

Desarrollo de una herramienta basada en IoT para el control y monitoreo del nivel de CO2 en las aulas de la Universidad Industrial de Santander

Iván Daniel Maestre Muza

Trabajo de Grado para Optar al Título de Ingeniero de Sistemas

Director

Gabriel Rodrigo Pedraza Ferreira

PhD. Ciencias de la Computación

Codirector

Henry Andrés Jiménez Herrera

Magister en Ingeniería de Sistemas e Informática

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Ingeniería de Sistemas

Bucaramanga

2023

Tabla de Contenido

		Pág.
Introducción		10
1	Generalidades del proyecto	12
1.1	Título del proyecto	12
1.2	Objetivos	12
1.2.1	<i>Objetivo General</i>	12
1.2.2	<i>Objetivos Específicos</i>	12
1.3	Metodología.....	13
1.3.1	<i>Fase uno: Ambientación tecnológica y contextual</i>	13
1.3.2	<i>Fase dos: Definición de los requerimientos</i>	14
1.3.3	<i>Fase tres: Diseño de la solución (definición arquitectura, diseño detallado)</i>	14
1.3.4	<i>Fase cuatro: Prototipado de la solución</i>	14
1.3.5	<i>Fase cinco: Evaluación de la solución</i>	15
2	Estado del arte	16
3	Marco de referencia	19
3.1	Internet de las cosas	19
3.2	Smart Campus	20
3.3	Gateway	20
3.4	Sensor.....	20
3.5	Interfaz.....	20
3.6	Framework	21
3.7	Extensibilidad.....	21
3.8	Monitoreo de CO2	21
4	Desarrollo del Proyecto	22
4.1	Definición de requerimientos.....	23
4.1.1	<i>Requisitos funcionales</i>	23
4.1.2	<i>Requisitos no funcionales</i>	26
4.2	Selección tecnológica.....	27

4.2.1	<i>Tecnología IoT (Internet de las Cosas)</i>	27
4.2.2	<i>Protocolo de mensajería MQTT</i>	28
4.2.3	<i>Selección de sensores</i>	29
4.3	Diseño de la solución	30
4.3.1	<i>Definición de la arquitectura</i>	30
4.3.2	<i>Diseño detallado</i>	32
4.4	Implementación de la solución	38
4.5	Validación de la solución	48
4.5.1	<i>Resultados de la validación</i>	53
5	Trabajo a futuro	55
6	Conclusiones	56
7	Referencias Bibliográficas	58

Lista de Tablas

	Pág.
Tabla 1. Cumplimiento de los objetivos específicos	22
Tabla 2. Comparativo de Sensores.....	30

Lista de Figuras

	Pág.
Figura 1. Diagrama metodológico	13
Figura 2. El esqueleto de un sistema IoT	19
Figura 3. Arquitectura MQTT.....	28
Figura 4. Diagrama de arquitectura.....	31
Figura 5. Diagrama de flujo de evaluación cruzada.....	32
Figura 6. Interfaz Ingreso.....	33
Figura 7. Interfaz Inicio – Vista Administrador.....	34
Figura 8. Interfaz Inicio – Vista Encargado.....	34
Figura 9. Interfaz Perfil.....	35
Figura 10. Interfaz Gestión Usuarios	35
Figura 11. Interfaz Dashboard	36
Figura 12. Interfaz Agregar Aula.....	36
Figura 13. Interfaz Actualizar Aula	37
Figura 14. Interfaz Recuperar Contraseña	37
Figura 15. Interfaz Cambiar Contraseña.....	38
Figura 16. Conexión Broker MQTT	40
Figura 17. Generar valores aleatorios	41
Figura 18. Envío de datos	42
Figura 19. Código ejecutado.....	43
Figura 20. Visualización en tiempo real – Datos CO2	44
Figura 21. Prototipo Inicio – Login	45
Figura 22. Prototipo Inicio – Vista Administrador.....	45
Figura 23. Prototipo Inicio – Vista Encargado	46
Figura 24. Prototipo - Dashboard.....	46
Figura 25. Prototipo – Gráficas de histograma y barras	47
Figura 26. Prototipo – Gestión de Usuarios.....	47
Figura 27. Prototipo - Perfil	48
Figura 28. Diagrama de flujo – Modelo de aprendizaje automático.....	49
Figura 29. Prototipo - Monitoreo	53

Figura 30. Dato real 54

Figura 31. Dato de predicción..... 54

Lista de Apéndices

(Ver apéndices adjuntos y pueden ser consultados en la base de datos de la Biblioteca UIS)

Apéndice A. Interfaces gráficas de referencia

Apéndice B. Código fuente de la aplicación

Apéndice C. Credenciales de acceso

Resumen

Título: Desarrollo de una herramienta basada en IoT para el control y monitoreo del nivel de co2 en las aulas de la Universidad Industrial de Santander.^{1*}

Autor: Iván Daniel Maestre Muza^{2**}

Palabras clave: Internet de las cosas, framework, software, campus inteligente, CO2.

Descripción:

Además del uso de mascarilla existen diferentes factores que influyen en el contagio de enfermedades ocasionadas por la propagación de aerosoles, como lo es el COVID-19, entre estos factores podemos destacar la duración del encuentro o la ventilación del lugar. Es sabido que en un espacio cerrado con poca circulación de aire hay una mayor propagación de gérmenes, entre ellos el virus SARS COV 2, causante de dicha enfermedad.

Para evitar el contagio del virus es importante poder realizar un monitoreo permanente del CO2 que nos permita saber si el lugar cuenta con buena ventilación, ya que el riesgo asociado es mayor cuando las partículas del virus se quedan en suspensión hasta que son respiradas por alguien más.

En este proyecto se desarrolla una herramienta IoT que sirve para apoyar el monitoreo de CO2 en las aulas de la Universidad Industrial de Santander o en espacios dentro de un campus inteligente, y de esta forma determinar si el aire que se respira se renueva lo suficiente. Esta información es clave para determinar las medidas necesarias que hagan que las aulas sean seguras, garantizando que los ciclos de renovación del aire se cumplan con una ventilación forzada en los casos donde se requiera. Para probar la herramienta, se generaron datos aleatorios sobre niveles de CO2 con el fin de poder monitorear y visualizar cómo varían las mediciones en los ciclos de tiempo que se dispongan. En los ciclos que se evidencie un valor que sobrepase el umbral límite establecido, se envía una alarma indicando que se debe disminuir la cantidad de CO2 en el espacio.

Por último, se usa un modelo de machine learning para dotar a la herramienta de la capacidad de predecir valores futuros, dando cumplimiento a la utilidad que tiene esta herramienta para implementarse en arquitecturas IoT en un campus inteligente.

^{1*} Trabajo de Grado

^{2**} Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática.

Abstract

Title: Development of a tool based on IoT for the control and monitoring of the CO₂ level in the classrooms of the Industrial University of Santander.^{3*}

Author: Iván Daniel Maestre Muza^{4**}

Keywords: Internet of things, framework, software, gateway, smart campus, CO₂.

Description:

In addition to the use of masks, there are different factors that influence the contagion of diseases caused by the spread of aerosols, such as COVID-19, among these factors we can highlight the duration of the meeting or the ventilation of the place. It is known that in an enclosed space with little air circulation there is a greater spread of germs, including the SARS COV 2 virus, which causes this disease.

To avoid the spread of the virus it is important to be able to perform a permanent monitoring of the CO₂ that allows us to know if the place has good ventilation, since the associated risk is greater when the virus particles remain in suspension until they are breathed by someone else.

This project develops an IoT tool to support the monitoring of CO₂ in the classrooms of the Universidad Industrial de Santander or in spaces within a smart campus, and thus determine whether the air that is breathed is sufficiently renewed. This information is key to determine the necessary measures to make the classrooms safe, ensuring that the air renewal cycles are met with forced ventilation in cases where it is required. To test the tool, random data on CO₂ levels were generated in order to monitor and visualize how the measurements vary over the available time cycles. In the cycles that show a value that exceeds the established threshold limit, an alarm is sent indicating that the amount of CO₂ in the space should be reduced.

Finally, a machine learning model is used to provide the tool with the ability to predict future values, fulfilling the usefulness of this tool to be implemented in IoT architectures in a smart campus.

^{3*} Bachelor Thesis

^{4**} Faculty of Physicomechanical Engineering. School of Systems Engineering and Informatics Director: Gabriel Rodrigo Pedraza Ferreira, PhD. Computer Science.

Introducción

La acelerada propagación del COVID-19 ha ocasionado que los especialistas en el tema aborden una mayor cantidad de opciones para contrarrestar dicha enfermedad. El COVID-19 se transmite principalmente por aerosoles, razón por la cual uno de los factores que tiene mayor impacto para evitar el contagio son los espacios bien ventilados, ya que en estos lugares el CO2 se acumula con el tiempo y genera un riesgo muy alto de contagiarse.

En cualquier situación donde se presente la acumulación de personas durante mucho tiempo, se debe tomar medidas rápidamente para aislarnos del riesgo de contagio, comenzando con portar las mascarillas, aumentar la ventilación del lugar y disminuir la duración del encuentro. Por esta razón, es muy importante monitorear el nivel de CO2 en los espacios concurridos, porque nos permite determinar si es necesario tomar medidas como: abrir ventanas, disminuir el aforo del personal, verificar el correcto uso del tapabocas y reducir el tiempo de duración del encuentro.

En este proyecto se hace uso de un modelo de machine learning que permite comparar y tener una proyección de los datos de CO2 recolectados, y con el desarrollo del software podemos tener control y monitoreo a través de IoT. En la actualidad ya existen organizaciones IoT que utilizan sensores para medir el aire, teniendo una buena estabilidad, larga vida útil y con la capacidad de funcionar en varios tipos de entornos, como ejemplo, se ha desarrollado en áreas como: Transporte de alimentos y bebidas, empaquetado, producción de cerveza, instalaciones de almacenamiento en frío, entre otras.

El objetivo final es desarrollar una herramienta IoT que ayude a monitorear los niveles de CO2 en las aulas de la Universidad Industrial de Santander y de esta manera poder tomar las medidas oportunas para que estos espacios sean más seguros, revisando que el aire que se mantiene

se renueve lo suficiente con una buena ventilación. Además, esta herramienta se entrega con el propósito de ser usada a futuro en otros espacios que sean Smart Campus.

1 Generalidades del proyecto

A continuación, se presentan las características generales del proyecto:

1.1 Título del proyecto

“Desarrollo de una herramienta basada en IoT para el control y monitoreo del nivel de CO2 en las aulas de la Universidad Industrial de Santander”

1.2 Objetivos

1.2.1 *Objetivo General*

Desarrollar una herramienta basada en IoT para apoyar el monitoreo de CO2 en las aulas de la Universidad Industrial de Santander.

1.2.2 *Objetivos Específicos*

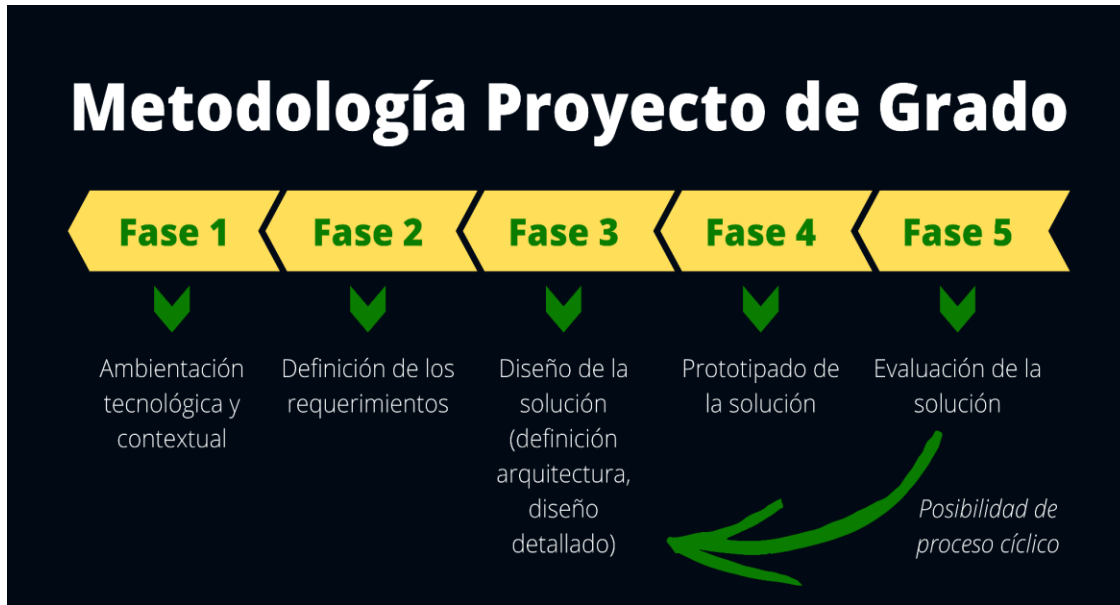
- Definir los requerimientos de la herramienta para medir el CO2 en las aulas de la universidad.
- Tener una selección de los sensores, la tecnología implementada y la plataforma en la que se desarrollará el proyecto.
- Diseñar una arquitectura tecnológica que permita cumplir con los requerimientos previos.
- Desarrollar e implementar un prototipo de la herramienta basado en la arquitectura.
- Evaluar el prototipo implementado de la solución.

1.3 Metodología

Con el fin de realizar el trabajo de investigación, se presentan las siguientes fases a realizar:

Figura 1

Diagrama metodológico



1.3.1 Fase uno: Ambientación tecnológica y contextual

En la primera fase se adquirieron los conocimientos necesarios para dar inicio al proyecto. Se revisó documentación acerca de la tecnología IoT y del contexto de las aplicaciones relacionadas al tema de investigación.

Las actividades realizadas en esta fase fueron:

- Estudio sobre la tecnología IoT (Internet de las Cosas).
- Investigación sobre protocolos de comunicación de mensajería a través de Internet.
- Consulta sobre sensores con medición CO2 que son útiles para el tipo de medición requerido en el caso de estudio.

- Consulta de tutoriales para apoyarse en la implementación sobre cómo se aplican estas tecnologías.

1.3.2 Fase dos: Definición de los requerimientos

En esta etapa se definieron los requerimientos necesarios para dar funcionamiento al software y poder dar cumplimiento con los objetivos propuestos.

Las actividades realizadas en esta fase fueron:

- Realizar listado de especificaciones sobre las funciones, propiedades esenciales y deseables del aplicativo.

1.3.3 Fase tres: Diseño de la solución (definición arquitectura, diseño detallado)

Se diseñó la arquitectura framework de la aplicación y sus componentes detalladamente teniendo en cuenta el alcance del proyecto. Además, se realizó un diseño detallado del software para tener una visualización más precisa de la herramienta a entregar.

Las actividades realizadas en esta fase fueron:

- Definición de la arquitectura utilizada para el caso de estudio.
- Creación de interfaces para tener un diseño detallado de la herramienta.

1.3.4 Fase cuatro: Prototipado de la solución

En esta etapa se desarrolló la codificación de la arquitectura diseñada en la etapa anterior teniendo en cuenta las funcionalidades y características definidas, junto con la correcta integración con la plataforma IoT.

Las actividades realizadas en esta fase fueron:

- Creación de la apariencia y estructura visual del portal web para verificar el diseño y las características especificadas en los requerimientos y arquitectura de la aplicación.

1.3.5 Fase cinco: Evaluación de la solución

En la última fase, se procedió a implementar una evaluación cruzada utilizando técnicas de aprendizaje automático. Con esto, verificamos que la aplicación cumpla con los requerimientos trazados anteriormente.

Para tener una evaluación correcta se realizó un proceso cíclico, esto quiere decir que luego de las evaluaciones realizadas regresamos a la fase tres para verificar el diseño de la solución y corregir el prototipo en caso de ser necesario.

Las actividades realizadas en esta fase fueron:

- Implementar la técnica de aprendizaje automático (ARIMA)
- Integración completa del framework en la plataforma.
- Verificar evaluaciones obtenidas para mejorar diseño, entendimiento, funcionamiento e innovación del prototipo implementado.

2 Estado del arte

El Internet de las cosas (IoT) es un campo de la tecnología que ha crecido rápidamente en los últimos años y se espera que siga creciendo en el futuro cercano. El IoT se refiere a la conexión de dispositivos físicos a internet, permitiendo la recopilación y transmisión de datos en tiempo real.

Las primeras aplicaciones de IoT se remontan a la década de 1990, cuando se desarrollaron los primeros experimentos de telemetría para conectar una serie de dispositivos y recopilar información meteorológica y de profundidad de la nieve [28]. Con el tiempo, esta tecnología ha evolucionado para abarcar una amplia variedad de campos de aplicación, como la manufactura, el transporte, la salud, la agricultura, la seguridad, el hogar inteligente, la energía y muchos más.

La tecnología IoT se ha logrado implementar en diferentes proyectos para los Smart Campus, como lo es el “diseño de una aplicación web extensible para la administración de una plataforma IoT diseñada para Smart Campus” [15], donde el objetivo es mejorar la calidad de vida de los usuarios del campus universitario mediante la transformación de las instalaciones en Smart Campus, además se presenta el diseño de una aplicación web para gestionar una arquitectura basada en tecnología IoT. Esto se logra a través del monitoreo de métricas importantes como la calidad del aire y la temperatura, así como la optimización de recursos físicos y la automatización de labores cotidianas.

Otro proyecto de investigación que podemos tomar de referencia es el “diseño de un framework software extensible para dispositivos tipo gateway integrados en plataformas IoT para Smart Campus” [16], debido a que trata una problemática enfocada en la creciente cantidad de personas en las universidades. Esto representa un desafío en términos de gestión y optimización

de recursos físicos y de control de dispositivos, como la temperatura, la iluminación y la humedad, que cada vez son más numerosos y complicados de administrar. Una gran solución para mejorar la calidad de vida de las personas que utilizan estas instalaciones es la implementación del Internet de las Cosas (IoT) para transformarlas en espacios inteligentes (Smart Campus). En este trabajo de investigación se presenta el diseño de un framework de software extensible que permite a los dispositivos tipo gateway comunicarse y almacenar datos de los sensores y actuadores, conectarse con plataformas IoT especializadas en Smart Campus y facilitar la implementación de diferentes casos de uso de IoT enfocándose en el manejo de los dispositivos finales.

En el entorno de los campus universitarios, se generan una gran cantidad de datos cada día, los cuales pueden ser aprovechados de manera inteligente para beneficiar a la comunidad educativa. Una plataforma IoT centralizada permitiría un acceso rápido a información valiosa, como la seguridad de las diferentes zonas del campus o la gestión de recursos. En otro proyecto de investigación llamado “diseño del componente software backend orientado a una plataforma IoT diseñada para Smart Campus” [17] se desarrolló una arquitectura que permite la integración de un conjunto de gateways y dispositivos, con el propósito de guardar los datos generados para luego ser consumidos por aplicaciones externas. El software fue probado con dispositivos reales instalados en gateways y una plataforma de administración en un entorno real de producción, demostrando su capacidad como un componente cohesivo en una plataforma IoT para Smart Campus.

Con esto se ha logrado diseñar frameworks de software extensible que permite a los dispositivos tipo gateway la comunicación y el almacenamiento de datos enviados y recibidos por sensores, con la capacidad de conectarse con plataformas IoT dirigidas a Smart Campus.

La arquitectura IoT hace posible la interacción con el mundo físico para transformar las detecciones en información útil para las plataformas (capa sensorial), a la vez que muestra las funciones que prestan los dispositivos tipo gateway permitiendo un canal de comunicación, procesamiento y almacenamiento para los dispositivos finales y así, facilita la interacción de estos componentes a las soluciones IoT (capa de gateway). Además de ser una estructura escalable y ofrecer varios servicios, tiene una visión global de todos los dispositivos conectados (capa de backend). Por último, nos ayuda a administrar los sensores, actuadores y gateways en la plataforma (capa de aplicación) para el uso de la información y conocer los estados de los dispositivos [16].

Es importante resaltar también la inclusión de machine learning en este trabajo de investigación, el cual se trata del modelo de aprendizaje automático ARIMA (AutoRegressive Integrated Moving Average), un método estadístico utilizado para analizar y predecir series de tiempo. Se utiliza comúnmente en áreas como la economía, la meteorología y la ingeniería para analizar patrones y predecir futuros valores en función de los valores pasados o datos históricos [25].

Integrar un modelo de machine learning como lo es ARIMA con tecnología IoT puede ser utilizado para predecir valores capturados por sensores en función de los valores pasados. Por ejemplo, este modelo es utilizado para predecir el precio de las acciones en diferentes sectores teniendo como información base los datos de periodos anteriores [26]. Esto permite tomar medidas proactivas y preventivas según el caso de estudio que se desee aplicar.

3 Marco de referencia

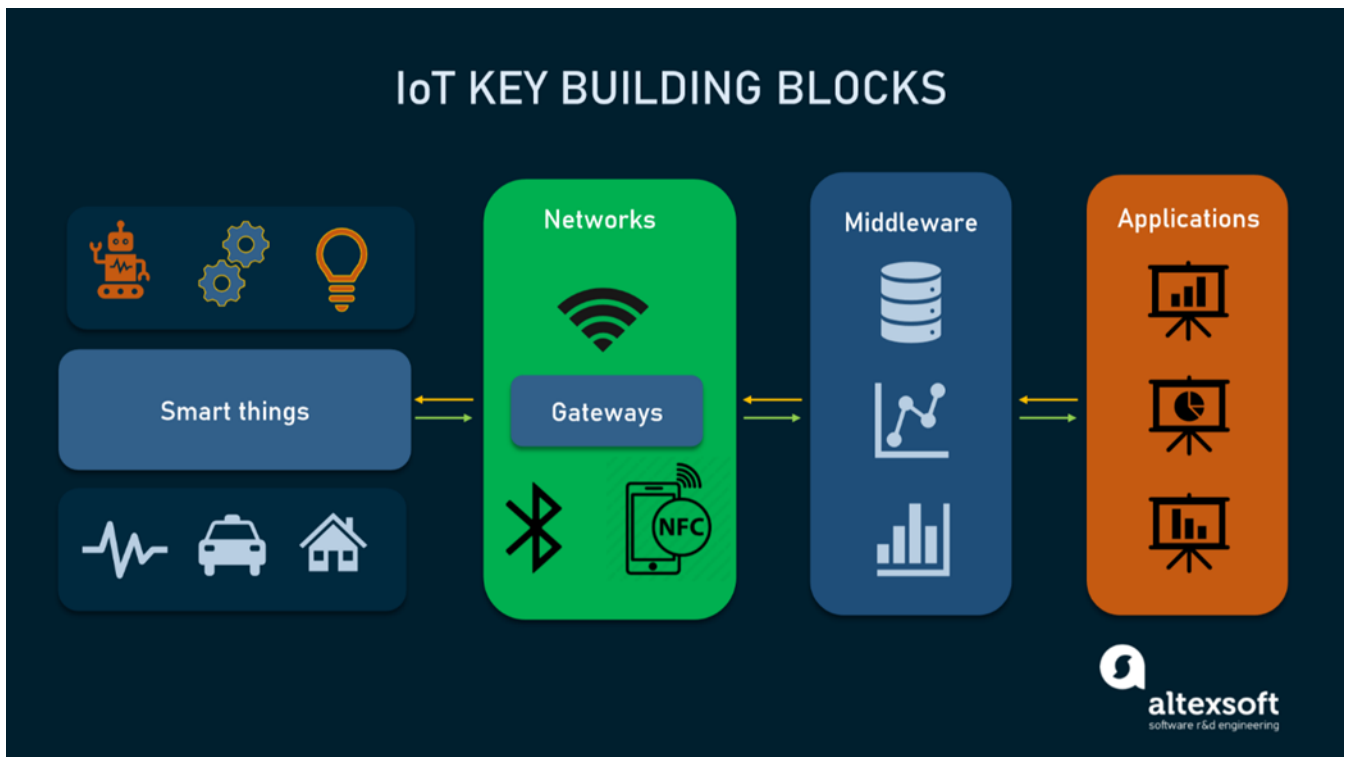
En el proyecto abordaremos temas que nos orienten a tener más claridad sobre el monitoreo del CO2 y cómo se utiliza con la tecnología IoT. En este apartado definimos los términos que nos permiten tener un mayor contexto del fenómeno a tratar.

3.1 Internet de las cosas

El Internet de las cosas (IoT) describe la red de los objetos físicos que incorporan sensores, software y otras tecnologías para intercambiar y conectar datos con otros dispositivos integrados a través de Internet [4].

Figura 2

El esqueleto de un sistema IoT.



Nota. Tomada de <https://www.altexsoft.com/blog/iot-architecture-layers-components/>

En la imagen, se puede detallar un esquema general del Internet de las cosas para un Campus Inteligente. En el mundo actual, los sistemas digitales pueden supervisar y ajustar cada interacción entre las cosas que están conectadas inalámbricamente con una mínima intervención humana, permitiendo una comunicación perfecta entre las personas, procesos y las cosas.

3.2 Smart Campus

Se define como Smart Campus a las instituciones que hacen un énfasis en el uso de la tecnología con el propósito de fomentar el aprendizaje [5]. Desde hace un tiempo, además de utilizarse en empresas e industrias se encontró con la educación, con la prioridad de que las instituciones educativas crecieran en infraestructura y conocimiento.

3.3 Gateway

Es una puerta de enlace que hace de interfaz de conexión entre dispositivos [9]. Permite compartir recursos traduciendo la información del protocolo utilizado en una red inicial al usado en la red de destino.

3.4 Sensor

Es un dispositivo que detecta acciones o estímulos en el entorno y da respuesta a salidas en otro sistema. Es capaz de cambiar una propiedad ante magnitudes físicas o químicas y transformarlas en señales eléctricas [10].

3.5 Interfaz

Una interfaz permite una comunicación entre distintos niveles mediante una conexión entre dos dispositivos, sistemas o programas con intercambio de información [11].

3.6 Framework

Un framework o marco de trabajo, es un esquema o estructura que se establece para desarrollar y organizar un software de una forma determinada. Es un entorno que facilita las habilidades del desarrollador.

3.7 Extensibilidad

En informática, la extensibilidad define la capacidad que tiene un programa para soportar nuevas funcionalidades cuando se incluyan nuevos cambios sin que afecte (o lo más mínimo) a los elementos existentes [12]. En otras palabras, es la flexibilidad que tiene una aplicación o programa para posibles cambios.

3.8 Monitoreo de CO2

Se refiere a la medición de la concentración del dióxido de carbono (CO₂) en el aire. Se utiliza para medir la calidad del aire normalmente en ambientes cerrados y detectar posibles problemas de ventilación.

El monitoreo de CO₂ se realiza mediante sensores que miden la concentración de CO₂ en el aire y son recopilados en datos que se pueden transmitir a través de una red Wi-Fi a una plataforma IoT para su visualización, análisis y control. Esto permite a los usuarios monitorear la concentración de CO₂ en tiempo real y tomar decisiones para mejorar la calidad del aire si es necesario.

4 Desarrollo del Proyecto

El objetivo principal de este proyecto de investigación es desarrollar una herramienta basada en IoT que permita visualizar y monitorear los niveles de CO2 en las aulas de la Universidad Industrial de Santander, cabe resaltar que para validar el prototipo creado se generaron datos aleatorios de niveles de CO2 que permiten visualizar cómo pueden variar las mediciones según los ciclos de tiempos que se disponga.

Tabla 1

Cumplimiento de los objetivos específicos

Objetivo	Cumplimiento
Definir los requerimientos de la herramienta para medir el CO2 en las aulas de la universidad.	El cumplimiento de este objetivo se evidencia en el numeral 4.1.1 correspondiente a los requisitos funcionales y en el numeral 4.1.2 correspondiente a los requisitos no funcionales de la herramienta.
Tener una selección de los sensores, la tecnología implementada y la plataforma en la que se desarrollará el proyecto.	El cumplimiento de este objetivo se evidencia en el numeral 4.2 correspondiente a la selección tecnológica y en el numeral 4.3.2 correspondiente al diseño detallado de la plataforma propuesta para la solución.
Diseñar una arquitectura tecnológica que permita cumplir con los requerimientos previos.	El cumplimiento de este objetivo se evidencia en el numeral 4.3.1 correspondiente a la definición de la arquitectura.
Desarrollar e implementar un prototipo de la herramienta basado en la arquitectura.	El cumplimiento de este objetivo se evidencia en el numeral 4.4 correspondiente a la implementación del prototipado de la solución

	que contiene el desarrollo del portal web como herramienta.
Evaluar el prototipo implementado de la solución.	El cumplimiento de este objetivo se evidencia en el numeral 4.5 correspondiente a la validación de la solución que contiene los resultados de la implementación.

Nota. Esta tabla muestra el cumplimiento de cada uno de los objetivos específicos.

Para cumplir con el desarrollo de este trabajo, procedemos a definir los requerimientos deseables del aplicativo que nos sirve como base teniendo en cuenta los objetivos específicos de este proyecto.

4.1 Definición de requerimientos

4.1.1 *Requisitos funcionales*

Identificación del requisito	RF01
Nombre del requisito	Visualización de aulas
Descripción	El portal web debe permitir seleccionar el aula deseada para visualizar datos dependiendo del edificio y piso seleccionado.
Prioridad	Alta

Identificación del requisito	RF02
Nombre del requisito	Vinculación de la API
Descripción	El aplicativo debe permitir la vinculación con la API del sensor para cargar los datos.
Prioridad	Alta

Identificación del requisito	RF03
Nombre del requisito	Estado del sensor
Descripción	La interfaz de visualización de la información del aula debe mostrar en tiempo real la última captura del nivel de CO2 del sensor, de tal forma que si durante la visualización de la interfaz se toma un nuevo registro debe mostrarse inmediatamente.
Prioridad	Alta

Identificación del requisito	RF04
Nombre del requisito	Gestión de usuarios
Descripción	El portal web debe permitir la gestión de usuarios con las funcionalidades de: agregar, eliminar y asignar perfil (<i>administrador, encargado</i>) desde un usuario con el perfil de ADMINISTRADOR . Además, todos los usuarios registrados podrán acceder al portal a través de una interfaz de inicio de sesión con las credenciales de acceso: usuario y contraseña.
Prioridad	Media

Identificación del requisito	RF05
Nombre del requisito	Histórico de niveles de CO2
Descripción	La interfaz de visualización de la información de las aulas debe construir una gráfica con el histórico de todos los registros de los niveles de CO2 del último mes. Por otra parte, se podrá modificar los rangos de tiempo de la gráfica para ampliar los datos. La gráfica podrá descargarse en un archivo con formato PNG por si el usuario lo requiere.
Prioridad	Alta

Identificación del requisito	RF06
Nombre del requisito	Monitoreo de los datos
Descripción	El portal web debe permitir visualizar la evaluación de los datos obtenidos de los niveles de CO2, para así obtener una comparación con el modelo de aprendizaje automático implementado que ayude a monitorear las mediciones realizadas.
Prioridad	Alta

4.1.2 *Requisitos no funcionales*

Identificación del requisito	RNF01
Nombre del requisito	Diseño de portal web intuitivo
Descripción	Ofrecer un portal web que sea de fácil orientación y ejecución en los procesos. De manera que se utilice de forma sencilla y ágil para los usuarios.
Prioridad	Media

Identificación del requisito	RNF02
Nombre del requisito	Alto rendimiento
Descripción	Ofrecer una navegación rápida y fluida en la ejecución de los procesos realizados por los usuarios.
Prioridad	Media

4.2 Selección tecnológica

4.2.1 Tecnología IoT (*Internet de las Cosas*)

La tecnología IoT es una red de dispositivos conectados que están equipados con tecnologías de comunicación inalámbrica para que puedan transmitir y recibir datos a través de internet. Estos dispositivos están conectados entre sí y a una plataforma de software centralizada para recopilar y analizar los datos recogidos, lo que permite a los usuarios automatizar procesos y tomar decisiones basadas en datos en tiempo real.

En un entorno universitario, la tecnología IoT puede utilizarse para el control automatizado de la iluminación y el clima, el seguimiento de la asistencia de los estudiantes, y el monitoreo del uso de los recursos tecnológicos [5]. También ayuda a mejorar la seguridad en el campus, como el monitoreo de las cámaras de seguridad o la detección de disparos para alertar automáticamente a los estudiantes y el personal [14]. En este proyecto nos dirigimos a mejorar la ventilación de las

aulas de la Universidad Industrial de Santander con ayuda de unos sensores que nos permiten tomar mediciones de niveles de CO2.

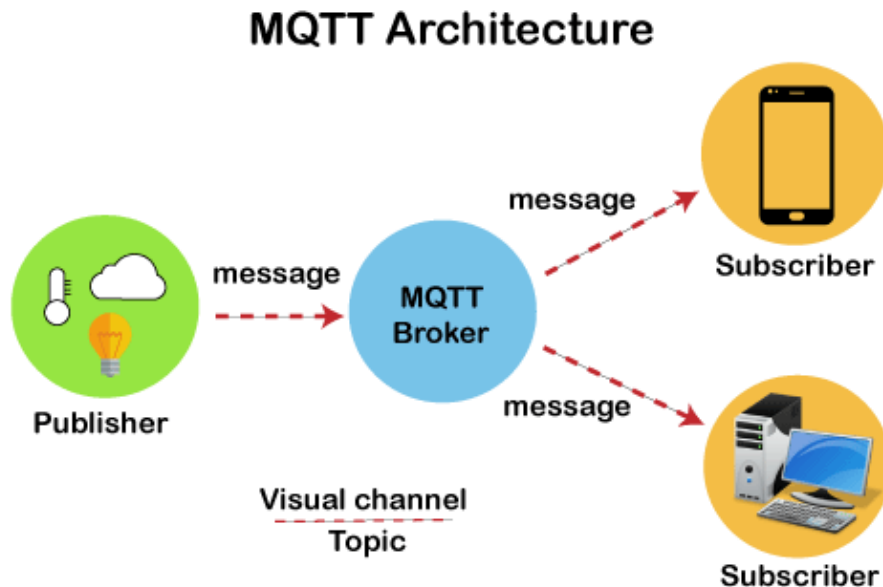
4.2.2 Protocolo de mensajería MQTT

MQTT (Message Queue Telemetry Transport) es un protocolo de comunicación de mensajería de código abierto diseñado que sirve para la conectividad de dispositivos con tecnología IoT. Entre sus utilidades, está la de transmitir datos de sensores y dispositivos de control a través de una red Wi-Fi.

Este protocolo de comunicación se basa en un modelo de publicación (publisher) y suscripción (subscriber), en el cual los dispositivos se suscriben a uno o varios tópicos y reciben mensajes de otros dispositivos que publican la información en esos tópicos.

Figura 3

Arquitectura MQTT



Nota. Tomada de <https://www.javatpoint.com/mqtt-protocol>

Agregado a lo anterior, MQTT tiene varias características que lo hacen adecuado para IoT, como una pequeña cantidad de código, un bajo consumo de energía y una alta fiabilidad. También tiene soporte para seguridad mediante el uso de autenticación y encriptación, lo cual es importante para garantizar la privacidad y la seguridad de los datos.

4.2.3 Selección de sensores

Antes de seleccionar el sensor adecuado para el trabajo de investigación, es importante recalcar el nivel de importancia que tienen estos dispositivos para realizar mediciones precisas y puedan detectar los niveles de CO₂. Estos son algunos puntos a destacar:

- Los sensores pueden detectar los niveles de CO₂ en el aire y en espacios cerrados. Esto es importante ya que la concentración de CO₂ en el aire aumenta cuando la ventilación en el espacio no es la adecuada, lo cual puede ser perjudicial para la salud de las personas.
- Los sensores de CO₂ permiten obtener lecturas confiables y precisas. Esto debido a que utilizan tecnologías específicas para medir la concentración de dióxido de carbono en el aire.
- Los sensores de CO₂ permiten monitorear en tiempo real. Ya que pueden transmitir los datos recopilados a una plataforma para controlar y visualizar en tiempo real, lo que permite a los usuarios del aplicativo monitorear los niveles de CO₂ en tiempo real y tomar decisiones para mejorar la calidad del aire cuando sea necesario.
- Los sensores de CO₂ se usan en las industrias para mejorar la eficiencia de los procesos de producción. Además, ayuda a detectar fugas de gases tóxicos y perjudiciales para las personas y el medio ambiente.

A continuación, se muestra un comparativo realizado de sensores de CO₂:

Tabla 2*Comparativo de Sensores*

Sensores	Corriente media (mA)	Rango de medición (ppm)	Tiempo de respuesta (s)	Tiempo de precalentamiento (min)	Peso (g)	Temperatura de detección (°C)	Precio (US)
MH-Z14A NDIR CO2	< 60 mA	0 - 10000 ppm	90 s - 120 s	3 min	15g	0°C a 50°C	\$27.98
MH-410D NDIR CO2	< 85 mA	0 - 6000 ppm	30 s - 90 s	3 min	35g	-20°C a 60°C	\$246.00
MH-Z19B NDIR CO2	< 60 mA	400 - 5000 ppm	< 120 s	3 min	5g	0°C a 50°C	\$35.00
MH-Z16 NDIR CO2	< 85 mA	0 - 6000 ppm	30 s - 90 s	3 min	21g	0°C a 50°C	\$45.99

Nota. Adaptado de [18], [19], [29], [30], [31], [32], [33] y [34].

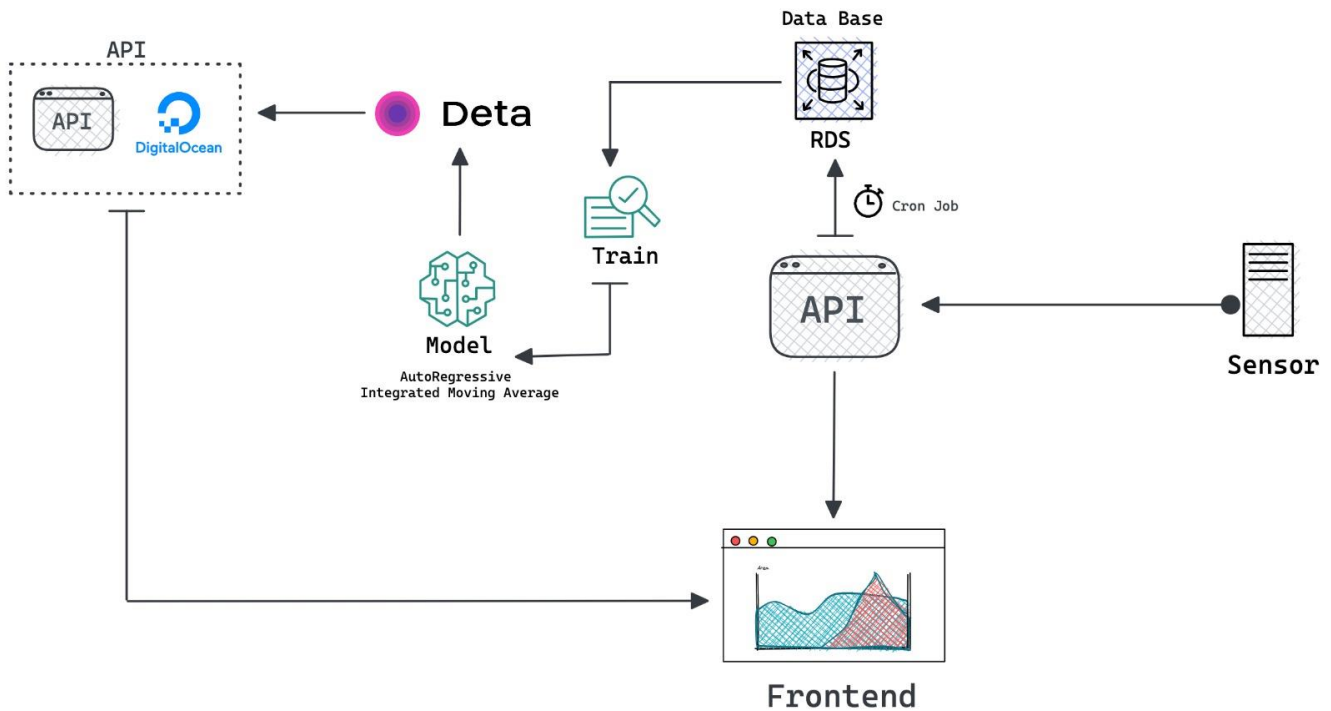
Luego revisar los diferentes sensores que pueden servir para realizar medidas de CO₂, recomendamos que el mejor sensor para realizar mediciones en un aula de clase es el MH-Z14A NDIR CO₂, debido a que sus características técnicas cuentan con la capacidad de realizar el trabajo requerido, además de tener un precio bajo con respecto a demás sensores.

4.3 Diseño de la solución

4.3.1 Definición de la arquitectura

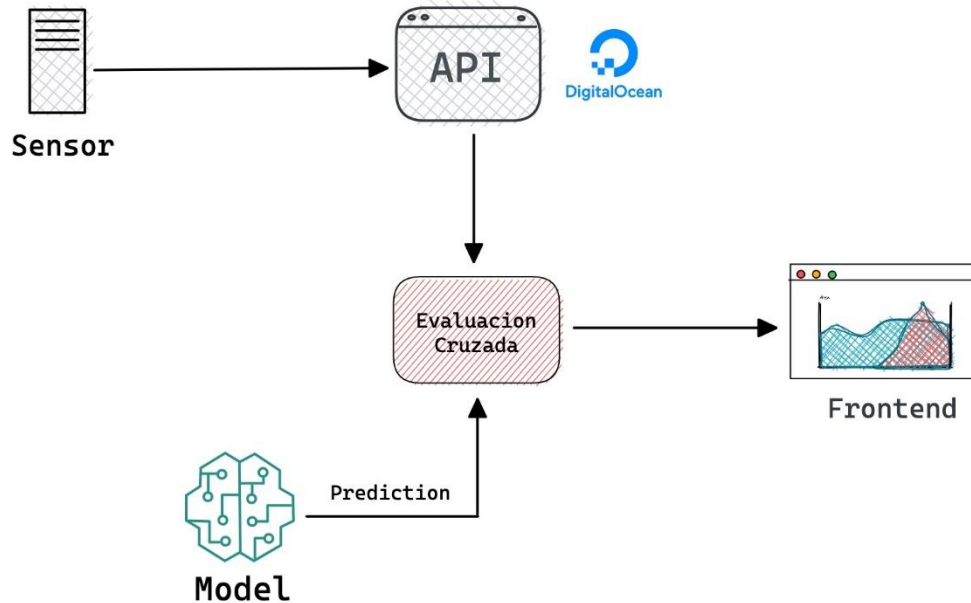
Tener una arquitectura de referencia es muy importante ya que proporciona una representación visual y comprensible de la lógica detrás de la estructura de un sistema o proyecto en general. Además, ayuda a comprender la visión general y los objetivos específicos del proyecto, a la vez de poder definir las relaciones y dependencias entre sus componentes. También es de gran utilidad para identificar posibles problemas de diseño que permite una planificación y mantenimiento del sistema a largo plazo.

A continuación, presentamos el diagrama de dicha arquitectura:

Figura 4*Diagrama de arquitectura*

- **Sensor:** Dispositivo donde se recogen los datos
- **RDS:** Base de datos relacional
- **Train:** Entrenamiento de los datos recolectados
- **Model:** Modelo de machine learning
- **Deta:** Gestor de Base de Datos SQL
- **Digital Ocean:** Droplet (Compute Cloud)

Teniendo como base dicha arquitectura podemos poner en ejecución este proyecto. Adicionalmente, podemos agregar un diagrama de flujo de la evaluación cruzada que existe entre los datos recolectados y los datos obtenidos de las predicciones del modelo. Esta evaluación de los datos es mostrada en el Frontend del aplicativo.

Figura 5*Diagrama de flujo de evaluación cruzada*

4.3.2 *Diseño detallado*

En esta fase se describe el diseño gráfico del prototipo que se va a utilizar para la implementación del proyecto. Un diseño detallado nos permite tener una guía del portal web para ofrecer un software que sea intuitivo y cuente con una buena experiencia del usuario. A continuación, algunos puntos a resaltar sobre su importancia:

- El diseño gráfico es la primera impresión que los usuarios perciben al entrar a cualquier aplicación, portal o sitio web. El diseño puede ser sencillo, pero lo importante es tener uno que sea atractivo y profesional para lograr transmitir confianza a los usuarios.
- Un buen diseño gráfico contribuye a una experiencia de usuario satisfactoria y fácil de usar. Un diseño claro y organizado facilita una navegación intuitiva de la información en el portal.

- El diseño gráfico también es importante para transmitir la identidad de la empresa o marca, esta debe ser coherente con los detalles incluidos en el sitio, como lo es la imagen utilizada como logo/ícono en el portal web, así como otras características.

A continuación, se presentan las interfaces gráficas de referencia que se realizaron con la herramienta en línea draw.io (Diagramas.net):

Figura 6

Interfaz Ingreso



Figura 7

Interfaz Inicio - Vista Administrador

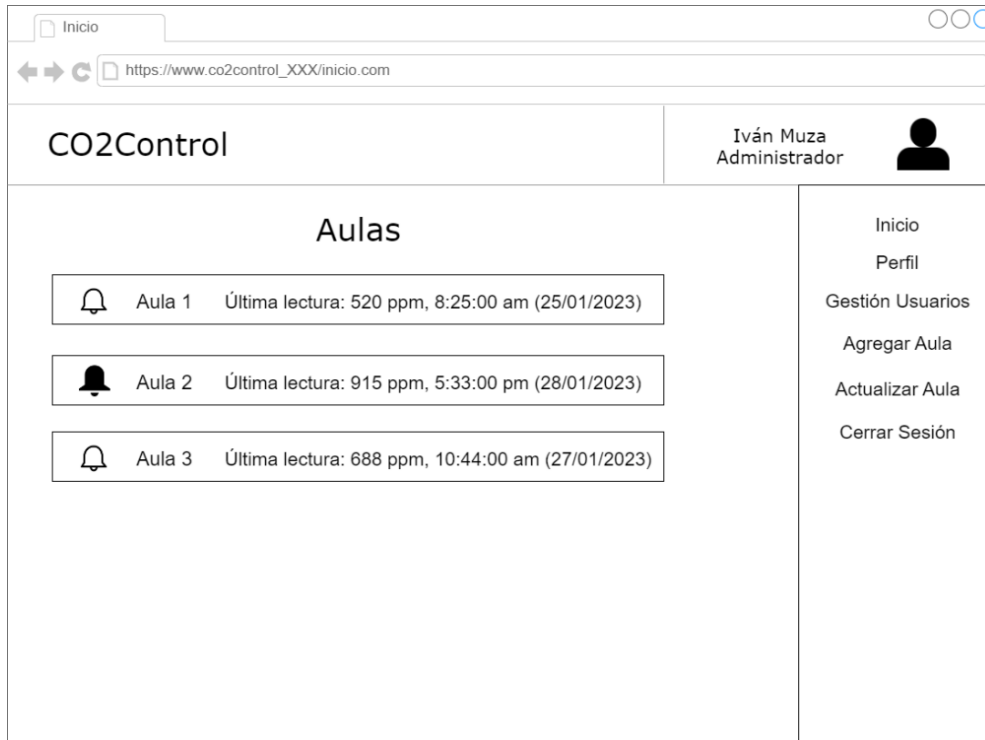


Figura 8

Interfaz Inicio - Vista Encargado

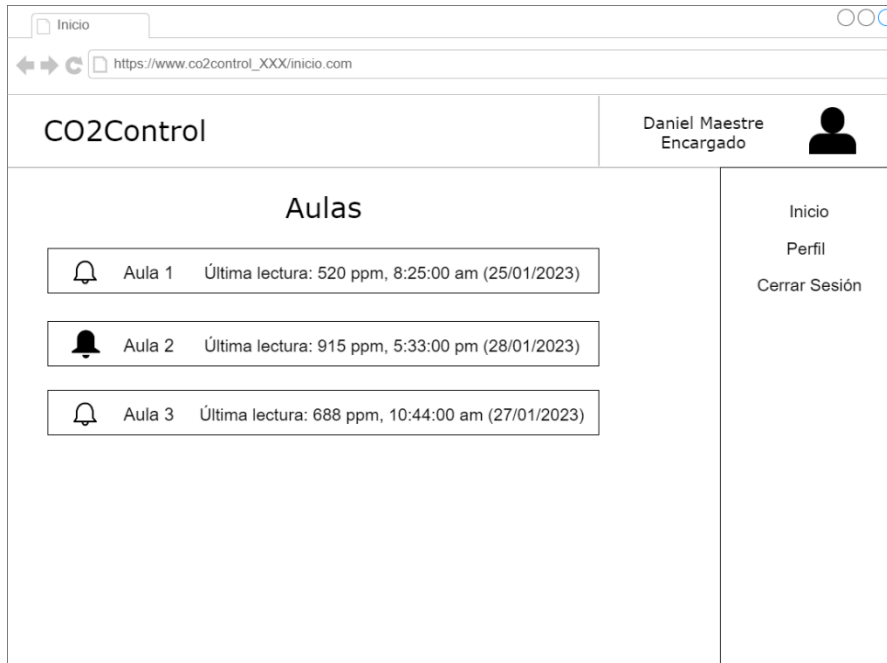


Figura 9

Interfaz Perfil

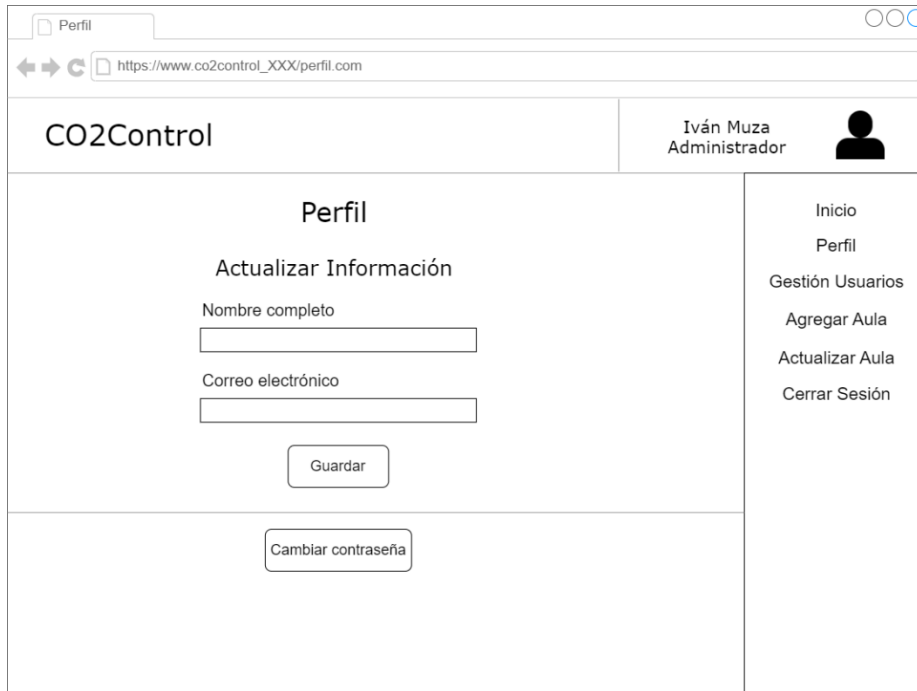


Figura 10

Interfaz Gestión Usuarios

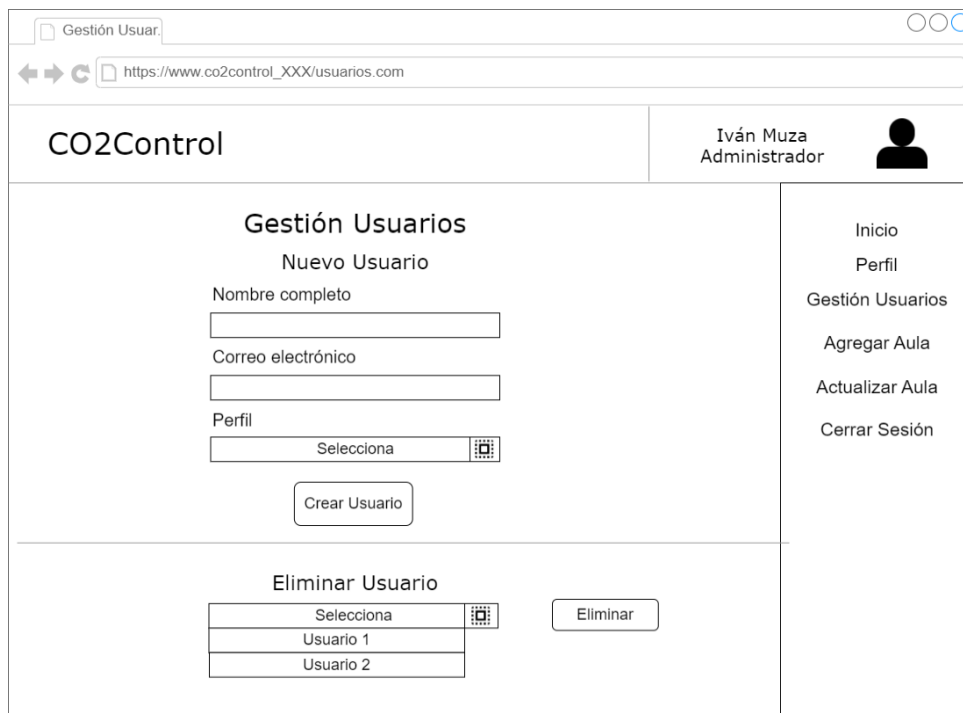


Figura 11

Interfaz Dashboard

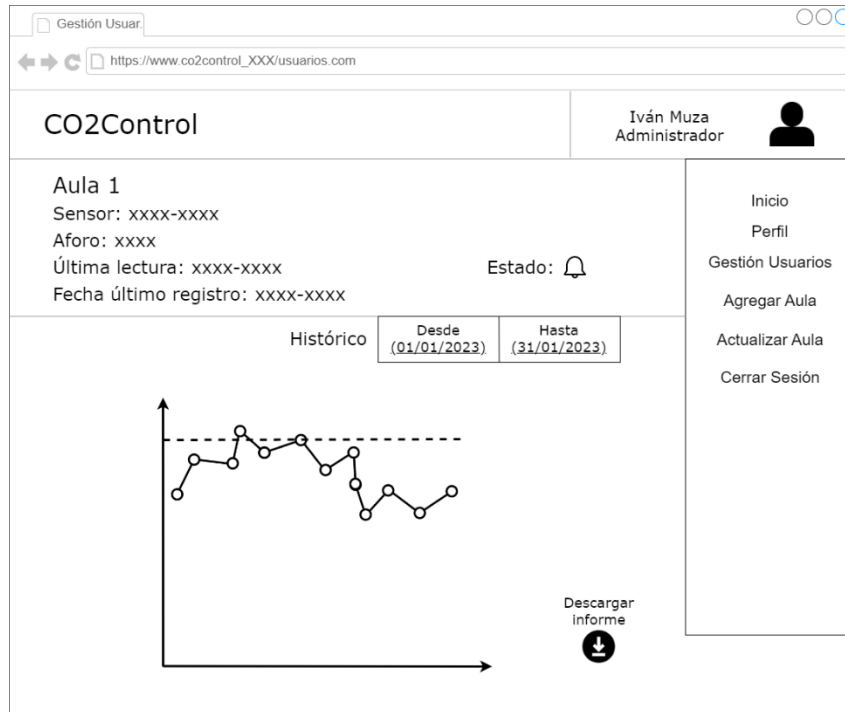


Figura 12

Interfaz Agregar Aula

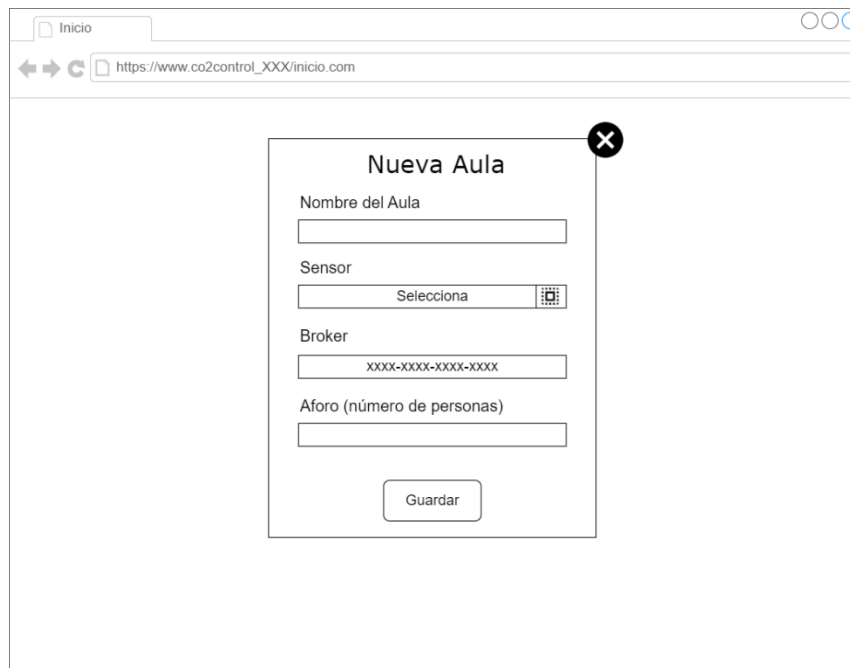


Figura 13*Interfaz Actualizar Aula*

Inicio

https://www.co2control_XXX/inicio.com

Actualizar Aula

Seleccionar Aula
Aula 2

Nombre del Aula
Aula 2

Sensor
xxxx-xxxx

Broker
xxxx-xxxx-xxxx-xxxx

Aforo (número de personas)
xxxx

Eliminar Actualizar

Figura 14*Interfaz Recuperar Contraseña*

Inicio

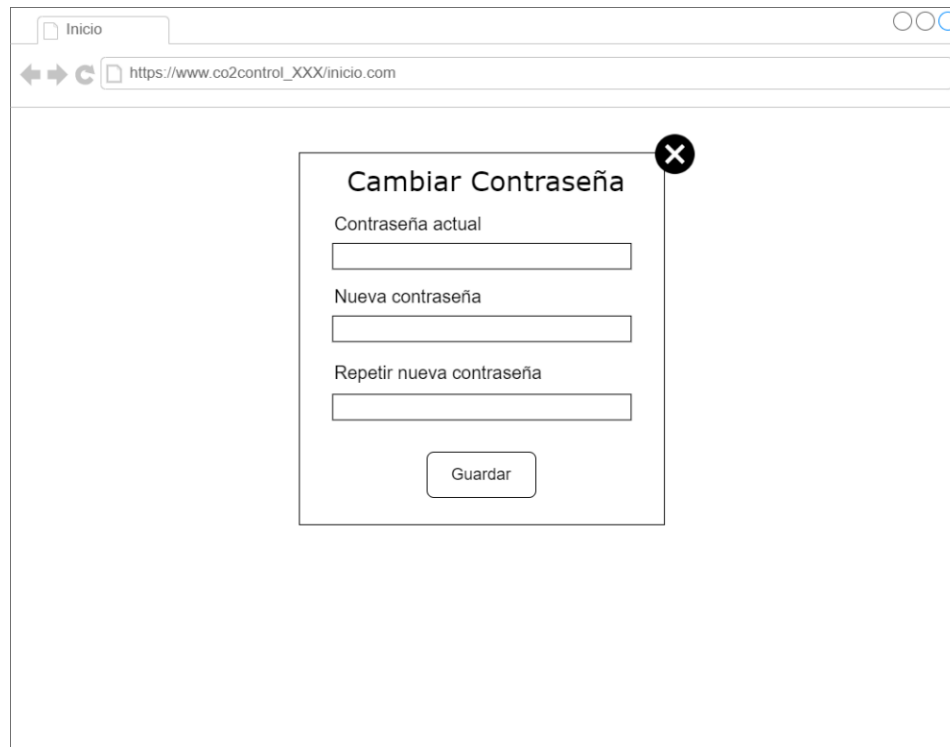
https://www.co2control_XXX/inicio.com

Recuperar Contraseña

Tu contraseña se ha cambiado por:
co2control

Dale clic en Continuar para ingresar
y luego cambia la contraseña.

Continuar

Figura 15*Interfaz Cambiar Contraseña*

The image shows a web browser window with a single tab titled 'Inicio'. The address bar contains the URL 'https://www.co2control_XXX/inicio.com'. The main content area displays a modal form titled 'Cambiar Contraseña' with a close button (X) in the top right corner. The form includes three text input fields labeled 'Contraseña actual', 'Nueva contraseña', and 'Repetir nueva contraseña', followed by a 'Guardar' button.

4.4 Implementación de la solución

En base al inciso anterior, en donde se realizó el diseño de las interfaces gráficas del portal web, ahora podemos proceder a la creación de la apariencia y la estructura visual. En esta fase, nos basamos en la creación del prototipo, donde veremos a más a detalle los elementos gráficos como botones, menús, íconos, tipografía, colores, entre otros, y su disposición en la página.

En un proyecto de investigación, el prototipado de la solución es un proceso de desarrollo en el que se crean versiones o modelos simplificados de una solución propuesta para resolver un problema. Este modelo permite probar y evaluar la solución antes de su implementación completa, dando como resultado una solución más eficiente y efectiva.

En este trabajo de grado se ofrece este prototipo como herramienta para futuros trabajos de investigación o implementaciones deseadas con el objetivo de probar la solución con datos reales y obtener una comprensión más profunda de su comportamiento.

El prototipado de la solución se desarrolló con ayuda de Streamlit, un framework de código abierto que sirve para crear aplicaciones web de forma rápida y sencilla. Se utiliza principalmente para crear aplicaciones de visualización de datos interactivas, aplicaciones web de demostración y prototipos de modelos de aprendizaje automático (tal como se implementa en este caso de estudio).

Streamlit proporciona una interfaz de usuario muy fácil de usar que permite a los desarrolladores crear aplicaciones web interactivas utilizando Python, además de poder crear aplicaciones web utilizando herramientas familiares como Pandas, Numpy y Matplotlib.

Las aplicaciones creadas con Streamlit se ejecutan en un servidor web y se pueden compartir fácilmente en línea. Streamlit se encarga de los aspectos técnicos subyacentes de la creación de aplicaciones web, como la configuración del servidor, la gestión de la caché y la actualización en tiempo real de los datos de la aplicación.

Antes de construir el prototipado, se desarrolló un programa en el cual nos conectamos a un broker MQTT público utilizando WebSocket (un método de comunicación instantánea para aplicaciones web a través de conexión TCP) con el que podemos ir generando valores aleatorios y mostrar en tiempo real publicando los datos en una página web sencilla, la cual se diseñó con HTML y JavaScript.

Para el programa se utilizó lenguaje C#, el cual cuenta con una excelente integración con el entorno de desarrollo de Microsoft como Visual Studio, lo que facilita la creación de aplicaciones para Windows. Entre algunos beneficios de usar C# encontramos que es un lenguaje

que ofrece una seguridad de tipos sólida, lo que significa que se detectan errores de tipos en tiempo de compilación, lo que ayuda a prevenir errores en tiempo de ejecución. Adicionalmente, cuenta con un soporte nativo para programación concurrente y paralela, lo que permite a los programadores escribir código que se ejecuta en múltiples hilos al mismo tiempo.

A continuación, veremos partes del código donde realizamos la conexión al bróker MQTT para luego generar datos aleatorios dentro de un umbral predefinido:

Figura 16

Conexión Broker MQTT

```
connectMQTT.js x
1 //const client = mqtt.connect('ws://public.mqtthq.com:8083/mqtt')
2 const client = mqtt.connect('ws://broker.emqx.io:8083/mqtt')
3
4 function connectEvent(){
5     client.subscribe('Room/mqtt/#', function (err) {
6         if (!err) {
7             //client.publish('testIvan/room', 'Conectado')
8         }
9     })
10 }
11
12 function messageEvent(topic, message){
13     if (topic == 'testIvan/room'){
14         console.log(message.toString());
15     }
16     const node = document.createElement("h5");
17     const textnode = document.createTextNode(topic + " - " + message.toString());
18     node.appendChild(textnode);
19     document.getElementById("myDiv").appendChild(node);
20 }
21
22 client.on('connect', connectEvent)
23 client.on('message', messageEvent)
```

Figura 17

Generar valores aleatorios

```
static void Publish(MqttClient client, string topic, string msg)
{
    client.Publish(topic, System.Text.Encoding.UTF8.GetBytes(msg));
    Console.WriteLine("Send `{0}` to topic `{1}`", msg, topic);
}

static int rndTimeFrame()
{
    Random p = new Random();
    return p.Next(1, 10);
}

static float rndLevelCO2()
{
    Random p = new Random();
    return p.Next(400, 1000);
}

static void sleep()
{
    System.Threading.Thread.Sleep(rndTimeFrame() * 1000);
}

public static DateTime UnixTimeStampToDateTime(double unixTimeStamp)
{
    // Unix timestamp is seconds past epoch
    DateTime dateTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
    dateTime = dateTime.AddSeconds(unixTimeStamp).ToLocalTime();
    return dateTime;
}
```

Figura 18*Envío de datos*

```
static void startSendMessage(MqttClient client, string topic)
{
    while (true)
    {
        float lvCo2 = rndLevelCO2();
        //DateTime now = DateTime.Now;
        long unixTime = DateTimeOffset.UtcNow.ToUnixTimeSeconds();

        var co2capture = new co2Capture
        {
            lvlCo2 = lvCo2,
            timeUnix = unixTime,
            timeStr = UnixTimeStampToDateTime(unixTime).ToString("dd/MM/yyyy HH:mm:ss"),
        };

        string fileName = "co2capture.json";
        string jsonString = JsonSerializer.Serialize(co2capture);
        File.WriteAllText(fileName, jsonString);

        Publish(client, topic, File.ReadAllText(fileName));
        sleep();
    }
}

static void Subscribe(MqttClient client, string topic)
{
    client.MqttMsgPublishReceived += client_MqttMsgPublishReceived;
    client.Subscribe(new string[] { topic }, new byte[] { MqttMsgBase.QOS_LEVEL_AT_MOST_ONCE });
}

static void client_MqttMsgPublishReceived(object sender, MqttMsgPublishEventArgs e)
{
    string payload = System.Text.Encoding.Default.GetString(e.Message);
    Console.WriteLine("Received `{0}` from `{1}` topic", payload, e.Topic.ToString());
}
```

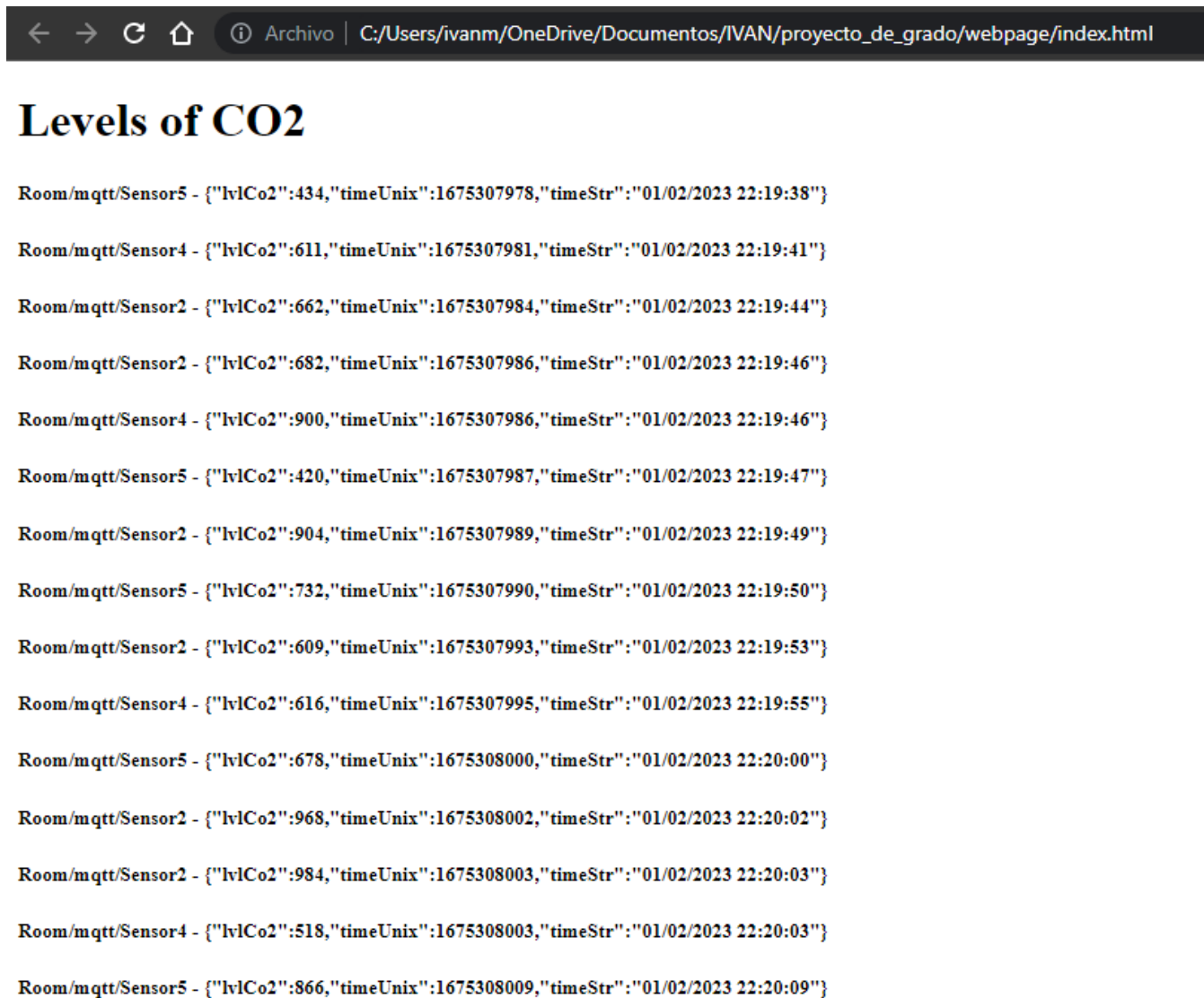
Figura 19

Código ejecutado

```
99     static void Main(string[] args)
100     {
101         for (int i = 1; i <= 5; i++)
102         {
103             String nameSensor = "Sensor" + i;
104             string broker = "broker.emqx.io";
105             int port = 1883;
106             string topic = "Room/mqtt/" + nameSensor;
107             string clientId = Guid.NewGuid().ToString();
108             string username = "emqx";
109             string password = "public";
110             Console.WriteLine(clientId);
111
112             MqttClient client = ConnectMQTT(broker, port, clientId, username, password);
113             Subscribe(client, topic);
114             var t = Task.Run(() =>
115             {
116                 startSendMessage(client, topic);
117             });
118         }
119     }
120 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POLYGLOT NOTEBOOK

```
Connected to MQTT Broker
Send {"lvlCo2":593,"timeUnix":1675307975,"timeStr":"01/02/2023 22:19:35"} to topic `Room/mqtt/Sensor5`
Received {"lvlCo2":593,"timeUnix":1675307975,"timeStr":"01/02/2023 22:19:35"} from `Room/mqtt/Sensor5` topic
Send {"lvlCo2":617,"timeUnix":1675307976,"timeStr":"01/02/2023 22:19:36"} to topic `Room/mqtt/Sensor4`
Received {"lvlCo2":617,"timeUnix":1675307976,"timeStr":"01/02/2023 22:19:36"} from `Room/mqtt/Sensor4` topic
Send {"lvlCo2":434,"timeUnix":1675307978,"timeStr":"01/02/2023 22:19:38"} to topic `Room/mqtt/Sensor5`
Received {"lvlCo2":434,"timeUnix":1675307978,"timeStr":"01/02/2023 22:19:38"} from `Room/mqtt/Sensor5` topic
Send {"lvlCo2":611,"timeUnix":1675307981,"timeStr":"01/02/2023 22:19:41"} to topic `Room/mqtt/Sensor4`
Received {"lvlCo2":611,"timeUnix":1675307981,"timeStr":"01/02/2023 22:19:41"} from `Room/mqtt/Sensor4` topic
Send {"lvlCo2":662,"timeUnix":1675307984,"timeStr":"01/02/2023 22:19:44"} to topic `Room/mqtt/Sensor2`
Received {"lvlCo2":662,"timeUnix":1675307984,"timeStr":"01/02/2023 22:19:44"} from `Room/mqtt/Sensor2` topic
Send {"lvlCo2":682,"timeUnix":1675307986,"timeStr":"01/02/2023 22:19:46"} to topic `Room/mqtt/Sensor2`
Received {"lvlCo2":682,"timeUnix":1675307986,"timeStr":"01/02/2023 22:19:46"} from `Room/mqtt/Sensor2` topic
Send {"lvlCo2":900,"timeUnix":1675307986,"timeStr":"01/02/2023 22:19:46"} to topic `Room/mqtt/Sensor4`
Received {"lvlCo2":900,"timeUnix":1675307986,"timeStr":"01/02/2023 22:19:46"} from `Room/mqtt/Sensor4` topic
Send {"lvlCo2":420,"timeUnix":1675307987,"timeStr":"01/02/2023 22:19:47"} to topic `Room/mqtt/Sensor5`
Received {"lvlCo2":420,"timeUnix":1675307987,"timeStr":"01/02/2023 22:19:47"} from `Room/mqtt/Sensor5` topic
```


Figura 20*Visualización en tiempo real - Datos CO2*


Tomando el anterior desarrollo como base, continuamos con la creación del prototipado.

A continuación, se presentan algunas capturas del prototipo desarrollado con Streamlit:

Figura 21

Prototipo - Login


CO2 Control Authentication

Login 

Username


Password
  7/15

Figura 22

Prototipo Inicio - Vista Administrador

CO2 Control Ivan Daniel Administrator

Aulas | Gestion de Usuarios | Perfil | Bases de Datos | Cerrar Sesion

Seleccionar edificio: | Seleccionar piso:

Informacion Aulas

- ▶ **Aula 1 | Last Lecture: 02-12-2023 | Co2 Levels: 869ppm** ●
- ▶ Aula 2 | Last Lecture: 02-12-2023 | Co2 Levels: 603ppm ●
- ▶ Aula 3 | Last Lecture: 02-12-2023 | Co2 Levels: 862ppm ●

● ppm: [1000,2000] — ● ppm: [800,999] — ● ppm: [400,799] — ○ error de conexion

Figura 23

Prototipo Inicio - Vista Encargado

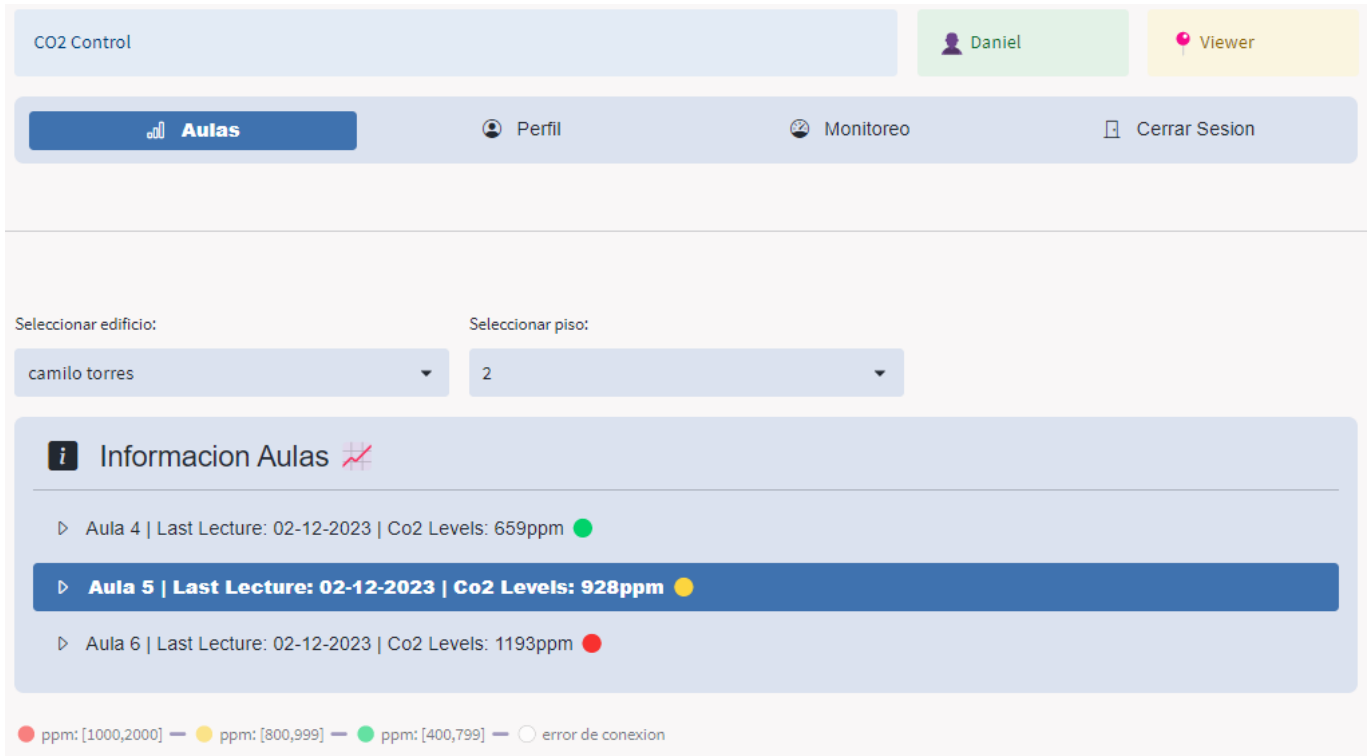


Figura 24

Prototipo - Dashboard

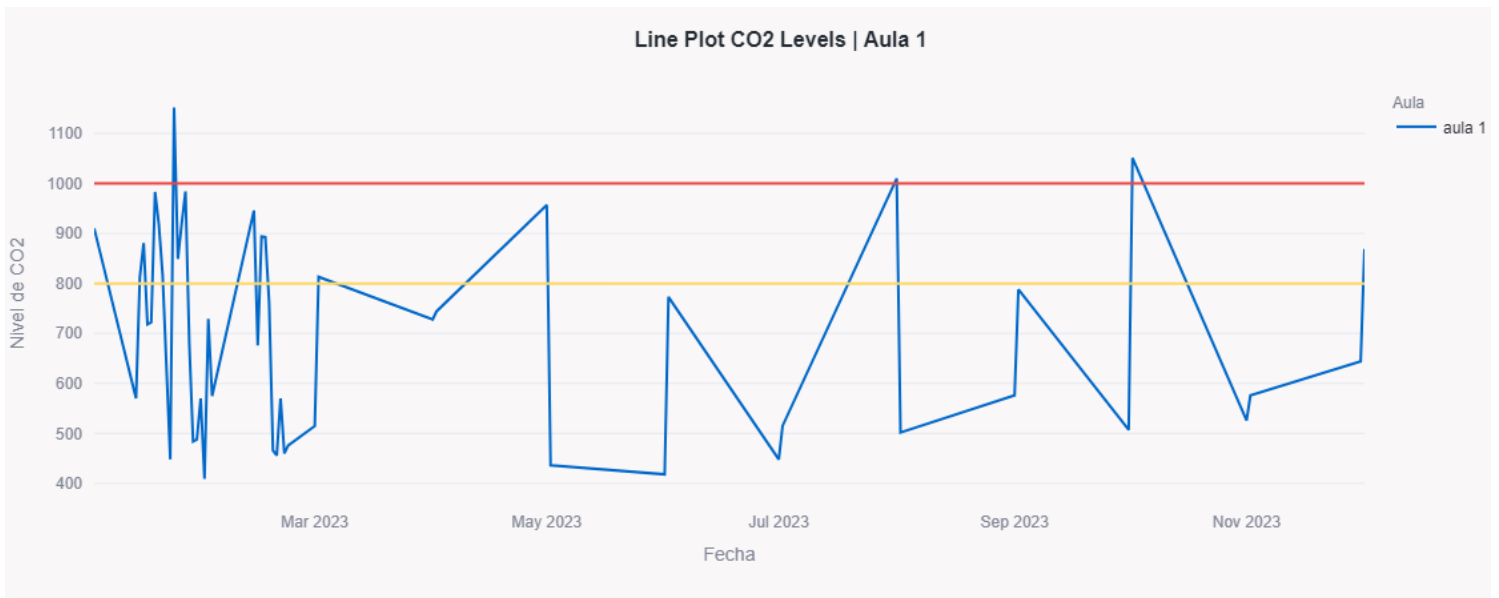


Figura 25

Prototipo - Gráficas de histograma y barras

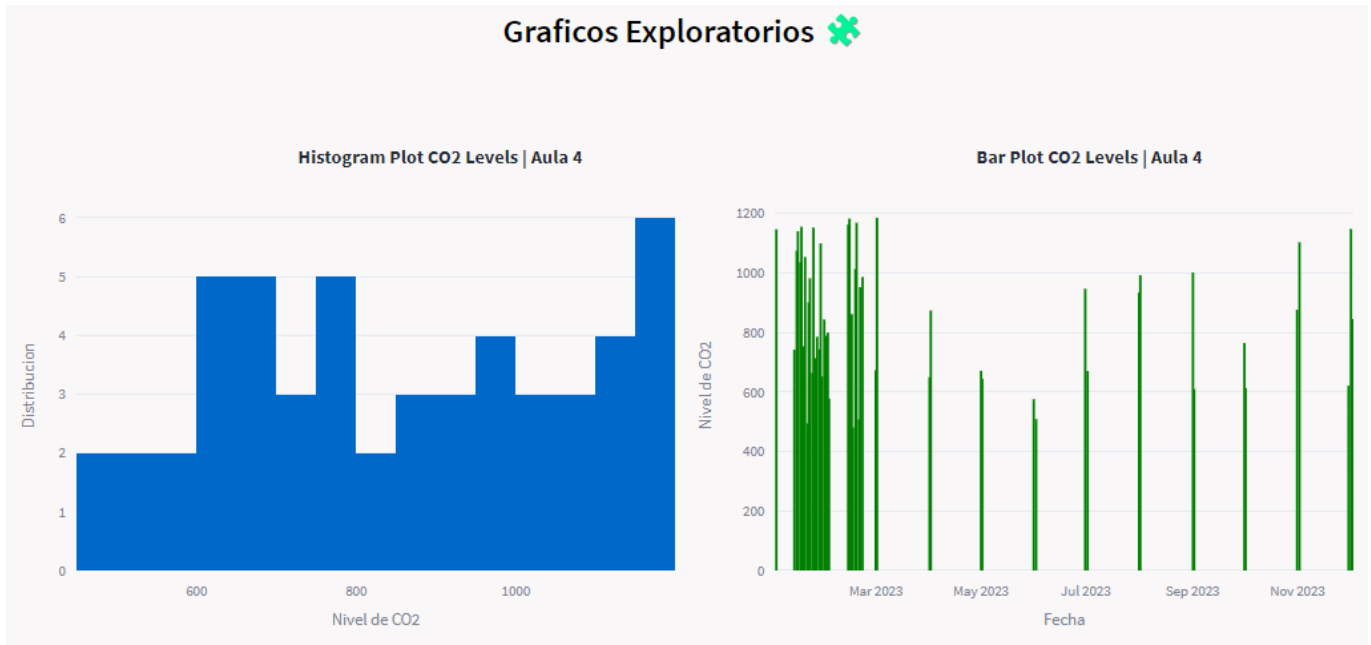


Figura 26

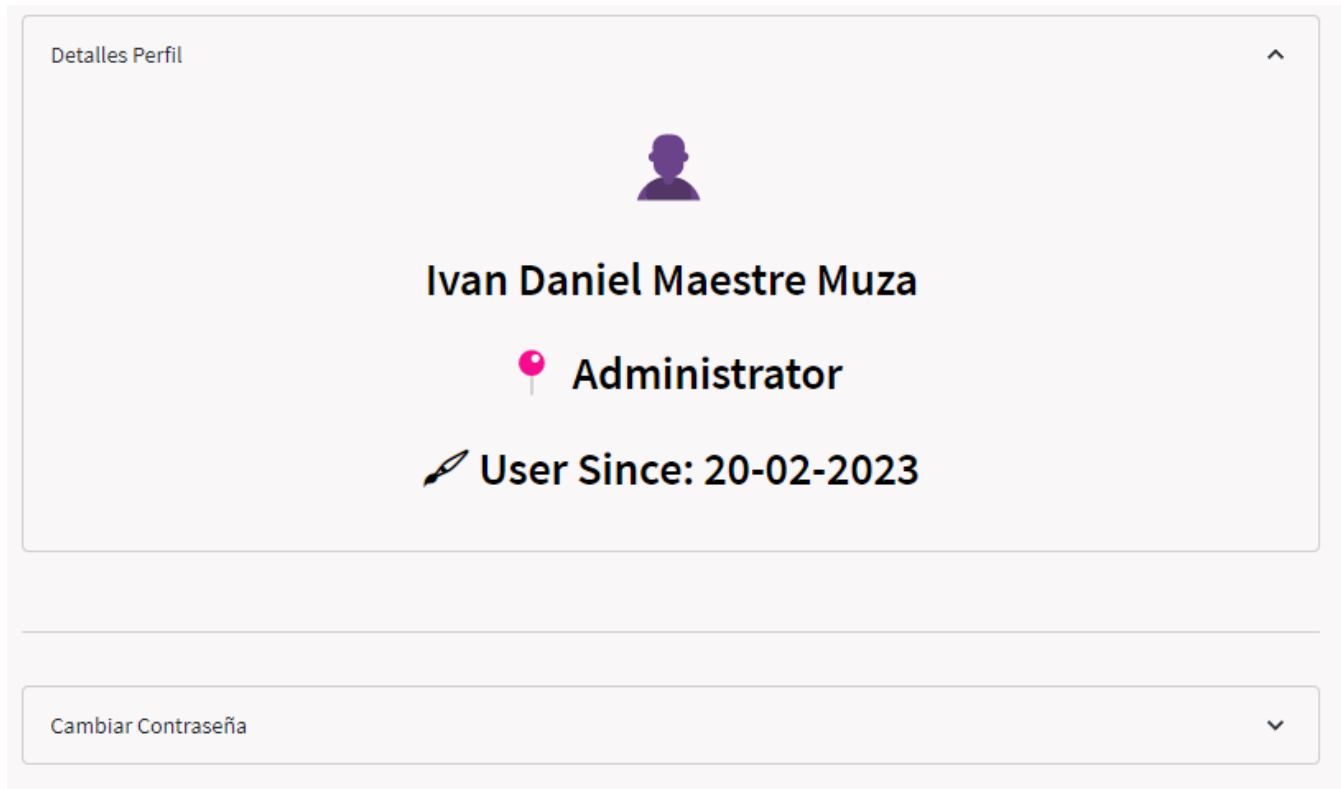
Prototipo - Gestión de Usuarios

CO2 Control

Ivan Daniel Administrator

Aulas **Gestion de Usuarios** Perfil Bases de Datos Cerrar Sesion

- Agregar Usuario
- Borrar Usuario
- Modificar Usuario

Figura 27*Prototipo – Perfil***4.5 Validación de la solución**

Al momento de implementar una solución tecnológica donde se articulan diferentes componentes es necesario generar estrategias de control y seguimiento con el objetivo de validar que todo el flujo de la herramienta esté generando las salidas esperadas en un tiempo determinado.

Para este caso de estudio, nos aproximamos a la evaluación de los datos por medio de una validación cruzada (cross-validation), una técnica utilizada en el aprendizaje automático para evaluar el rendimiento de un modelo de predicción. El objetivo principal de la técnica es estimar el comportamiento de un modelo de machine learning para mejorar la evaluación de su rendimiento.

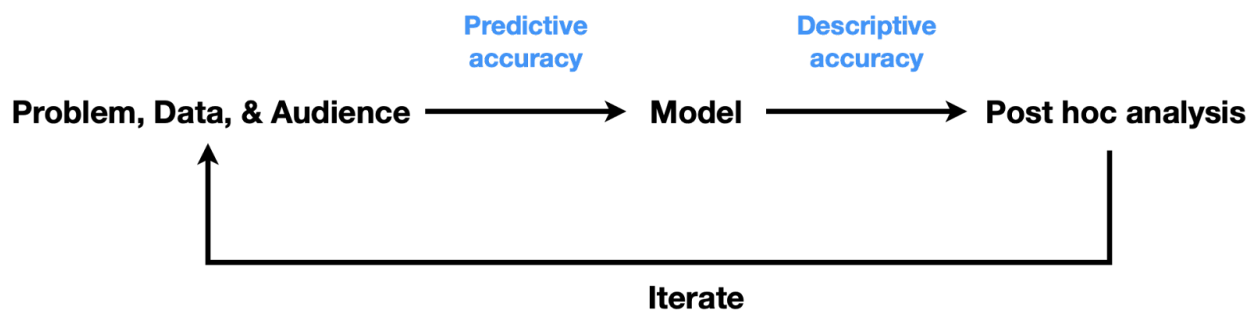
El método de cross-validation implica dividir el conjunto de datos en dos grupos: el conjunto de entrenamiento y el conjunto de prueba. El conjunto de entrenamiento se utiliza para ajustar el modelo, mientras que el conjunto de prueba se utiliza para evaluar el rendimiento del modelo.

La tecnología IoT complementa la técnica de cross-validation en el aprendizaje automático, ya que los sensores y dispositivos conectados pueden recopilar datos en tiempo real y de manera continua para evaluar el rendimiento del modelo. Esto permite una mayor precisión en la evaluación del modelo, ya que los datos de prueba se actualizan constantemente y el modelo se puede ajustar en consecuencia para mejorar su rendimiento en prácticas reales.

Los modelos de aprendizaje automático han demostrado un gran éxito en el aprendizaje de patrones complejos que permiten hacer predicciones sobre datos no observados (Murdoch, Singh, Karl, Abbasi-Asl, & Yu, 2019).

Figura 28

Diagrama de flujo - Modelo de aprendizaje automático



La figura 28 presenta una descripción general de este proceso, que pretende captar la mayoría de los problemas de la ciencia de datos.

Lo que generalmente se denomina interpretación se produce en gran medida en las fases de modelización y análisis post hoc, en las que el problema, los datos y el público proporcionan el contexto necesario para elegir los métodos adecuados (Murdoch, Singh, Karl, Abbasi-Asl, & Yu, 2019).

Por lo tanto, haciendo uso de los datos históricos, se desarrollará un modelo de aprendizaje automático basado en series temporales, cuyo objetivo será comparar la muestra tomada por el sensor, y el valor que el modelo espera que tenga dicha toma.

Una serie temporal es un conjunto de datos bien definidos recogidos en puntos sucesivos a intervalos de tiempo uniformes. El análisis de series temporales es una parte importante de la estadística, que recoge conjuntos de datos para estudiar sus características y ayuda a predecir valores futuros de las series basándose en dichas características (Mondal, Shit, & Goswami, 2014).

Para ello, es necesario contar con un proceso largo de obtención de datos, en donde se almacenará la mayor cantidad de información posible. Con estos datos históricos, nos aproximamos a una predicción utilizando AutoRegressive Integrated Moving Average (ARIMA), una de las técnicas de predicción de series temporales usada en algunos de los proyectos más relevantes en machine learning como, por ejemplo, obtener la predicción de la demanda de electricidad de Reino Unido [35], analizar y predecir las tasas de cambio de las monedas de diferentes países [36], o también, predecir el número de pasajeros que llegarían a un aeropuerto [37].

AutoRegressiveMoving Average (ARMA) es un método importante para estudiar series temporales. El concepto de modelos autorregresivos (AR) y de media móvil (MA) fue formulado por los trabajos de Yule, Slutsky, Walker y Yaglom (Chen, Lan, Hu, Liu, & Deng, 2014). La media

móvil integrada autorregresiva (ARIMA) se basa en el modelo ARMA. La diferencia es que el modelo ARIMA convierte los datos no estacionarios en estacionarios antes de trabajar con ellos. El modelo ARIMA se utiliza ampliamente para predecir datos de series temporales lineales (Box & Tiao, 1975).

En la actualidad, existen varios modelos de predicción de series temporales que pueden ser utilizados para diferentes tipos de proyectos, estos son algunos ejemplos:

- **Redes neuronales recurrentes (RNN):** Un modelo útil para la predicción de series temporales debido a su capacidad para recordar información histórica de la secuencia y utilizarla para predecir valores futuros [38].
- **Modelos de suavizado exponencial (ES):** Se basa en la idea de que los datos históricos deben recibir diferentes pesos según su antigüedad. Es adecuado para series temporales con tendencias, estacionalidad y ruido aleatorio [39].
- **Vector Autoregression (VAR):** Este modelo es relevante cuando hay más de una variable que se analiza para predecir las series temporales (multivariadas). La técnica VAR utiliza la información histórica de todas las variables para realizar la predicción [40].
- **Prophet:** Es una técnica desarrollada por Facebook que utiliza un modelo aditivo que incluye componentes estacionales y de tendencia. Puede manejar cambios abruptos y ausencias de datos [41].

El modelo ARIMA es el seleccionado para este proyecto de investigación debido a que es ideal para trabajar con datos estacionarios, además de poder modelar y predecir series temporales con tendencias claras. Además, esta técnica es capaz de manejar datos con rango específico de valores, lo que lo hace adecuado para el proyecto. Otra de las ventajas de este modelo es la facilidad

para interpretar e implementar, lo que resulta muy útil a la hora de requerir cambios en la configuración del modelo o realizar ajustes sobre la marcha.

La fórmula principal del modelo ARIMA se puede escribir de la siguiente manera:

$$\text{ARIMA}(p, d, q) = \text{AR}(p) + \text{I}(d) + \text{MA}(q)$$

Donde:

- p es el número de términos autorregresivos (AR)
- d es el número de diferenciaciones no estacionales (I)
- q es el número de términos de media móvil (MA)

Los términos AR representan la correlación entre una observación y sus p observaciones anteriores, los términos MA representan la correlación entre una observación y los errores de predicción de las q observaciones anteriores, y los términos I representan la diferenciación para eliminar la tendencia no estacionaria en los datos. La combinación de estos términos permite modelar una amplia variedad de series temporales.

La fórmula anteriormente descrita combina tres elementos:

- Un modelo autorregresivo (AR) que utiliza los valores pasados de la serie temporal para predecir los valores futuros.
- Un modelo de media móvil (MA) que utiliza los errores pasados de la serie temporal para predecir los datos posteriores.
- Un modelo integrado (I) que se utiliza para transformar una serie temporal no estacionaria en una serie temporal estacionaria, lo que facilita la predicción.

Con esta técnica, podremos tener una predicción basada en el comportamiento de las mediciones anteriores, logrando así, no solo la evaluación del prototipo, sino también, en la detección de anomalías en la detección de muestras del prototipo.

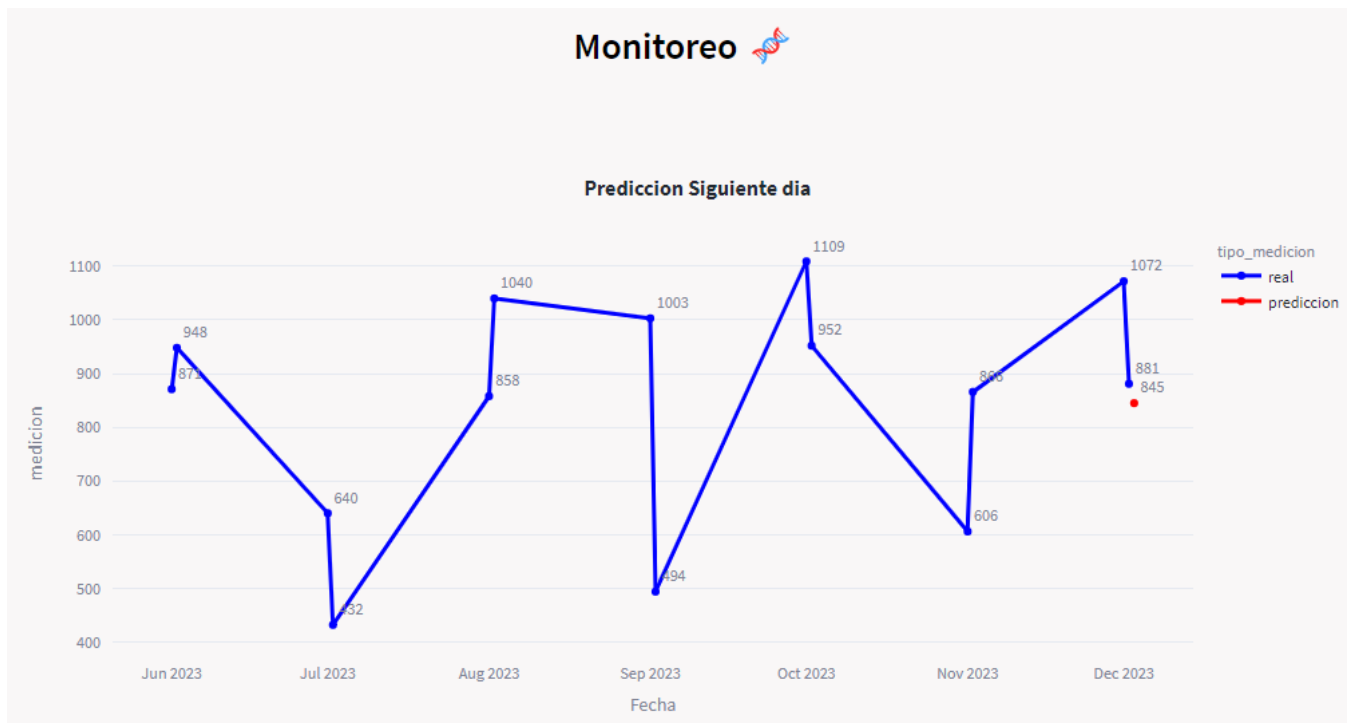
4.5.1 Resultados de la validación

Para realizar la evaluación del prototipo nos apoyamos de los resultados obtenidos de la evaluación cruzada de los datos con el modelo de machine learning implementado. Es importante aclarar que al tratarse con datos simulados aleatoriamente pueden surgir diferencias importantes en las predicciones encontradas. Este caso de estudio sirve de apoyo para que en un futuro se implemente con datos de sensores reales.

A continuación, presentamos una gráfica donde visualizamos un ejemplo de predicción:

Figura 29

Prototipo – Monitoreo



Al pasar el cursor por los puntos de la gráfica obtenemos con precisión la medida tomada en ese momento. Para este caso, la comparación fue con 1 día diferencia:

Figura 30

Dato real

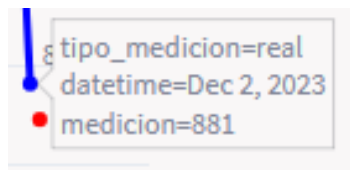
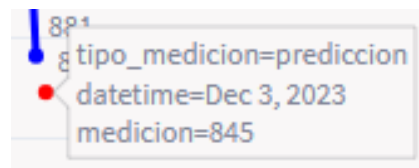


Figura 31

Dato de predicción



Gracias a este modelo implementado, podemos monitorear de una manera más completa los niveles de CO2 en las aulas de la universidad. Dependiendo del caso a tratar, se puede modificar el rango de fechas o tiempo y el tipo de medición requerido (que en este caso fue el nivel de CO2).

5 Trabajo a futuro

- Implementar la herramienta desarrollada en este proyecto para utilizarse en espacios Smart Campus de manera general con el fin de realizar mediciones de CO2 en las aulas de la Universidad Industrial de Santander.
- Realizar la revisión y toma de decisiones acertadas sobre los niveles de CO2 encontrados en dichas aulas con vista a mejorar la ventilación de los espacios de la comunidad universitaria. Esto debe estar a cargo de personal especializado.

6 Conclusiones

- La tecnología IoT permite el desarrollo de soluciones que busquen medir la calidad del aire en espacios cerrados como el aula de clase, una solución adecuada debe basarse en una plataforma software que provea las capacidades de transmisión, almacenamiento, visualización y consulta de datos de una forma adecuada.
- Se demostró que diseñar requerimientos funcionales, no funcionales y definir la arquitectura implementada permitió conocer a grandes rasgos el alcance, usabilidad y posterior construcción del aplicativo web.
- Se utilizó un modelo de machine learning para evaluar el prototipo realizado, el cual nos permitió tener una predicción de valores de CO2 a partir de los datos obtenidos anteriormente. Esta validación nos ayuda a tomar medidas preventivas y proactivas para mejorar la concentración de CO2 en el aire en las aulas de la universidad.
- Cabe destacar que la implementación de sensores de CO2 reales en esta herramienta IoT podría mejorar aún más la precisión y efectividad de las lecturas. Además de poder realizar un monitoreo de los niveles de CO2 en tiempo real que brinde información rápida y eficaz para modificar las condiciones del espacio en los momentos que se necesite.
- La integración de la herramienta IoT entregada con el modelo de machine learning ARIMA puede ser utilizada como base para el desarrollo de futuros proyectos sobre control y monitoreo de datos que se quieran medir como lo son: la calidad del aire, temperatura, agua, energía, entre otros.

En resumen, la implementación de una herramienta basada en IoT para el control y monitoreo del nivel de CO2 en un Smart Campus, junto con un modelo de machine learning para predecir el siguiente nivel de CO2 esperado, resultó efectiva para medir la calidad del aire y tomar

medidas preventivas. La implementación futura de sensores de CO2 reales podría mejorar aún más la precisión de la herramienta. La aplicación de esta tecnología podría tener un impacto muy positivo en la salud y el bienestar de las personas que ocupan espacios cerrados.

7 Referencias Bibliográficas

- M. Zafra and J. Salas. (2021, Mar 28). No respire el aire de otro: cómo esquivar el coronavirus en interiores [Online]. Available: <https://elpais.com/ciencia/2021-03-28/no-respires-el-aire-de-otro-como-esquivar-el-coronavirus-en-interiores.html>
- M. Zafra and J. Salas. (2020, Oct 28). Un salón, un bar y una clase: así contagia el coronavirus en el aire [Online]. Available: <https://elpais.com/ciencia/2020-10-24/un-salon-un-bar-y-una-clase-asi-contagia-el-coronavirus-en-el-aire.html>
- M. Zafra and J. Salas. (2020, Jun 8). Radiografía de tres brotes: así se contagiaron y así podemos evitarlo [Online]. Available: <https://elpais.com/ciencia/2020-06-06/radiografia-de-tres-brotes-asi-se-contagiaron-y-asi-podemos-evitarlo.html#?rel=mas>
- (2021). ¿Qué es el Internet de las cosas (IoT)? [Online]. Available: <https://www.redhat.com/es/topics/internet-of-things/what-is-iot>
- N. Martínez. (2017, Nov 22). Smart Campus, construyendo universidades más inteligentes [Online]. Available: <https://www.nobbot.com/futuro/smart-campus-construyendo-universidades-mas-inteligentes/>
- (2021). Mediciones de Dióxido de Carbono CO2 [Online]. Available: <https://www.vaisala.com/es/measurement/carbon-dioxide-co2-measurements>

(2020, Oct 19). El aire en espacios cerrados y el coronavirus (COVID-19) [Online]. Available:
<https://espanol.epa.gov/cai/el-aire-en-espacios-cerrados-y-el-coronavirus-covid-19>

(2021, Jul 14). La Importancia De Un Medidor De CO2 Para La Prevención En El Contagio Del Coronavirus [Online]. Available: <https://sensorgo.mx/medidor-de-co2/>

(2021). Puerta de enlace [Online]. Available: https://es.wikipedia.org/wiki/Puerta_de_enlace

G. Maloy. (2020, Mar 9). ¿Qué es un Sensor y Qué Hace? [Online]. Available:
<https://dewesoft.com/es/daq/que-es-un-sensor>

(2021). Interfaz [Online]. Available: <https://es.wikipedia.org/wiki/Interfaz>

M. J. Lamarca. (2018, Jul 29). Extensibilidad [Online]. Available:
<http://www.hipertexto.info/documentos/extensib.htm>

(2021). Sensor CO2 [Online]. Available:
https://www.amazon.com/s?k=Sensor+co2&__mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&ref=nb_sb_noss_2

(2020, Ene 8). Creating Smarter Schools: Benefits and Applications of IoT in Education [Online]. Available: <https://www.igor-tech.com/news-and-insights/articles/creating-smarter-schools-benefits-and-applications-of-iot-in-education>

D. Camacho and G. Pedraza, "DISEÑO DE UNA APLICACION WEB EXTENSIBLE PARA LA ADMINISTRACION DE UNA PLATAFORMA IOT DISEÑADA PARA SMART CAMPUS", Trabajo de grado, Santander, Universidad Industrial de Santander, Bucaramanga, 2019.

C. Gutiérrez and G. Pedraza, "DISEÑO DE UN FRAMEWORK SOFTWARE EXTENSIBLE PARA DISPOSITIVOS TIPO GATEWAY INTEGRADOS EN PLATAFORMAS IOT PARA SMART CAMPUS", Trabajo de grado, Santander, Universidad Industrial de Santander, Bucaramanga, 2019.

K. Arias, J. Estupiñan and G. Pedraza, "DISEÑO DEL COMPONENTE SOFTWARE BACKEND ORIENTADO A UNA PLATAFORMA IOT DISEÑADA PARA SMART CAMPUS", Trabajo de grado, Santander, Universidad Industrial de Santander, Bucaramanga, 2019.

(2022). MH-Z14A NDIR CO2 SENSOR FOR CARBON DIOXIDE DETECTION [Online]. Available: <https://www.winsen-sensor.com/sensors/co2-sensor/mh-z14a.html>

(2023, Abr 15). Nuevo Sensor de CO2 MH-Z14A 0~5000ppm ORIGINAL NUEVO [Online]. Available: <https://www.ebay.com/itm/152259438179>

(2020). Monitor de aire de CO2 [Online]. Available: https://obw.ie/our-products/co2-air-monitor/?utm_term=co2%20gas%20detector

¿Qué es MQTT? [Online]. Available: <https://aws.amazon.com/es/what-is/mqtt/>

MQTT protocol [Online]. Available: <https://www.javatpoint.com/mqtt-protocol>

Shifan Yu (2020, Nov 5) Connect to MQTT broker with WebSocket [Online]. Available: <https://www.emqx.com/en/blog/connect-to-mqtt-broker-with-websocket>

Box, G., & Tiao, G. (1975). Intervention Analysis with Applications to Economic and Environmental Problems. 70-79.

Chen, S., Lan, X., Hu, Y., Liu, Q., & Deng, Y. (2014). The time series forecasting: from the aspect of network [versión PDF]. doi:<https://doi.org/10.48550/arXiv.1403.1713>

Mondal, P., Shit, L., & Goswami, S. (2014). Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices [versión PDF]. Obtenido de <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9ec1f14cb7d48ca50a18c66e69de5865c55686dd>

G. E. P. Box & G. C. Tiao (1975) Intervention Analysis with Applications to Economic and Environmental Problems, *Journal of the American Statistical Association*, 70:349, 70-79, DOI: 10.1080/01621459.1975.10480264

(2020, Sep 22) Breve historia de Internet de las cosas (IoT) [Online]. Available:

<https://empresas.blogthinkbig.com/breve-historia-de-internet-de-las-cosas->

[iot/#:~:text=El%20m%C3%A1s%20antiguo%20del%20que,de%20un%20enlace%20de%20radio.](https://empresas.blogthinkbig.com/breve-historia-de-internet-de-las-cosas-iot/#:~:text=El%20m%C3%A1s%20antiguo%20del%20que,de%20un%20enlace%20de%20radio.)

(2023) MH-410D NDIR CO2 SENSOR [Online]. Available: [https://www.winsen-](https://www.winsen-sensor.com/sensors/co2-sensor/mh-410d.html)

[sensor.com/sensors/co2-sensor/mh-410d.html](https://www.winsen-sensor.com/sensors/co2-sensor/mh-410d.html)

(2023) SENSOR DE CO2 MH-410D NDIR PARA DETECCIÓN DE DIÓXIDO DE CARBONO

[Online]. Available: <https://www.ebay.com/itm/143404546161>

(2023) MH-Z19B ndir CO2 sensor for indoor air quality monitoring [Online]. Available:

<https://www.winsen-sensor.com/sensors/co2-sensor/mh-z19b.html>

(2018, Dic 7) MH-Z19 NDIR - Módulo de sensor de CO2 por infrarrojos (0 a 5000 ppm, para monitores de CO2, sensor de dióxido de carbono, MH-Z19B, con líneas) [Online].

Available: <https://www.amazon.com/-/es/MH-Z19-NDIR-infrarrojos-monitores-di%C3%B3xido/dp/B07L6QZDK7>

(2021) MH-Z16 NDIR CO2 SENSOR [Online]. Available: [https://www.winsen-](https://www.winsen-sensor.com/sensors/co2-sensor/mh-z16.html)

[sensor.com/sensors/co2-sensor/mh-z16.html](https://www.winsen-sensor.com/sensors/co2-sensor/mh-z16.html)

(2023) MH-Z16 CO2 Sensor Gas Sensor Module Carbon Dioxide Detector 0~5%VOL 0-5000ppm

[Online]. Available: <https://www.elecbee.com/en-25435-MH-Z16-CO2-Sensor-Gas-Sensor-Module-Carbon-Dioxide-Detector-0-5-VOL-0-5000ppm>

C. Nichiforov, I. Stamatescu, I. Făgărășan and G. Stamatescu, "Energy consumption forecasting using ARIMA and neural network models," 2017 5th International Symposium on Electrical and Electronics Engineering (ISEEE), Galati, Romania, 2017, pp. 1-4, doi: 10.1109/ISEEE.2017.8170657

D. S. Dev, A. Ray and J. Austin, "ARIMA Model based Time Series Modelling and Prediction of Foreign Exchange Rate against US Dollar," 2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC), Bhubaneswar, India, 2022, pp. 1-7, doi: 10.1109/ASSIC55218.2022.10088356

Ferhatosmanoglu, N., & Macit, B. (2016). Incorporating Explanatory Effects of Neighbour Airports in Forecasting Models for Airline Passenger Volumes. In ICORES (pp. 178-185)

Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618. Genetic Programming and Evolvable Machines, 19(1-2), 305-307.

Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.

Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. Springer Science & Business Media.

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37-45