalías en señales operativas para que los operadores puedan tomar decisiones informadas sobre el mantenimiento y ajuste del sistema. Este enfoque proporcionará un marco claro para el desarrollo de la investigación y asegurará que se aborden los aspectos más críticos de la continuidad operativa en sistemas industriales [3, 4].

# Capítulo 1: Fundamentos Teóricos

Este capítulo establece los fundamentos teóricos necesarios para comprender y alcanzar los objetivos propuestos en este trabajo. Se discuten conceptos clave relacionados con la detección de anomalías, los sistemas industriales, los algoritmos de minería de datos y aprendizaje automático, y las tecnologías asociadas.

## 1.1 Detección de Anomalías en Sistemas Industriales

La detección de anomalías es una técnica fundamental utilizada para identificar eventos inusuales o patrones de comportamiento que se desvían de la operación normal de un sistema. En un sistema industrial, la identificación temprana de anomalías es crucial para prevenir fallos operacionales o deterioros en los equipos, lo que podría causar tiempos de inactividad no planificados o interrupciones en la producción. La detección de anomalías permite que los operadores puedan intervenir de manera preventiva y corregir posibles problemas antes de que estos escalen [6].

### 1.1.1 Tipos de Anomalías en Sistemas Industriales

En los sistemas industriales, las anomalías se pueden clasificar en tres tipos principales:

Anomalías puntuales: son eventos aislados que se desvían del comportamiento esperado. Por ejemplo, un sensor de temperatura que registra un valor anormal durante un breve período.

Anomalías contextuales: ocurren cuando un evento es anormal en un contexto específico. Por ejemplo, una temperatura elevada podría ser normal en un proceso, pero anómala en otro contexto operativo.

Anomalías colectivas: se refieren a un grupo de puntos de datos que, colectivamente, representan un comportamiento anómalo. Esto podría ser, por ejemplo, una tendencia inusual en los niveles de vibración de una máquina durante un período de tiempo.

### 1.1.2 Principios de la Detección de Anomalías

La detección de anomalías en sistemas industriales se basa en los siguientes principios fundamentales:

Monitorización en tiempo real: los datos operativos de sensores y controladores se analizan constantemente para detectar comportamientos anómalos.

Modelado del comportamiento normal: el sistema aprende a reconocer lo que se considera una operación normal del sistema industrial a través de la recopilación y análisis de datos históricos.

Respuesta a tiempo: una vez detectada una anomalía, el sistema genera alertas en tiempo real que permiten a los operadores tomar medidas correctivas antes de que ocurran daños graves.

### 1.1.3 Algoritmos para la Detección de Anomalías

Existen diferentes tipos de algoritmos utilizados para la detección de anomalías en sistemas industriales, entre ellos:

Métodos basados en reglas: se configuran reglas específicas que definen los umbrales operativos normales de un sistema. Cuando un dato excede estos umbrales, se detecta una anomalía.

Métodos estadísticos: se basan en modelos probabilísticos que evalúan la probabilidad de que un dato sea anómalo con respecto al comportamiento histórico del sistema.

Algoritmos de minería de datos y aprendizaje automático: estos métodos permiten el análisis de grandes volúmenes de datos operativos. Los modelos de aprendizaje automático pueden entrenarse para identificar patrones complejos que indican anomalías. Algoritmos como el k-means clustering, SVM (Support Vector Machines) y redes neuronales se utilizan comúnmente para este fin [7] .

### 1.1.4 Sistemas Industriales y su Monitoreo

Los sistemas industriales son entornos altamente interconectados donde diferentes dispositivos, como sensores, controladores lógicos programables (PLC), y sistemas SCADA (Supervisory Control and Data Acquisition), supervisan y controlan procesos físicos. La cantidad de datos que estos sistemas generan es considerable, y la detección de anomalías en estos sistemas requiere el procesamiento continuo de datos en tiempo real.

Sensores y controladores: monitorean variables clave como temperatura, presión, velocidad de motores, entre otros. Estos datos son fundamentales para la toma de decisiones en tiempo real.

PLC y SCADA: permiten la automatización de procesos y la recopilación de datos operativos. Estos datos proporcionan la base para aplicar algoritmos de detección de anomalías y generar alertas en caso de que ocurra un comportamiento fuera de lo esperado.

### 1.1.5 Minería de Datos y Aprendizaje Automático en la Detección de Anomalías

El uso de técnicas de minería de datos y aprendizaje automático ha revolucionado la manera en que se detectan las anomalías en sistemas industriales. Estas técnicas permiten analizar grandes cantidades de datos y detectar patrones sutiles que podrían no ser evidentes mediante métodos convencionales.

Minería de datos: implica la extracción de información útil a partir de grandes volúmenes de datos. En el contexto de los sistemas industriales, se aplica para identificar tendencias y patrones anómalos.

Aprendizaje automático: a través de técnicas supervisadas y no supervisadas, los algoritmos de aprendizaje automático pueden aprender a diferenciar entre el comportamiento normal y anómalo. Modelos como las redes neuronales profundas y los modelos de regresión son ampliamente utilizados en estos contextos [8, 9].

### 1.1.6 Desafíos en la Detección de Anomalías

Algunos de los principales desafíos en la detección de anomalías en sistemas industriales incluyen:

Falsos positivos y negativos: uno de los mayores retos es garantizar que el sistema sea lo suficientemente preciso para minimizar las falsas alarmas (falsos positivos) y, al mismo tiempo, no pase por alto comportamientos anómalos reales (falsos negativos).

Procesamiento en tiempo real: debido a la gran cantidad de datos generados, los sistemas deben ser capaces de procesar la información en tiempo real sin comprometer el rendimiento de la operación industrial.

### 1.1.7 Impacto de la Detección de Anomalías en la Industria 4.0

La implementación de módulos de detección de anomalías contribuye significativamente a la Industria 4.0, proporcionando un nivel adicional de seguridad y optimización operativa. Estos módulos permiten que los sistemas industriales se vuelvan más resilientes, mejorando tanto la eficiencia como la seguridad operativa, y ayudando a prevenir interrupciones en la producción.

## 1.2 Fundamentación de la Tecnología a Usar

En el desarrollo de soluciones para la detección de anomalías en sistemas industriales, la elección de las tecnologías adecuadas es un paso crucial que puede influir significativamente en la eficacia y eficiencia de la solución final. En este contexto, hemos seleccionado varias tecnologías clave para emplear en el desarrollo de nuestra solución: el lenguaje de programación Python y sus bibliotecas asociadas, SQLite3 para la base de datos, PyCharm como entorno de desarrollo integrado (IDE), y herramientas específicas para el manejo de algoritmos y pruebas como scikit-learn, SciPy, y TensorFlow/Keras.

A continuación, presentamos una justificación detallada para la selección de cada una de estas tecnologías, destacando cómo su uso contribuye a la eficacia de nuestra solución y cómo se alinean con los objetivos de nuestra investigación. Nuestra meta es proporcionar una base sólida para el desarrollo de una solución eficiente y eficaz que permita la identificación de comportamientos anómalos en sistemas industriales.

1. Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general que ha ganado popularidad en el ámbito de la ciencia de datos y la inteligencia artificial. Su sintaxis sencilla y clara permite un desarrollo rápido y eficiente. Python cuenta con una extensa comunidad de desarrolladores y una rica colección de bibliotecas que facilitan la implementación de algoritmos de detección de anomalías y procesamiento de datos. La flexibilidad de Python lo convierte en una excelente elección para prototipos y desarrollo ágil, permitiendo una rápida iteración y prueba de ideas.

2. Pandas y NumPy

- Pandas: Es una biblioteca fundamental para la manipulación y análisis de datos en Python. Proporciona estructuras de datos flexibles como DataFrames, que facilitan la carga, manipulación y análisis de grandes volúmenes de datos operativos generados por sistemas industriales. Con Pandas, es posible realizar operaciones complejas de filtrado y transformación de datos de manera eficiente, lo que es crucial para preparar los datos antes de aplicar algoritmos de detección de anomalías.

- NumPy: Esta biblioteca proporciona soporte para arreglos multidimensionales y operaciones matemáticas avanzadas, lo que es especialmente útil para cálculos numéricos y análisis estadístico. Al trabajar con datos de sensores en tiempo real, NumPy permite realizar cálculos eficientes que son esenciales para implementar algoritmos de aprendizaje automático en la detección de anomalías.

3. Scikit-learn y SciPy

- Scikit-learn (sklearn): Es una biblioteca clave para implementar algoritmos de aprendizaje automático. Proporciona herramientas robustas para técnicas como Isolation Forest, Support Vector Machines (SVM) y detección de outliers, que son esenciales en el módulo de detección de anomalías. Su facilidad de uso y documentación extensa la convierten en una opción ideal para experimentación y desarrollo.

- SciPy: Es utilizada para cálculos científicos y estadísticos avanzados, incluyendo la optimización, integración y análisis de datos. Es particularmente útil en las pruebas estadísticas necesarias para validar la detección de anomalías.

4. TensorFlow y Keras

Estas bibliotecas se emplean para la implementación de modelos de redes neuronales avanzados, como autoencoders, utilizados para detectar patrones anómalos en datos complejos o de alta dimensionalidad.

- Keras: Ofrece una API intuitiva y de alto nivel para construir y entrenar modelos.

- TensorFlow: Proporciona la potencia computacional subyacente para entrenar modelos en CPU o GPU, lo cual es esencial para manejar grandes volúmenes de datos operativos en sistemas industriales.

5. SQLite3

SQLite es un sistema de gestión de bases de datos relacional ligero y de código abierto. Su integración con Python a través del módulo sqlite3 facilita el almacenamiento y gestión de datos sin la necesidad de configurar un servidor de base de datos. Esto es ventajoso para proyectos de menor escala o durante las etapas de desarrollo y prueba, ya que permite un acceso rápido y eficiente a los datos sin complicaciones adicionales. SQLite es ideal para almacenar los registros de datos operativos y las alertas generadas por el sistema de detección de anomalías.

6. PyCharm

PyCharm es un entorno de desarrollo integrado (IDE) muy popular para Python, conocido por su robustez y su amplia gama de características que facilitan el desarrollo de software. Ofrece herramientas como autocompletado de código, depuración avanzada y control de versiones, lo que mejora la productividad del desarrollador y reduce la posibilidad de errores. Además, PyCharm proporciona integración con sistemas de control de versiones y soporte para bibliotecas de ciencia de datos, lo que lo convierte en una opción ideal para el desarrollo de soluciones complejas en detección de anomalías.

7. Otras Herramientas: time

La biblioteca estándar `time` se utilizó para realizar mediciones precisas de rendimiento en las pruebas de detección de anomalías. Esto permitió evaluar la eficiencia de los algoritmos en términos de tiempo de ejecución, un factor crítico en sistemas industriales donde la detección en tiempo real es fundamental.

Con esta selección tecnológica, la solución está equipada para manejar datos operativos de alta complejidad, implementar técnicas avanzadas de detección y proporcionar resultados en tiempo real.

## 1.3 Conclusiones Parciales

A lo largo de este capítulo, se ha establecido una base sólida para el desarrollo de un módulo de detección de anomalías en sistemas industriales, destacando su importancia como mecanismo crucial para garantizar la seguridad y continuidad operativa en entornos conectados. Este enfoque permite identificar de manera temprana comportamientos inusuales, contribuyendo a prevenir fallos operativos, minimizar tiempos de inactividad y mitigar riesgos cibernéticos.

Se ha definido un marco conceptual que incluye los tipos de anomalías y los principios fundamentales que rigen su detección en sistemas industriales. Además, se han identificado métodos efectivos de análisis, especialmente aquellos basados en aprendizaje automático, redes neuronales y minería de datos, como herramientas clave para reducir falsos positivos y negativos en la detección de comportamientos anómalos.

La elección de Python como lenguaje de programación, junto con bibliotecas como Pandas, NumPy, Scikit-learn y SciPy, ha proporcionado una base sólida para el manejo, análisis y procesamiento de datos operativos. Estas herramientas facilitan no solo la manipulación eficiente de grandes volúmenes de datos, sino también la implementación de algoritmos avanzados de detección de anomalías.

Asimismo, la integración de TensorFlow y Keras ha permitido diseñar modelos de aprendizaje profundo, para analizar datos complejos y detectar patrones inusuales en tiempo real. Estas tecnologías, combinadas con herramientas como SQLite3 para el almacenamiento eficiente de datos y PyCharm como entorno de desarrollo, han creado un ecosistema tecnológico propicio para un desarrollo ágil y efectivo. La biblioteca `time`, utilizada para medir el rendimiento de los algoritmos, asegura que las soluciones desarrolladas sean prácticas y escalables en entornos industriales.

En conclusión, este capítulo ha logrado definir un marco teórico y práctico robusto para el desarrollo del módulo de detección de anomalías, alineándose con los objetivos de la investigación. La combinación de tecnologías seleccionadas asegura que la solución propuesta sea eficiente, innovadora y capaz de mejorar la seguridad y la eficiencia operativa en sistemas industriales.

# Capítulo 2: Solución a la Problemática

En este capítulo se presenta la solución propuesta para abordar la problemática de garantizar la seguridad y fiabilidad en sistemas industriales frente a posibles fallos operativos y amenazas cibernéticas. La solución se centra en el desarrollo e implementación de un módulo de detección de anomalías diseñado para identificar patrones inusuales en los datos operativos y de red del sistema.

El módulo de detección de anomalías forma parte de un sistema experimental que simula un entorno de control industrial (ICS) accesible y modular. Este banco de pruebas permite analizar el comportamiento del sistema bajo condiciones normales y ante escenarios de ataque, facilitando la identificación temprana de anomalías que podrían comprometer su estabilidad y seguridad.

A lo largo del capítulo, se describen los componentes principales de la solución, la arquitectura del entorno de ICS simulado y las funcionalidades específicas del módulo de detección de anomalías. Además, se detalla cómo este módulo contribuye a mitigar riesgos operativos y cibernéticos mediante la integración de técnicas avanzadas de análisis de datos.

## 2.1 Entorno de ICS simulado

En esta sección se describe la arquitectura, diseño e implementación del sistema experimental desarrollado para simular y analizar escenarios en un entorno de control industrial. Este banco de pruebas, diseñado para ser accesible y de bajo costo, permite estudiar y validar estrategias de seguridad en sistemas industriales, replicando de forma precisa el funcionamiento de un Sistema de Control Industrial (ICS).

El sistema integra componentes hardware y software, incluyendo sensores, controladores, interfaces hombre-máquina (HMI) y sistemas SCADA, para recrear un entorno funcional que puede ser utilizado en simulaciones, investigaciones académicas y formación técnica en ciberseguridad industrial.

2.1.1 Arquitectura del Sistema

La arquitectura del sistema está diseñada para reflejar un ICS real, manteniendo una estructura modular que facilita el análisis y la experimentación. La Figura 1 Arquitectura del entorno ICS simulado ilustra la arquitectura general del banco de pruebas, destacando las interacciones entre sus componentes principales.

Изображение выглядит как диаграмма, снимок экрана, линия, круг

Автоматически созданное описание

*Figura 1* *Arquitectura del entorno ICS simulado*

### 2.1.2 Componentes Principales

1. Sensores

- Sensores de posición de carga: Basados en fotoresistores, ubicados al inicio y al final de la cinta transportadora.

- Sensores de posición del manipulador: Utilizan interruptores de límite para registrar las posiciones mínima y máxima.

2. Microcontroladores (STM32)

- Actúan como dispositivos de entrada/salida remota que recopilan los datos de los sensores y los envían al Controlador Lógico Programable (PLC).

3. Controlador Lógico Programable (PLC)

- Se utiliza un Soft-PLC implementado en una plataforma Raspberry Pi, ejecutando OpenPLC para la lógica de control. El PLC gestiona las entradas y salidas digitales y analógicas, permitiendo el control en tiempo real del sistema.

4. Interfaz Hombre-Máquina (HMI)

- Implementada en Raspberry Pi, esta interfaz permite visualizar datos en tiempo real y optimizar procesos mediante un panel interactivo que muestra el estado de los sensores, parámetros del sistema y tendencias históricas.

5. Sistema de Supervisión y Adquisición de Datos (SCADA)

- Utilizado para monitorear el sistema de manera centralizada, gestionar alarmas y responder a eventos en tiempo real. El sistema SCADA está basado en software alojado en Raspberry Pi.

6. Servidor y Análisis de Tráfico

- Un Orange Pi actúa como servidor central, recopilando datos de red y del sistema a través de Docker. Los datos son gestionados con tecnologías como Node.js, NestJS, PostgreSQL y Vue3, asegurando escalabilidad y flexibilidad.

### 2.1.3 Recopilación de Datos

El sistema recopila datos operativos y de red para analizar el comportamiento del entorno. Los datos se clasifican en:

- Datos del sistema: Servicios activos, interfaces de red, recursos disponibles (CPU, memoria, almacenamiento).

- Datos de red: Tablas ARP, tráfico por protocolo (ARP, ICMP, TCP, UDP, HTTP, ModBus).

- Datos de aplicaciones: Información sobre procesos en ejecución, puertos abiertos y servicios asociados.

La recopilación se realiza mediante scripts en Python, utilizando bibliotecas como scapy y psutil, y los datos son almacenados en una base de datos relacional PostgreSQL.

### 2.1.4 Implementación Técnica

1 Plataforma Hardware

La elección de componentes accesibles y económicos, como Raspberry Pi y Orange Pi, permite replicar funcionalidades avanzadas de un ICS real manteniendo costos bajos. Cada componente ha sido configurado para maximizar su interoperabilidad y rendimiento.

2 Plataforma Software

El software instalado incluye:

- Soft-PLC: OpenPLC para la lógica de control.

- SCADA y HMI: Herramientas gráficas para la supervisión en tiempo real.

- Monitoreo de red y sistema: Scripts personalizados en Python para la captura y procesamiento de datos.

- Servidor: Infraestructura basada en contenedores Docker para garantizar modularidad y escalabilidad.

### 2.1.5 Escenarios de Pruebas y Seguridad

El sistema soporta múltiples escenarios de prueba diseñados para evaluar la resiliencia de los ICS frente a ataques comunes. Algunos de los escenarios incluyen:

1. Manipulación de sensores: Interferencia en los datos recolectados por los sensores para evaluar la respuesta del sistema.

2. Ataques de red: Simulación de ataques como interceptación de paquetes o manipulación de ARP para evaluar la robustez de la red.

3. Intrusión en el PLC: Pruebas de seguridad en la lógica del Soft-PLC, buscando vulnerabilidades en su programación o hardware subyacente.

### 2.1.6 Base de Datos

La arquitectura de la base de datos está optimizada para gestionar grandes volúmenes de datos en tiempo real, incluyendo información del sistema, red y tráfico. La Figura 2 presenta el diseño lógico de la base de datos, con las siguientes tablas principales:

- Aplicaciones instaladas: Información sobre nombre, versión y descripción.

- ARP y tráfico de red: Direcciones IP y físicas, protocolos utilizados y estadísticas de paquetes.

- Servicios del sistema: Estado y configuración de los servicios ejecutándose en los dispositivos.

### 2.1.7 Conclusión

El diseño modular y accesible de este banco de pruebas proporciona un entorno ideal para evaluar y perfeccionar módulos de detección de anomalías en sistemas industriales. Su capacidad para simular con precisión procesos y escenarios industriales reales permite a los usuarios realizar pruebas exhaustivas en un contexto controlado, replicando tanto condiciones normales como situaciones anómalas.

La implementación de tecnologías abiertas y la integración de hardware de bajo costo aseguran que el sistema sea funcional y asequible, lo que facilita su adopción en entornos académicos e industriales. Este enfoque no solo fomenta la formación en ciberseguridad, sino que también garantiza que las pruebas de los módulos de detección de anomalías sean eficientes y representativas de los desafíos del mundo real, contribuyendo así al desarrollo de soluciones más robustas y confiables.

## 2.2 Ingeniería de Requisitos

Esta sección proporciona una visión integral de los requisitos necesarios para el desarrollo del sistema, abordando tanto los aspectos funcionales como no funcionales, el flujo de datos, el modelo de dominio, y se incluye un glosario de términos clave. Además, se presentan los requisitos funcionales que describen las acciones específicas que el software debe realizar y los requisitos no funcionales que destacan las cualidades deseadas del sistema, así como las restricciones y problemas frecuentes que deben considerarse durante el proceso de desarrollo.

### 2.2.1 Breve descripción de la propuesta

La propuesta se centra en desarrollar soluciones avanzadas de detección de anomalías en sistemas industriales, diseñadas para identificar y mitigar comportamientos anómalos en tiempo real. Esto garantiza la seguridad operativa, la continuidad de los procesos y la optimización del rendimiento en entornos industriales críticos.

### 2.2.2 Diagrama de actividades

En la Figura 1 se muestra el flujo que sigue el módulo de detección de anomalías para analizar y clasificar los datos operativos del sistema en normales o anómalos.

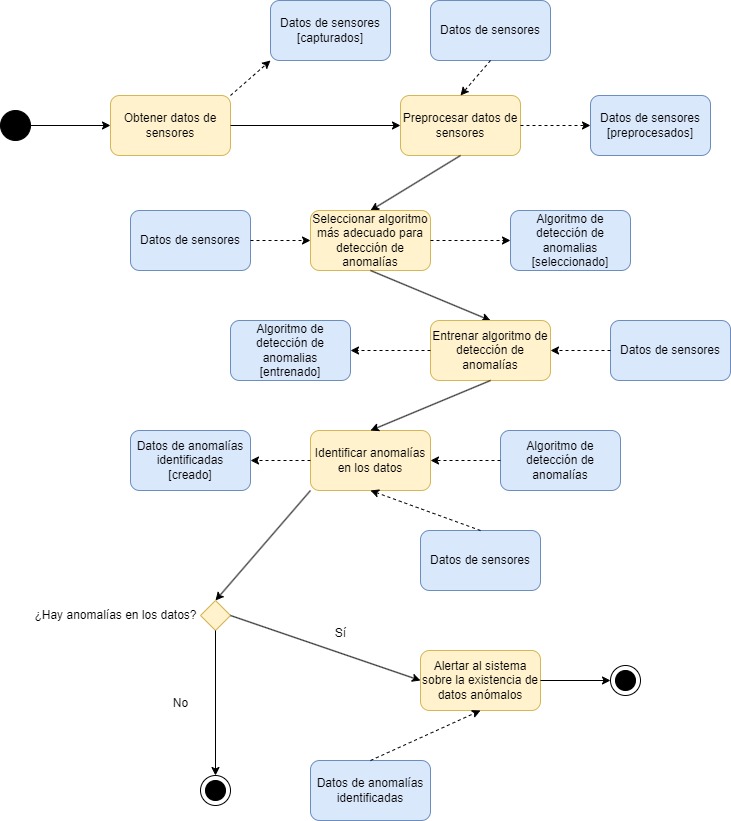


Figura 1 Diagrama de Actividades

En primera instancia, los datos generados por los sensores y mecanismos del entorno industrial son capturados y procesados de manera eficiente. Posteriormente, estos datos son analizados para determinar el algoritmo de detección de anomalías más adecuado, priorizando criterios de precisión y velocidad. Una vez seleccionado el algoritmo óptimo, este es entrenado con datos históricos y procesados para garantizar su desempeño en el contexto específico del sistema.

Tras el entrenamiento, el algoritmo clasifica los datos operativos en dos categorías: normales y anómalos. En caso de detectar anomalías, el sistema genera y envía una alerta inmediata a los componentes correspondientes, permitiendo una acción proactiva. Por el contrario, si no se identifican irregularidades, el sistema continúa operando sin emitir notificaciones, optimizando recursos y asegurando un flujo operativo continuo.

### 2.2.2 Modelo de Dominio

### 2.2.3 Glosario de Términos

Inteligencia Artificial: Campo de estudio y desarrollo tecnológico que busca crear sistemas o máquinas capaces de realizar tareas que requieren inteligencia humana, como el análisis de datos, toma de decisiones y detección de patrones complejos.

Minería de Datos: Proceso de extracción de información valiosa, patrones ocultos y conocimientos significativos a partir de grandes conjuntos de datos generados en sistemas industriales.

Aprendizaje Automático: Rama de la inteligencia artificial enfocada en desarrollar algoritmos y modelos que permiten a las máquinas aprender y mejorar automáticamente a partir de los datos sin ser programadas explícitamente.

Aprendizaje Supervisado: Técnica de aprendizaje automático en la que se entrena un modelo utilizando datos etiquetados, donde se proporcionan tanto las características de entrada como las etiquetas correspondientes. En el contexto industrial, se utiliza para clasificar datos operativos como normales o anómalos.

Meta-aprendizaje: Rama del aprendizaje automático que se enfoca en crear algoritmos que mejoran su capacidad de aprender al aprender de forma más eficiente y efectiva de nuevos conjuntos de datos.

Meta-componente: Componente dentro del módulo de detección de anomalías que tiene como objetivo mejorar la precisión y rapidez del sistema mediante la selección dinámica del algoritmo más adecuado según el contexto industrial.

Base de Hechos: Repositorio que almacena información histórica sobre el comportamiento normal y anómalo del sistema, utilizado para entrenar y evaluar los algoritmos de detección de anomalías.

Sensores Industriales: Dispositivos que recopilan datos sobre las condiciones operativas de un entorno industrial, como temperatura, presión, vibraciones o flujo, los cuales son fundamentales para la detección de anomalías.

Mecanismos Industriales: Componentes físicos y operativos de una planta industrial que generan datos de funcionamiento y cuya monitorización es esencial para identificar fallos o anomalías.

Isolation Forest: Algoritmo basado en árboles de decisión que se utiliza para la detección de anomalías al aislar los puntos de datos que son significativamente diferentes del resto.

Z-Score: Técnica estadística utilizada para identificar anomalías midiendo la desviación estándar de un punto de datos con respecto al promedio.

SQLite3: Sistema de gestión de bases de datos ligero e incrustado, utilizado para almacenar y gestionar los datos generados por sensores y procesados por el módulo de detección de anomalías.

Orange Pi: Miniordenador utilizado como servidor dentro del entorno industrial, encargado de centralizar y procesar los datos recolectados por los sensores.

SCADA (Supervisory Control and Data Acquisition): Sistema de control industrial que permite supervisar y adquirir datos en tiempo real desde los sensores y mecanismos, facilitando la integración con el módulo de detección de anomalías.

STM32: Microcontrolador que gestiona las interacciones entre los sensores y mecanismos industriales, y que juega un papel clave en el flujo de datos hacia el sistema de detección de anomalías.

Tema IR

[2-Modelado de negocio 4](#_Toc167377080)

[2.1 Modelo de dominio 4](#_Toc167377081)

[2.2 Glosario de términos 5](#_Toc167377082)

[2.3 Diagrama de Flujo 6](#_Toc167377083)

[2.4 Reglas del negocio 7](#_Toc167377084)

[3-Captura de requisitos 8](#_Toc167377085)

[3.1 Diagrama de caso de uso del sistema 8](#_Toc167377086)

[3.2 Requisitos Funcionales: 8](#_Toc167377087)

[3.3 Requisitos de Calidad 9](#_Toc167377088)

[3.4 Descripción de alto nivel de los casos de uso 11](#_Toc167377089)

[3.5 Descripción detallada de los casos de uso 15](#_Toc167377090)

Tema DS

[1. Documentación de Ingeniería de Requisitos 1](#_Toc177507987)

[1.1 Breve descripción del negocio 1](#_Toc177507988)

[1.1.1 Diagrama de Actividades 1](#_Toc177507989)

[1.1.2 Modelo de dominio 2](#_Toc177507990)

[1.1.3 Glosario de términos 3](#_Toc177507991)

[1.2 Requisitos Funcionales: 3](#_Toc177507992)

[1.3 Requisitos no funcionales. Problemas frecuentes 4](#_Toc177507993)

[1.3.1 Requisitos de Calidad 5](#_Toc177507994)

[1.3.2 Restricciones 5](#_Toc177507995)

[2. Arquitectura Candidata 6](#_Toc177507996)

[2.1 Estudio de las tecnologías candidatas 6](#_Toc177507997)

[2.2 Posibles lenguajes: 7](#_Toc177507998)

[2.2.1 Tratamiento de los problemas frecuentes mediante la tecnología seleccionada 12](#_Toc177507999)

*[2.3 Framework](#_Toc177508000)* [y bibliotecas 13](#_Toc177508000)

[3. Estilos y patrones de arquitectura identificados 14](#_Toc177508001)

[3.1 Llamada y retorno 15](#_Toc177508002)

[3.1.1 Patrón N-capas 15](#_Toc177508003)

[3.2 Centrado en datos 17](#_Toc177508004)

[3.2.1 Patrón BlackBoard 17](#_Toc177508005)

[3.3 Flujo de datos 18](#_Toc177508006)

[3.3.1 Pipes and Filters 18](#_Toc177508007)

[4. Patrones de diseño 19](#_Toc177508008)

[4.1 Template method 19](#_Toc177508009)

[4.2 Patrón Strategy 20](#_Toc177508010)

[4.3 Patrón Facade 21](#_Toc177508011)

[5. Técnicas de Refactorización 22](#_Toc177508012)

[6. Propuesta de Despliegue 32](#_Toc177508013)

[6.1 Mecanismos de Diseño 33](#_Toc177508014)

Dado que tu proyecto no tiene un "negocio" como tal, sino que se enfoca en un módulo técnico para la detección de anomalías en un entorno industrial experimental, puedes adaptar estos elementos para que reflejen el propósito técnico y funcional del sistema en lugar de un negocio. Aquí te dejo una guía para abordar cada sección:

## ****2-Modelado de Negocio****

Aunque no hay un "negocio", puedes modelar el propósito técnico del sistema como si fuera un entorno operativo. El enfoque debe centrarse en el **contexto del módulo de detección de anomalías** dentro del entorno industrial simulado.

### ****2.1 Modelo de dominio****

El modelo de dominio debe representar los elementos principales del entorno industrial y cómo interactúan entre sí. Por ejemplo:

* **Entidades clave**: Sensores, PLC, SCADA, HMI, Módulo de detección de anomalías, Base de datos.
* **Relaciones**: Cómo los datos fluyen desde los sensores al módulo, y cómo este módulo comunica alertas al SCADA o HMI.

**Ejemplo:** Un diagrama UML de clases que muestre:

* Sensores como fuentes de datos.
* Microcontroladores STM32 como intermediarios entre sensores y PLC.
* PLC como controlador central que envía datos al SCADA y HMI.
* Módulo de detección de anomalías que analiza los datos del sistema.

### ****2.2 Glosario de términos****

Define los términos técnicos clave utilizados en el proyecto. Esto ayuda a establecer un vocabulario común para los lectores del proyecto. Ejemplo:

* **Anomalía**: Cualquier desviación significativa de los patrones operativos normales.
* **PLC**: Controlador Lógico Programable, responsable de la lógica del sistema.
* **SCADA**: Sistema de Supervisión y Adquisición de Datos, utilizado para monitorear y controlar el entorno.
* **HMI**: Interfaz Hombre-Máquina, que muestra información operativa al usuario.

### ****2.3 Diagrama de flujo****

Si no tienes un flujo de "negocio", utiliza un diagrama de flujo para describir **cómo funciona el módulo de detección de anomalías** dentro del sistema. Ejemplo:

1. Captura de datos desde sensores.
2. Procesamiento inicial por los microcontroladores STM32.
3. Transmisión al módulo de detección de anomalías.
4. Análisis y clasificación de datos (normal/anómalo).
5. Comunicación de alertas al SCADA o HMI.

Esto muestra cómo se mueve la información y qué decisiones toma el sistema.

### ****2.4 Reglas del negocio****

En lugar de reglas de negocio, puedes definir **reglas operativas** o condiciones bajo las cuales el sistema debe funcionar. Ejemplo:

* **Reglas de datos**: Los datos deben provenir de sensores configurados previamente y con sincronización correcta.
* **Reglas de alertas**: Las alertas solo deben generarse si se supera un umbral de anomalía definido.
* **Reglas de procesamiento**: El módulo debe procesar datos en tiempo real con una latencia menor a X milisegundos.

## ****3-Captura de Requisitos****

En este caso, captura los requisitos funcionales y no funcionales del módulo de detección de anomalías y su integración con el entorno.

### ****3.1 Diagrama de caso de uso del sistema****

Diseña un diagrama de casos de uso que muestre las interacciones entre los actores y el sistema. Por ejemplo:

* **Actores**: Sensores, Operador (usuario), SCADA, Base de datos.
* **Casos de uso**: Capturar datos, Analizar anomalías, Generar alertas, Visualizar datos en HMI.

### ****3.2 Requisitos Funcionales****

Define los requisitos funcionales del módulo. Ejemplo:

* El sistema debe capturar datos operativos de sensores en tiempo real.
* El módulo debe identificar y clasificar anomalías basándose en patrones históricos.
* El sistema debe generar alertas visibles en el HMI y almacenarlas en la base de datos.

### ****3.3 Requisitos de Calidad****

Estos son los requisitos no funcionales. Ejemplo:

* **Rendimiento**: El módulo debe procesar X registros por segundo con una latencia menor a Y milisegundos.
* **Escalabilidad**: El sistema debe soportar la integración de hasta Z sensores.
* **Confiabilidad**: La tasa de falsos positivos no debe superar el 5%.

### ****3.4 Descripción de alto nivel de los casos de uso****

Para cada caso de uso, describe su propósito y cómo encaja en el sistema. Ejemplo:

* **Captura de datos**: El sistema recoge información de los sensores para análisis.
* **Detección de anomalías**: El módulo analiza los datos en busca de patrones inusuales.
* **Generación de alertas**: Cuando se detectan anomalías, se envía una notificación al operador a través del HMI.

### ****3.5 Descripción detallada de los casos de uso****

Detalla cada caso de uso utilizando un formato estructurado:

* **Nombre del caso de uso**: Capturar datos de sensores.
* **Actor principal**: Sensores.
* **Propósito**: Recoger datos en tiempo real para su análisis.
* **Curso normal**:
  1. Los sensores transmiten datos al microcontrolador STM32.
  2. Los datos son enviados al módulo de detección de anomalías.
* **Curso alternativo**: Si un sensor no responde, se registra un error en el sistema.

Con este enfoque, cada sección estará adaptada al propósito técnico de tu proyecto, sin necesidad de simular un "negocio" tradicional. ¿Quieres que desarrolle un ejemplo completo para alguna sección específica? 😊

# Referencias Bibliográficas

1. Lasi, H., et al., *Industry 4.0.* Business & Information Systems Engineering, 2014. **6**: p. 239-242.

2. Chandola, V., A. Banerjee, and V. Kumar, *Anomaly Detection: A Survey.* ACM Comput. Surv., 2009. **41**.

3. Ahmed, M., A. Mahmood, and J. Hu, *A Survey of Network Anomaly Detection Techniques.* Journal of Network and Computer Applications, 2015. **60**: p. 19-31.

4. Qin, J., *Survey on data-driven industrial process monitoring and diagnosis.* Annual Reviews in Control, 2012. **36**: p. 220–234.

5. Hastie, T., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2009.

6. Grunova, D., et al., *Machine Learning for Anomaly Detection in Industrial Environments*. 2024. 25.

7. Liu, J., et al., *Anomaly Detection Method for Industrial Control System Operation Data Based on Time–Frequency Fusion Feature Attention Encoding.* Sensors, 2024. **24**: p. 6131.

8. Kim, B., et al., *A Comparative Study of Time Series Anomaly Detection Models for Industrial Control Systems.* Sensors, 2023. **23**: p. 21.

9. Mokhtari, S., et al., *A Machine Learning Approach for Anomaly Detection in Industrial Control Systems Based on Measurement Data.* Electronics, 2021. **10**: p. 407.