



UNIVERSIDAD EAN

PROYECTO DE GRADO

**MODELO DE AUTOMATIZACIÓN PARA CONTROL DE PRECIOS
DINÁMICOS EN ALIMENTOS PERECEDEROS**

**Alex Buitrago Basallo – Ingeniería de Sistemas
David Esteban Rodríguez Rojas- Ingeniería Mecatrónica**

Johanna Karina Solano - directora de tesis

BOGOTÁ D.C



Resumen

Este proyecto tiene como objetivo desarrollar un sistema de control dinámico de precios para almacenes de mercado, esto mediante la implementación de sensores físicos y algoritmos predictivos. A lo largo del documento se identifica la problemática del desperdicio alimentario y la inseguridad alimentaria en Colombia, subrayando la necesidad urgente de mejorar la gestión de alimentos perecederos. Utilizando inteligencia artificial, el sistema pretende mejorar la eficiencia operativa y reducir las pérdidas económicas, promoviendo una distribución más justa y asequible de los alimentos. El marco teórico del proyecto incluye aspectos clave sobre el funcionamiento de los lenguajes de programación, automatización, aprendizaje automático, y modelos predictivos, así como el estudio de la degradación de productos orgánicos. Con un diseño metodológico centrado en la recolección de datos sobre variables de precios y fisicoquímicas de productos perecederos, el sistema crea un modelo dinámico de precios que responde a las condiciones del inventario. La alternativa de solución propone un algoritmo para ajustar los precios en función del estado de los productos y en el análisis de costos en el mercado colombiano, para esto se considera la inversión en sensores, infraestructura y desarrollo del algoritmo. Finalmente, el plan de implementación prevé una fase piloto para evaluar la efectividad del sistema ajustando las variables en tiempo real.

Contenido

1. Introducción.....	3
----------------------	---



2. Objetivos.....	4
Objetivo general.....	4
Objetivos específicos.....	5
3. Definición del problema	5
4. Pregunta problema	7
5. Justificación.....	7
6. Análisis de Requerimientos	9
Requerimientos funcionales.....	9
Requerimientos no funcionales	10
7. Estado del arte	11
Modelos de gestión en la actualidad.....	11
Automatización	12
8. Marco Teórico.....	13
Lenguajes de programación	13
Bases de datos	15
Aprendizaje automático.....	15
Modelos predictivos.....	17
Degradación de productos orgánicos.....	18
9. Análisis de restricciones	20
10. Metodología y desarrollo central	22
Instrumentos para la recolección de información	22
Selección de Equipos y Dispositivos de Medición	25
Fuentes de Datos Externos para el Modelo Predictivo	33
Procesamiento y programación del Modelo Predictivo	37
Prueba del sistema en entorno controlado	45
11. Análisis de costos	50
12. Conclusiones.....	53
13. Bibliografía.....	55

1. Introducción

El desperdicio de alimentos constituye un problema global con profundas implicaciones



económicas, sociales y ambientales. En Colombia, este fenómeno exacerba la inseguridad alimentaria y afecta la economía nacional. Según un estudio del Departamento Nacional de Planeación (DNP), aproximadamente un tercio de los alimentos producidos en el país se pierde o desperdicia, lo que genera un impacto significativo en la seguridad alimentaria y en la economía (Gaviria, 2016).

Esta situación es paradójica, ya que mientras se desperdician grandes cantidades de alimentos, una porción considerable de la población colombiana carece de acceso a una alimentación adecuada (Jiménez, 2022). El desperdicio no solo representa una pérdida de recursos naturales, sino que también incrementa los precios de los alimentos, dificultando el acceso de los hogares más vulnerables.

Ante esta problemática, se propone implementar un sistema dinámico de gestión de inventarios en almacenes minoristas. Este sistema, basado en sensores, monitoreará en tiempo real el estado de los productos frescos y ajustará los precios de manera estratégica. De esta forma, se reducirá el desperdicio al minimizar las pérdidas por deterioro y se facilitará el acceso a alimentos frescos a precios más asequibles.

Al reducir el desperdicio y mejorar la eficiencia en la cadena de suministro, este proyecto contribuirá a la generación de precios más justos, aumentar la disponibilidad de productos frescos y mejorar la seguridad alimentaria de la población. Además, se alinea con los Objetivos de Desarrollo Sostenible, promoviendo una economía circular y sistemas alimentarios más sostenibles. La implementación de este proyecto se basa en datos reales y en el contexto específico de Colombia, donde el desperdicio de alimentos y la inseguridad alimentaria son desafíos urgentes. Al abordar estas problemáticas de manera integral, se busca construir un futuro más sostenible y equitativo para todos.

2. Objetivos

Objetivo general

Desarrollar un sistema de control de precios dinámico en tiempo real para superficies comerciales



mediante la implementación de sensores físicos y algoritmos predictivos.

Objetivos específicos

1. Analizar los datos de tendencias actuales en la gestión de precios de alimentos e inventarios en Colombia, evaluando su aplicabilidad en el desarrollo de métodos predictivos basados en tecnologías emergentes.
2. Desarrollar un algoritmo predictivo que ajuste dinámicamente los precios de los alimentos perecederos, utilizando datos provenientes del sistema de monitoreo, patrones de consumo y factores para la rotación de inventario.
3. Diseñar un prototipo de sistema de monitoreo en tiempo real basado en sensores físicos para determinar el estado de los alimentos perecederos en mercados minoristas, detectando cambios en sus condiciones.

3. Definición del problema

La inseguridad alimentaria es un problema global que afecta de manera particular a los países en vía de desarrollo como Colombia. Según el Departamento Administrativo Nacional de Estadística (DANE), en 2022 la inseguridad alimentaria moderada o grave afectó al 28,1 % de los hogares colombianos, lo que implica que 28 de cada 100 hogares experimentaron dificultades para acceder a



alimentos de calidad.

Además, cifras del Departamento Nacional de Planeación (DNP) muestran que, en promedio, por cada 3 toneladas de alimentos producidos se pierde o se desperdicia una tonelada. Así mismo, en un estudio realizado sobre la cadena de consumo en Colombia, se determinó que anualmente el país cuenta con una oferta potencial de alimentos que supera los 28 millones de toneladas, por lo que se estaría hablando de pérdidas cercanas a las 9.24 toneladas. Si bien estos desperdicios surgen durante distintas etapas de la cadena de suministro, el 30.4% de los desechos se generan en el almacenamiento y distribución, ya sea por malas prácticas de conservación, gestión inadecuada del inventario o condiciones ambientales (Gaviria S, 2016).

Uno de los factores más importantes es el control ambiental deficiente. Los alimentos, especialmente perecederos como frutas y verduras, requieren condiciones específicas de temperatura, humedad, y ventilación para mantenerse en buen estado. Sin embargo, sin un control adecuado de factores ambientales, los alimentos pueden pasar de un estado óptimo a uno en riesgo de deterioro sin que los operadores lo perciban a tiempo.

A esto se le agrega que muchos almacenes y puntos de distribución no se rigen por prácticas de inventariado eficientes, lo que significa que los productos más antiguos no se despachan primero, permitiendo que los alimentos más nuevos ocupen su lugar en las estanterías, lo que lleva al vencimiento y deterioro de los productos almacenados por más tiempo.

La logística ineficiente también juega un papel importante. Los retrasos en la distribución pueden hacer que los alimentos pasen más tiempo del necesario en los almacenes, reduciendo el tiempo de vida útil. Los alimentos, en principio en excelentes condiciones, se convierten en millones de toneladas desperdiciadas, mientras que el 54.2% de los hogares en Colombia corren riesgo de inseguridad alimentaria (Jiménez M, 2022) lo que aumenta la inequidad en el acceso a alimentos.

Al abordar la gestión de productos perecederos, reducir las pérdidas económicas y adoptar tecnologías



avanzadas, surge la cuestión de cómo podría integrarse sistemas de control en los comercios para reducir el desperdicio de alimentos perecederos, a través de la modificación dinámica de los precios. Esta implementación permitiría ajustar los precios en función de factores como la frescura y la demanda, optimizando los recursos y disminuyendo pérdidas para los comercios, mientras se contribuye a un modelo de consumo más eficiente y sostenible.

4. Pregunta problema

¿De qué manera puede integrarse sistemas de control en los comercios para reducir el desperdicio de alimentos perecederos, a través de la modificación de precios de acuerdo con la oferta y demanda?

5. Justificación

En Colombia, el desperdicio de alimentos agrava la inseguridad alimentaria y genera pérdidas económicas significativas. Según el Departamento Nacional de Planeación (2016), en 2015 se desperdiciaron 90.000 toneladas de alimentos, equivalentes a \$204 mil millones de pesos. Con el crecimiento poblacional y la inflación elevando los precios, optimizar la gestión de alimentos es cada vez más urgente.

Ante esta problemática, la implementación de tecnologías avanzadas como inteligencia artificial (IA) y herramientas de ciencia de datos se presenta como una solución innovadora y efectiva para mitigar el



desperdicio de alimentos. Estas tecnologías permiten prever la demanda de productos y ajustar los precios de manera con base en datos, optimizando la cadena de suministro y reduciendo las pérdidas. Por ejemplo, mediante la predicción de la demanda, los comercios pueden ajustar sus compras y reducir el riesgo de excedentes que acaban siendo desperdiciados. Adicionalmente, la implementación de precios dinámicos basados en factores como la cercanía de los productos a su fecha de vencimiento y las fluctuaciones en la demanda permite que los comercios ofrezcan descuentos estratégicos, incentivando el consumo oportuno de productos antes de que estos se deterioren.

La eficiencia de estas soluciones tecnológicas ya ha sido probada en otros sectores. Un estudio realizado por McKinsey & Company (2018) reveló que el uso de IA en la optimización de la cadena de suministro en el sector retail puede reducir hasta un 50 % el desperdicio de productos perecederos. Además, las herramientas de análisis predictivo permiten a las empresas ajustar dinámicamente la oferta y demanda, mejorando la precisión en la toma de decisiones y reduciendo tanto los costos operativos como las pérdidas económicas.

El proyecto propone un sistema de aprendizaje autónomo para gestionar la cantidad de alimentos que los comercios deben comprar, representa una solución clave para reducir el desperdicio alimentario en mercados y tiendas. Al combinar el análisis predictivo con la posibilidad de ajustar dinámicamente los precios en función de la oferta y la demanda, se espera no solo reducir el desperdicio, sino también aumentar el acceso a alimentos de calidad a precios más asequibles. Esto tiene el potencial de mejorar significativamente la economía distrital, al disminuir las pérdidas para los negocios locales, y, a largo plazo, contribuir a reducir la brecha de inseguridad alimentaria.

En adición, este proyecto responde a la creciente demanda del manejo sostenible de los recursos. Durante el Acuerdo de París (2015), los países se comprometieron a reducir sus emisiones de gases de efecto invernadero, y según informes de este el sector alimentario es uno de los principales contribuyentes a estas emisiones debido al desperdicio de alimentos. La implementación de tecnologías que permitan una gestión más eficiente de los alimentos ayudará a reducir el impacto ambiental, alineándose con los objetivos de sostenibilidad globales y mejorando la competitividad económica de los comercios locales al



reducir sus costos operativos.

Siendo así, el desarrollo de este sistema de aprendizaje automático es una iniciativa sólida y necesaria.

Este enfoque no solo aborda el desperdicio alimentario desde una perspectiva económica y medioambiental, pues, al aprovechar el potencial de la inteligencia artificial, el proyecto tiene el potencial de generar un impacto significativo en el modelo de consumo actual.

6. Análisis de Requerimientos

Requerimientos funcionales

Infraestructura informática

Un computador con suficiente capacidad de procesamiento, almacenamiento escalable para los volúmenes de datos registrados por los sensores y baja latencia en su conexión será imprescindible para ejecutar algoritmos predictivos y controlar los dispositivos.

Sensores

Para la implementación del sistema se propone montar sensores físicos de alta precisión, tales como el sensor de temperatura y humedad DHT22, sensores de gas MQ-135 para la detección de gases como amoníaco y CO₂, y sensores de etileno como el MQ-3 para los lixiviados. Estos sensores deben estar



integrados a una red de monitoreo que envíe datos a un controlador que se encargue de procesar la información antes de llegar al servidor central.

Entrenamiento y modelado

Para el desarrollo del algoritmo predictivo, se utilizará un entorno de programación robusto como Python, junto con bibliotecas como TensorFlow o Scikit-learn especializadas en la creación y entrenamiento de modelos de machine learning. Además, el manejo de bases de datos puede realizarse a través de MySQL. El proyecto busca formular pruebas piloto en un entorno controlado, por lo que se empleará una instalación de prueba simulada equipada con muestras de productos orgánicos, actuadores y servidor para evaluar la efectividad del sistema para su posterior escalabilidad.

Integración con la electrónica

El sensor de temperatura y humedad se conecta a un pin digital para que el controlador reciba los datos mientras que los sensores MQ-135 y MQ-3 se conectan a pines analógicos para leer concentraciones de gases. El controlador, para este caso un ESP32, está configurado para leer estos datos a intervalos regulares mediante librerías específicas, este o cualquier controlador debe contar con conexiones GPIO para conectar múltiples sensores. Luego debe de enviar la información a un servidor central a través de una conexión Wi-Fi usando protocolos como MQTT o HTTP.

Interfaz de Usuario (UI) para los Comerciantes

Se debe desarrollar una interfaz simple, intuitiva y accesible, diseñada específicamente para comerciantes que busquen una manera eficiente de gestionar el inventario de alimentos perecederos. A través de un panel de control interactivo, los usuarios podrán visualizar en tiempo real el estado de los productos

Requerimientos no funcionales

Tiempo de Respuesta

El sistema debe procesar y enviar datos de los sensores al servidor central en tiempo real. Idealmente, las lecturas deben actualizarse cada hora para permitir una gestión eficiente del inventario.

Facilidad de uso

La visualización de los datos debe ser intuitiva y fácil de usar, permitiendo que los usuarios interpretar



rápidamente la información sobre el estado de los alimentos y tomar decisiones informadas.

Adaptabilidad

Cada alimento tiene condiciones de degradación diferentes, sus cambios a nivel físico y químico difieren bastante, por lo que el sistema debe funcionar como base para ser reentrenado y adaptado a diferentes productos.

7. Estado del arte

Modelos de gestión en la actualidad

Sistemas POS

Sistemas de punto de venta, son herramientas tecnológicas utilizadas por los comercios para procesar transacciones de ventas, gestionar inventarios, y ofrecer servicios a los clientes. Estos sistemas han evolucionado desde simples registradoras hasta complejos sistemas integrados que no solo registran ventas, sino también rastrean datos de inventarios, generan informes y permiten el análisis de ventas y comportamiento de los consumidores. Los sistemas POS se componen generalmente de un hardware (terminal de ventas, escáner de códigos de barras, impresora de recibos) y un software que facilita la automatización de las transacciones comerciales. (Pantano, E., Timmermans, H., & Clodfelter, R., 2020)

Uno de los aspectos más importantes de los sistemas POS es su capacidad para integrar funciones de gestión de inventarios. A medida que los productos se escanean y se venden, el sistema actualiza



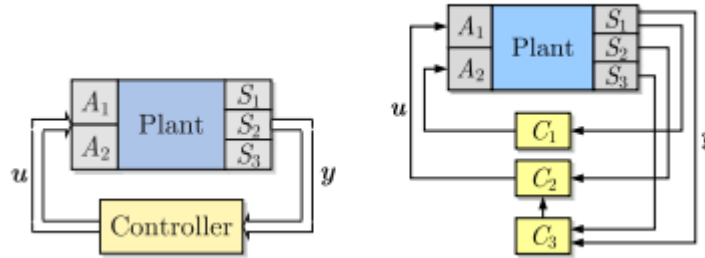
automáticamente el inventario, lo que permite una gestión más precisa y eficiente del stock. Esto es particularmente crucial en sectores como el de los supermercados, donde la variedad de productos y su rápida rotación demandan una gestión efectiva de inventarios (Laudon & Laudon, 2016).

En los supermercados, la gestión de inventarios es crítica debido a la rápida rotación de productos, especialmente los perecederos. Un sistema de inventario eficiente no solo reduce el desperdicio de productos, sino que también mejora la rentabilidad del supermercado al asegurarse de que siempre haya suficiente stock disponible para satisfacer la demanda de los clientes. Esto incluye características como la capacidad de rastrear el ciclo de vida de los productos, gestionar las órdenes de reabastecimiento y realizar ajustes en el stock en tiempo real (Render, Stair, & Hanna, 2017).

Automatización

Con el tiempo, las tareas humanas se han vuelto cada vez más complejas y sofisticadas, el mercado es muy competitivo y es fundamental aprovechar la mayor cantidad de recursos. La automatización nace como una forma de optimizar el uso de materia, tiempo, personal y herramientas para ejecutar tareas de manera eficiente. Para lograrlo, los ingenieros han incorporado tecnologías para el monitoreo, integración y comunicación de un sistema. Con estas herramientas, se puede leer el estado actual del sistema y así, mediante algoritmos y rangos ya preestablecido se toman decisiones para llevar el sistema a un estado ideal, lo que garantiza un rendimiento más eficaz y preciso en todas las etapas del proceso. Con el avance tecnológico, los controladores se hicieron más compactos y potentes en los últimos 20 años, esto ha permitido implementar bucles de retroalimentación sin costos adicionales y ha facilitado la expansión de control automático a través de conexiones inalámbricas, permitiendo el monitoreo desde casi cualquier lugar en una planta tecnológica (Colonius et al., 2014).

Figura 1: Sistema Multivariado vs Sistema de control de red



Fuente: *Colonius et al. (2014)*

Al tratarse de un sistema con múltiples entradas y variables que presentan en puntos separados del sistema la red permite el intercambio directo de información entre los controladores locales, tal como se muestra en la fig. 1. Las redes digitales modernas ofrecen una ventaja única al permitir que esta comunicación se use de manera amplia, si la información intercambiada mejora el rendimiento del sistema.

8. Marco Teórico

Lenguajes de programación

Los lenguajes de programación son herramientas fundamentales en el desarrollo de software, permitiendo la comunicación entre el programador y la máquina. Según Sebesta (2016), los lenguajes de programación se dividen en diferentes paradigmas, como los lenguajes imperativos, orientados a objetos, funcionales y declarativos. Cada paradigma está diseñado para abordar problemas específicos de manera eficiente.

Un lenguaje como Python se popularizó por su simplicidad y versatilidad, utilizado en campos como la ciencia de datos, el desarrollo web y la inteligencia artificial (Lutz, 2013). Por otro lado, Java es un lenguaje robusto y orientado a objetos, ampliamente utilizado en sistemas empresariales y aplicaciones móviles (Deitel & Deitel, 2015).

APIs

Una API (Application Programming Interface) es un conjunto de definiciones y protocolos que permiten que diferentes sistemas de software se comuniquen entre sí. Según Fielding y Taylor (2000), una



API define los métodos y funciones que una aplicación puede utilizar para interactuar con otro software. Las APIs se han vuelto esenciales en la era digital, especialmente en aplicaciones web, donde permiten la integración de servicios externos como pagos, mapas y redes sociales. “Las API actúan como intermediarios que exponen los servicios empresariales o los activos de una organización a los desarrolladores, permitiéndoles construir aplicaciones que mejoren la experiencia del usuario.” (De, 2022)

Las API regularmente funcionan a través de la emisión de solicitudes HTTP, donde el cliente (Quien solicita los datos) realiza una petición como GET (Solicitar datos específicos del servidor), POST (Envía datos al servidor), PUT (Actualiza información existente en el servidor) o DELETE, este último procesa la solicitud y responde a este llamado con los datos respectivos, normalmente en formato JSON o XML, dos tipos de formatos que son capaces de leer las API y mostrar la información legible.

Las APIs suelen estar compuestas de los siguientes elementos (De, B., 2023):

- EndPoints: Son las URL o direcciones específicas donde se envían las solicitudes. Por ejemplo, el EndPoint para obtener un recurso de una API alojada en <https://www.nombres.com> podría ser <https://www.nombres.com/nombre>, donde nombre es un parámetro que determina el recurso específico solicitado.
- Métodos HTTP: Los métodos como GET, POST, PUT y DELETE indican las operaciones que se pueden realizar en la API. Son bastante intuitivos y se definen tal que: GET nos permite obtener datos, POST envía datos para crear un nuevo recurso, PUT guarda o actualiza un recurso existente, y DELETE elimina un recurso específico.
- Parámetros: Estos son valores que se envían junto con la solicitud para especificar datos específicos, como IDs o nombres. En este sentido, ese es el que permitirá a los EndPoint poder determinar qué recursos deseamos manejar.



- Autenticación: Muchas APIs requieren mecanismos de autenticación (como OAuth o claves de API) para controlar el acceso a los recursos.

Bases de datos

Las bases de datos son sistemas que permiten el almacenamiento, acceso y gestión de grandes volúmenes de información de manera organizada. Los dos tipos principales de bases de datos son las relacionales, que utilizan tablas y relaciones entre datos, y las NoSQL, que están diseñadas para manejar datos no estructurados o semiestructurados.

Según Connolly y Begg (2015), las bases de datos relacionales, como MySQL y PostgreSQL, se basan en el modelo de datos relacional propuesto por Edgar F. Codd, que utiliza SQL (Structured Query Language) para la manipulación de datos. Las bases de datos NoSQL, como MongoDB o Cassandra, son ideales para aplicaciones que requieren escalabilidad horizontal y manejo de grandes volúmenes de datos (Gourley & Somers, 2016).

Aprendizaje automático

Se define que el aprendizaje automático es una rama de la inteligencia artificial que busca que un programa de computador aprenda de un conjunto de datos con los cuales se entrena, y buscará identificar un patrón con el que puede realizar predicciones sobre nuevos datos. En primera instancia, se procesan datos de entrada y mediante reglas, se genera una salida (Pineda P, 2021), a diferencia de la programación tradicional. En sí, se define que el aprendizaje automático es capaz de aprender sin explícitamente programar la lógica o los conocimientos que tiene o tendrá.

El aprendizaje automático se basa en la idea de que los sistemas pueden entrenarse con grandes volúmenes de datos para encontrar patrones o regularidades, que luego pueden aplicarse a nuevos datos para realizar predicciones o clasificaciones. “ML systems utilize data to uncover rules and patterns, experimenting with various approaches to optimize performance” (“Introduction to Machine Learning”,



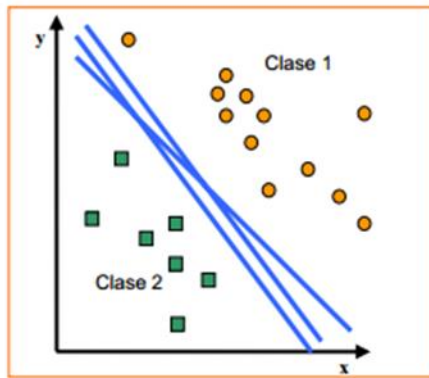
2022).

El proceso de aprendizaje en ML se puede dividir en tres etapas, primero el entrenamiento en donde el algoritmo aprende a partir de un conjunto de datos etiquetados (conocido como dataset). Posterior a esto, se genera la etapa de validación en la cual los modelos se evalúan utilizando un conjunto de datos de prueba para evitar el sobreajuste. Y finalmente la predicción, esto una vez que el modelo ha sido entrenado y validado, se usa para hacer predicciones o tomar decisiones con datos nuevos. Teniendo por completo un modelo predictivo.

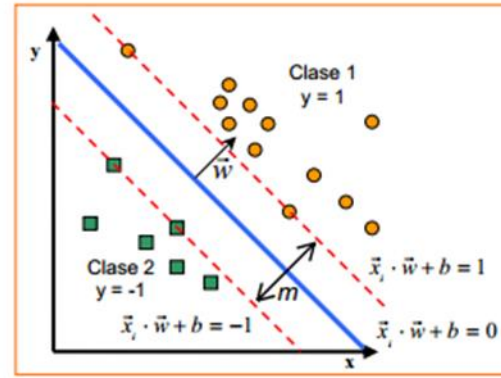
Como menciona Toscano, R. (2024), actualmente hay dos tipos de aprendizaje automático primarios, el aprendizaje supervisado en el cual se hace uso de datasets segmentados por etiquetas, se conoce qué es cada objetivo que se desea clasificar, y el aprendizaje no supervisado que en sí se usa un dataset sin etiquetas y el modelo buscará similitudes entre los datos y los clasificará según la similitud. Por ejemplo, en el aprendizaje automático se puede crear un modelo que identifique gatos o perros, en el aprendizaje supervisado el dataset separa explícitamente cuáles son gatos y cuales perros, en el no supervisado todos los datos están mezclados sin etiqueta o diferencia alguna.

Para hacer uso de los Datasets y poner en marcha cualquiera de los tipos de aprendizaje automático, se hacen uso de diversos algoritmos elementales, el más sencillo es el algoritmo de Regresión Lineal, este predice un valor numérico en función de las características de entrada. Es uno de los métodos más simples y se usa cuando la relación entre las variables es lineal. Por otro lado, tenemos un algoritmo más complejo para tareas no tan sencillas o lineales, este algoritmo es el de Máquinas de Soporte Vectorial (SVM) que funciona clasificando los datos mediante la creación de un hiperplano que separa las distintas clases de datos.

Figura 2: SVM gráficamente



Posibles hiperplanos de separación



Hiperplano de separación óptimo

Badouin (2022)

En cuanto a algoritmos más específicos, se encuentran las Redes Neuronales, las cuales, como su nombre indica, buscan imitar el funcionamiento del cerebro humano. Estas son particularmente eficaces en tareas como el reconocimiento de imágenes y el procesamiento del lenguaje natural, como, por ejemplo, en la clasificación de perros y gatos.

Asimismo, se dispone del algoritmo de Árboles de Decisión, el cual crea un modelo de decisiones en forma de árbol, donde cada nodo representa una decisión basada en una característica específica. Por otro lado, el algoritmo K-Means es un método de clustering que agrupa los datos en clusters o grupos basados en la similitud entre ellos. Los clusters son conjuntos formados a partir de similitudes, y este algoritmo va generando grupos de valores para definir cada conjunto (cluster) y asignarle una etiqueta.

Modelos predictivos

Los modelos predictivos son una parte esencial del análisis de datos y la inteligencia artificial, utilizados para predecir futuros comportamientos o eventos basándose en datos históricos. Estos modelos pueden construirse con algoritmos de aprendizaje automático, como la regresión lineal, los árboles de decisión, las redes neuronales y los de bosque aleatorio.

Según James et al. (2013), los modelos predictivos se utilizan en una variedad de aplicaciones, desde predicciones de mercado y detección de fraudes, hasta análisis de riesgo y recomendaciones personalizadas. La capacidad de estos modelos para aprender de los datos y mejorar con el tiempo los hace valiosos en el contexto de grandes volúmenes de información. En esencia, su objetivo es ayudar a



los sistemas a "aprender" de la información disponible y mejorar su precisión con el tiempo.

Dentro de la rama de los modelos predictivos se encuentran los modelos de regresión, estos son técnicas estadísticas que permiten relacionar una variable dependiente (para este caso, los precios) con una o más variables independientes (por ejemplo, la calidad de los alimentos, el tiempo de almacenamiento, o la demanda). Los modelos de regresión más comunes son los lineales, sin embargo, al tratar con múltiples variables y una correlación no lineal se debe optar por ajustar una curva como en la regresión polinómica. Este tipo de modelo es especialmente útil cuando las relaciones entre las variables son más complejas y no pueden describirse adecuadamente con una línea recta.

Degradación de productos orgánicos

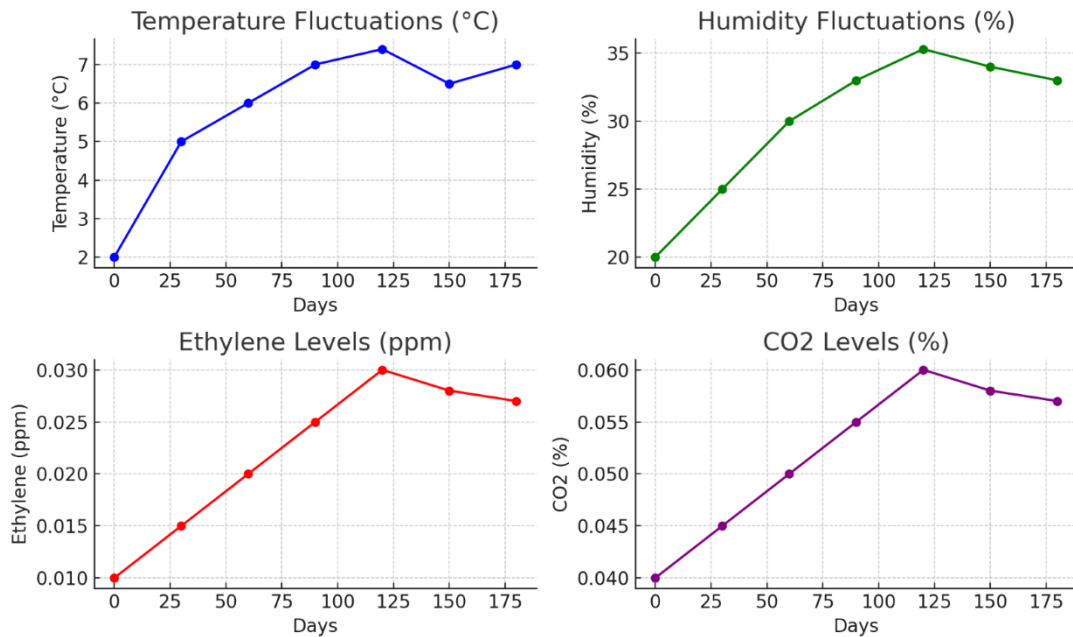
Parte esencial de la cadena de consumo radica en el almacenamiento y comercialización, sin embargo, este proceso es una carrera contra el tiempo, pues todos los productos pasan por un proceso de descomposición. Los factores que intervienen en la duración de un producto y a su vez en el deterioro fisiológico acelerado son: las altas temperaturas (mayor actividad de agua), las bajas humedades relativas y el daño físico (llamado daño mecánico) (Mora Fallas H. 2017). Adicionalmente existen factores como la exposición prolongada a la luz, especialmente a luz ultravioleta, el crecimiento de bacterias, hongos y otros patógenos puede acelerar la degradación, descoloración y pérdida de nutrientes.

Adicionalmente, la pérdida de calidad de los alimentos frescos en los comercios es bastante alta, especialmente cuando el estrés ambiental es elevado. En algunos casos, el almacenamiento inadecuado y subóptimo de este tipo de alimentos puede causar intoxicación alimentaria y lesiones personales (Kim et



al., 2015). Estudios realizados por Aguiar et al. (2022) muestran cómo varían las condiciones de almacenamiento a lo largo del tiempo. Los niveles de CO₂ comienzan en un 0.04% y aumentan hasta un 0.06%, lo que indica un incremento en la tasa de respiración de los alimentos. Este aumento en CO₂ es típico de la respiración de frutas y verduras almacenadas, lo que puede acelerar su degradación si no se controla adecuadamente, este comportamiento se puede mejorar en fig. 2.

Figura 3: Fluctuación de las variables claves en la descomposición



Fuente: Adaptado de Aguiar et al. (2022)

Es importante destacar la relación entre variables, pues en frutas las altas temperaturas aceleran la liberación de etileno, baja humedad puede incrementar el estrés, liberando más etileno y acelerando la degradación. La pérdida de firmeza, color amarillento, y deterioro son signos del incremento del CO₂ debido a la respiración.



9. Análisis de restricciones

La predicción de precios para productos perecederos requiere un volumen considerable de datos, tanto de precios históricos como de características de su estado a lo largo del tiempo. Por consiguiente, es esencial asegurar la consistencia de estos datos para obtener resultados confiables. La adquisición en tiempo real exige la instalación de sensores que monitoreen variables ambientales y químicas, además de realizar mediciones continuas para determinar el grado de frescura del producto. Sin embargo, la dificultad para obtener conjuntos de datos históricos y de frescura precisos podría afectar significativamente la exactitud entre las predicciones y las mediciones.

A nivel de procesamiento, integrar los datos en un algoritmo que realice ajustes de precios dinámicos requiere un análisis constante y un poder de cómputo considerable, especialmente si el sistema debe ser autónomo y rápido para ajustarse a condiciones cambiantes. Por lo que para una aplicación práctica se debe reducir la cantidad de lecturas al día, debido a los tiempos limitados para el desarrollo, priorizando la calidad sobre cantidad de datos.

Restricciones Económicas: El presupuesto asignado para la implementación del sistema de sensores, procesamiento de datos y algoritmos predictivos puede restringir la selección de ciertas tecnologías. También se debe considerar el gasto energético asociado con el funcionamiento continuo del sistema. Para que la solución sea viable, especialmente para pequeños y medianos comerciantes, el costo total



debe ser asequible; Por lo que se ha dispuesto de microcontroladores de bajo costo y alto rendimiento para la implementación. Sin embargo, el entrenamiento del modelo ya sea por parte de un profesional, o adquirir el sistema de predicción completo puede ser un factor económico a tener en cuenta.

Restricciones de Salud y Seguridad: El sistema debe estar diseñado de manera que no represente un riesgo para los trabajadores o consumidores. Los sensores y dispositivos deben ser instalados de acuerdo con normas de seguridad eléctrica y ocupacional dados en el Reglamento Técnico de Instalaciones Eléctricas (RETIE) y la Resolución 5018 de 2019 del Ministerio del Trabajo. Las partes energizadas peligrosas no deben ser accesibles a personas no competentes, y las partes accesibles no deben ser peligrosas, por lo que el microcontrolador y la fuente de alimentación deben ir ocultas, además se debe tener en cuenta que el entorno de desarrollo se encuentra clasificado como áreas peligrosas o de alto riesgo, pues se pueden generar atmósferas potencialmente inflamables o explosivas. Sin embargo, debido a que abarca áreas donde se puede esperar que exista la presencia de una atmósfera de gas inflamable de manera ocasional o poco frecuente debido a la baja acumulación de gases, esta se encuentra dentro de las Zonas 1, clase I: Gases, vapores y líquidos inflamables.

Restricciones Ambientales: La normatividad se encuentra establecida bajo el decreto 284 del 2018 del Ministerio de Ambiente y Desarrollo Sostenible. Esto implica la separación y clasificación de los dispositivos en el lugar de origen, promoviendo su reciclaje y reutilización para minimizar el impacto ambiental. Además, los sensores deben ser entregados en puntos de acopio autorizados, evitando su vertido en sitios no adecuados o su incineración sin tratamiento, para prevenir la liberación de sustancias tóxicas. (Ministerio de Ambiente y Desarrollo Sostenible, 2022).

Restricciones Sociales: Es importante entender que este sistema de ventas es novedoso en el país, si bien es común la fluctuación de precios de locales a lo largo de las semanas, es probable que tantos los consumidores como los vendedores se sientan incómodos con la fluctuación de precios basada en factores que no entienden completamente. También podría haber una percepción de desigualdad si los precios



varían considerablemente de un lugar a otro o en lapsos muy cortos del día, por lo que es importante hacer una asimilación gradual con los involucrados y concientizar acerca de los benéficos que trae la automatización.

Restricciones Éticas: La implementación de inteligencia artificial en aspectos cotidianos ha sido fuertemente criticada, la falta de claridad sobre cómo y por qué los precios cambian puede generar desconfianza entre los consumidores, pues tienen derecho a entender por qué un producto que compraron a un precio cambia repentinamente. Adicionalmente empresas que ya han incorporado tarifas dinámicas, como es el caso la cadena de mercados Kroger en EE. UU, yendo aún más allá e incorporando sistemas de reconocimiento facial para mostrar publicidad personalizada, generando un debate sobre el uso de información personal (Bean, 2024).

10. Metodología y desarrollo central

Para la elaboración del modelo, se ha dispuesto de un modelo de regresión polinomial, cuya naturaleza radica en la necesidad de considerar múltiples variables independientes para hacer una predicción razonable de los precios. Los modelos de regresión polinomial y de múltiples variables permiten predecir patrones más complejos de variaciones en la demanda, lo cual resulta especialmente útil cuando el deterioro de los productos no sigue una trayectoria simple y puede aumentar exponencialmente en ciertas condiciones. Para la creación del modelo de prueba, fue necesario adquirir datos tanto del comportamiento de los precios de alimentos como de los rangos de variables ambientales y químicas de los productos, que influyen directamente en su frescura y calidad.

Instrumentos para la recolección de información

Variables

La calidad de los datos es crucial para la efectividad del sistema. Esto incluye la precisión, consistencia y



rigurosidad de los datos. Para la ejecución del proyecto se requiere de la recolección de datos cuantitativos continuos.

Los datos clave a identificar y medir se dividen entre el sistema de prueba y la creación del algoritmo. En el caso de estudio se busca medir temperatura, humedad y concentración de gases (como CO₂ y etileno), pues estos influyen en la frescura de los alimentos. Mientras que, para el entrenamiento del algoritmo, se tomaron en cuenta datos históricos de los precios de los alimentos y los patrones de consumo en intervalos previos, mediante el uso de bases de datos de precios históricos proporcionados del DANE, entidad gubernamental encargada de generar estadísticas oficiales en Colombia. De esta manera se pudo identificar tendencias en el precio del producto. Estos datos deben tener un amplio volumen para entrenar el algoritmo predictivo, permitiendo obtener una representación adecuada del comportamiento real de los mercados colombianos.

Por otro lado, se debe establecer un programa con un comportamiento cercano a la realidad, útil en la práctica. Para esto se precisa de modelos de manejo de inventario que sean comúnmente empleados en la industrial. Existen muchas formas de manejar inventarios de las empresas, tales como el EOQ, donde es la cantidad de órdenes que minimiza la suma de los costos de ordenar y mantener de los inventarios de la compañía” (Obaidullah, 2019).

Gestión de alimentos perecederos en Colombia

Tras realizar consultas bibliográficas se encontraron técnicas de administración que, si bien datan de mediados del siglo XX, su efectividad mantiene su vigencia hasta hoy en día, impactando en áreas como la contabilidad de costos o teorías de logística y cadena de suministros. Dentro de ellas se destacan metodologías como la ya mencionada EOQ, FIFO (Primero en Entrar, Primero en Salir) y JIT (Just in Time). En el FIFO se busca que los productos con menor vida útil se utilicen primero, minimizando las pérdidas por caducidad, mientras que JIT se basa en ofertar



solo lo necesario, cuando sea necesario y en la cantidad necesaria (Heizer et al, 2021). Estos se combinan eficazmente con el modelo para optimizar la administración de productos perecederos, pues se pueden enlazar la tasa de deterioro con pronósticos de demanda y ajustar tanto el volumen de compras como la rotación del inventario.

Los sistemas inteligentes pueden ayudar a optimizar la logística al mejorar la precisión en la predicción de la demanda y el abastecimiento, los productores pueden ajustar sus cadenas de distribución para minimizar el desperdicio de productos perecederos. Esto es importante pues en el sector frutícola, a pesar de las ventajas con las que el país cuenta, se presentan problemas a lo largo de los eslabones de la cadena de abastecimiento por problemas relativos a sobreoferta de producto, tecnología utilizada, estándares de calidad (Herrera M et. Al, 2014).

El uso de metodologías como inventarios periódicos o perpetuos, así como el uso de tecnologías como radiofrecuencia o códigos de barras son comunes para hacer un seguimiento a los productos, en palabras de Espejo M. (2022). Esta tecnología permite identificar y rastrear artículos durante todo su ciclo de vida en la tienda, desde la recepción hasta el despacho. Es en este tipo de tecnologías que este proyecto puede ayudar a mejorar. Por ejemplo, los chips RFID facilitan el control de inventarios y ofrecen la capacidad de realizar un seguimiento, así como los códigos de barras tradicionales. Además, permite realizar inventarios más rápidos y con menos errores, ya que no requiere que el lector esté en contacto directo con el producto.

Con la implementación de nubes virtuales de datos, la información puede ser compartida en tiempo real entre todos los participantes de la cadena de suministro, lo que mejora la visibilidad y facilita la toma de decisiones. Esto no solo ayuda a detectar sobreabastecimientos, sino que también optimiza la planificación de la demanda y el momento adecuado para reponer productos en los anaques. Adicionalmente, Henao-Banea C (2024), proporciona un acercamiento al



control de inventarios en pequeños comercios por medio de Machine Learning. A través de modelos de aprendizaje supervisado, como máquinas de soporte vectorial y modelos autorregresivos, se predice la demanda futura, lo que es crucial para el control de inventario. Posteriormente, se aplica un modelo de programación lineal entera que refina los pedidos semanales, maximizando el retorno de la inversión

Selección de Equipos y Dispositivos de Medición

Para el control de la planta se requiere de un microcontrolador con alta capacidad de procesamiento y baja latencia en comunicación, además debe de contar con la capacidad de realizar transmisión de datos a un servidor central sin la necesidad de módulos externos. Si bien el ESP32 es escogido por su procesador dual de 32 bits, bajo consumo de energía, compatibilidad con Wifi y bajo precio, la Raspberry Pi es una alternativa con mayor poder de cómputo, sin embargo, representa una inversión mucho más elevada que el ESP32, además de requerir mayor espacio físico. Adicionalmente se debe de disponer de módulos de



sensores discretos.

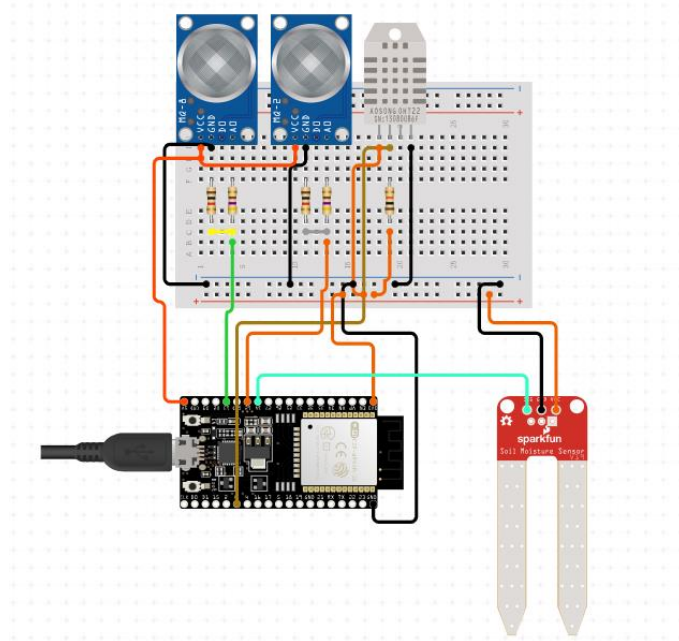
Estos son:

SENSOR	DESCRIPCIÓN	RESOLUCIÓN	PRECISIÓN	TAZA DE SONDEO
MQ7 Sensor de gases (monóxido de carbono)	Detecta concentraciones de monóxido de carbono (CO) en el aire. Funciona calentando un elemento de detección interno y mide la variación de resistencia según la concentración de CO.	10 - 1000 ppm de CO	±10%	1 - 2 Hz
MQ2 Sensor de gases	Es capaz de detectar glp, propano, i-butano, metano, alcohol e hidrogeno. Funciona midiendo cambios en la conductividad de su sensor interno según la presencia de gases,	300-10000ppm	Varía según el gas de lectura, ronda entre ±3%	Ajustable
YL69	Sensor de Humedad del suelo Resistivo Higrómetro.	0 - 1023 escala analoga	±5%	1 Hz
DHT11 Sensor de temperatura y humedad	Proporciona señales digitales que no requieren adaptación, lo que facilita su integración con microcontroladores.	1 °C 1% HR (Humedad Relativa)	±2 °C ±5% HR	0.5 Hz

Hay que detallar que el sensor de temperatura y humedad DHT11, dentro de sus consideraciones, incluyen una variación de 2°C y el cambio en un 20-80% de humedad, y debe ubicarse estratégicamente para evitar interferencias térmicas o de humedad provocadas por su entorno.

En cuanto a la familia de los MQ, funcionan midiendo cambios en la conductividad de su sensor interno según la presencia de gases, lo que genera una señal analógica que requiere conversión. La calibración es esencial para evitar falsos positivos. Además, este debe usarse en el pin 36 del ESP32 debido a temas de compatibilidad entre la comunicación WiFi y transmisión de datos entre pines.

Figura 4: Esquema de conexiones de sensores



Fuente: Elaboración propia

Para calibrar adecuadamente el circuito se debió hacer arreglos para el manejo adecuado de la sensibilidad de cada sensor. Además, se tiene que hacer una medición inicial para calibrar correctamente la reacción del sensor ante la interacción con gases. En primera medida se debe de establecer la sensibilidad del sensor al aire, conocido como R_0 , una vez conocida la resistencia del sensor (R_S), se utiliza la relación de calibración en la que el cociente entre R_S y R_0 es constante ($R_S / R_0 = 4.4 \text{ ppm}$) para determinar el valor de R_0 . Este valor de R_0 se obtiene promediando las lecturas analógicas del sensor, convirtiéndolas a voltaje y aplicando la fórmula mencionada (Jaycon-Dev, 2023).

Código 1: Ajuste de sensibilidad del MQ2

```
void loop() {

    float sensor_volt; // Define la variable para el voltaje del sensor

    float RS_air; // Define la variable para la resistencia del sensor en aire fresco

    float R0; // Define la variable para R0

    float sensorValue = 0; // Define la variable para las lecturas analógicas

    // Toma 500 muestras para obtener una lectura más estable
```



```

for(int x = 0; x < 500; x++) {
    sensorValue += analogRead(A0); // Suma los valores analógicos del sensor 500 veces
}

sensorValue = sensorValue / 500.0; // Calcula el promedio de las lecturas

sensor_volt = sensorValue * (5.0 / 1023.0); // Convierte el promedio de la lectura del
sensor a voltaje

// Calcula la resistencia del sensor en aire fresco (RS_air)

RS_air = ((5.0 * 10.0) / sensor_volt) - 10.0;

// Calcula R0 en base a la resistencia en aire fresco

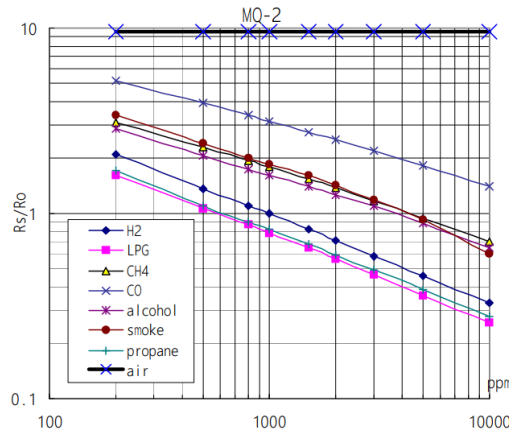
R0 = RS_air / 4.4; // Constante de calibración para la resistencia en aire fresco
}

```

Recuperado de Jaycon-Dev (2023)

De estas mediciones el R0 dio como resultado -1.27 , valor con el que empezaremos a trabajar en la configuración para el cálculo demás gases específicos como el metano (CH_4) y el etileno (C_2H_4). La Fig. 6 muestra las curvas de sensibilidad ante distintos gases del MQ2, si bien no se encuentra el etileno ni butano podemos hacer aproximaciones basadas en el a LPG, que incluye butano y el parecido entre etileno (C_2H_4) y propano (C_3H_8) como hidrocarburos simples.

Figura 5: Curva de sensibilidad MQ2



Recuperado de Hanwei Eletronics CO, s. F, 2019

Para el sistema de recolección de datos se siguió una metodología basada en la integración de sensores físicos en un recipiente con buena ventilación. En primer lugar, se seleccionaron y configuraron los sensores DHT11, YL9, MQ8 y MQ2 debido a su capacidad para medir variables ambientales relevantes concentraciones de gases en partes por millón (ppm). Posteriormente, se implementaron las librerías específicas de cada sensor, asegurando la correcta inicialización y calibración.

```
#include <DHT.h>
#include <MQUnifiedsensor.h>
#include <WiFi.h>
#include <HTTPClient.h>
```

La calibración del MQ2 se llevó a cabo en aire limpio, estableciendo un valor base para la relación R_s/R_0 , de acuerdo con las especificaciones del fabricante (Hanwei Electronics, 2019). Adicionalmente, se realizaron ajustes en las curvas de sensibilidad de los gases detectados por el MQ2.

```
// Inicialización del sensor MQ2
MQUnifiedsensor MQ2(Board, Voltage_Resolution, ADC_Bit_Resolution, MQ2_PIN, "MQ-2");
// Inicialización del DHT11 y validar lectura inicial
```



```
dht.begin();
delay(2000);
float h = dht.readHumidity();
float t = dht.readTemperature();

if (isnan(h) || isnan(t)) {
  Serial.println("Error: No se pueden leer datos del sensor DHT11. Verifique
conexiones.");
  return false;
}
Serial.println("Sensor DHT11: Lectura inicial correcta.");

// Configuración y calibración del sensor MQ2
MQ2.setRegressionMethod(1);
MQ2.init();
float calcR0 = 0;
for (int i = 1; i <= 10; i++) {
  MQ2.update();
  calcR0 += MQ2.calibrate(RatioMQ2CleanAir);
  delay(200);
}
MQ2.setR0(calcR0 / 10);

if (isinf(calcR0) || calcR0 == 0) {
  Serial.println("Error, problema en la calibración del sensor MQ2. Verifique
conexiones.");
  return false;
}
Serial.println("Sensor MQ2: Calibración y lectura inicial correctas.");
return true;
}
```

Adaptando las curvas predefinidas para gases como butano y etileno mediante aproximaciones



logarítmicas basadas en la documentación técnica disponible. El sistema también incluye una conexión Wi-Fi para la transmisión de datos, lo que permite una supervisión remota. Todo el código se escribió y verificó en el entorno de Arduino IDE, y las lecturas de depuración se habilitaron para garantizar el correcto funcionamiento y detectar posibles errores.

```
// Función para enviar datos al servidor
void enviarDatos(float temperatura, float humedad, float etileno, float co2,
float metano, float butano) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    http.begin(serverUrl);
    http.addHeader("Content-Type", "application/json");

    // Crear el objeto JSON con los datos
    String jsonData = String("{\"temperature\":") + temperatura +
      "\",\"humidity\":") + humedad +
      "\",\"ethylene\":") + etileno +
      "\",\"co2\":") + co2 +
      "\",\"methane\":") + metano +
      "\",\"butane\":") + butano + "}";

    // Enviar la solicitud POST
    int httpResponseCode = http.POST(jsonData);

    if (httpResponseCode > 0) {
      Serial.print("Datos enviados. Código de respuesta: ");
      Serial.println(httpResponseCode);
    } else {
      Serial.print("Error al enviar datos. Código de error: ");
      Serial.println(httpResponseCode);
    }
  }
}
```

Finalmente, estos datos son recibidos por la estación de monitoreo, validados y almacenados en un archivo CSV.



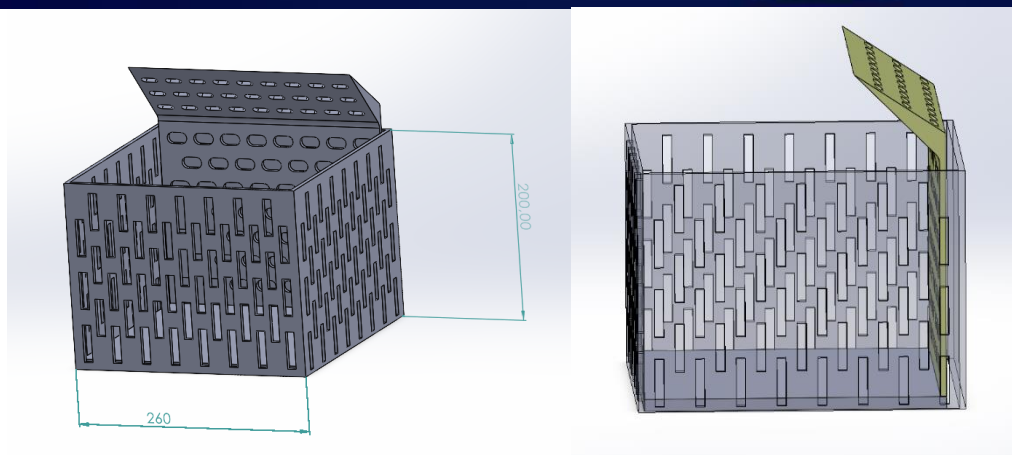
```
@app.route('/api/upload', methods=['POST'])
def upload_data():
    if request.method == 'POST':
        try:
            # Obtención de datos
            data = request.json
            temperature = data.get("temperature")
            humidity = data.get("humidity")
            ethylene = data.get("ethylene")
            co2 = data.get("co2")
            methane = data.get("methane")
            butane = data.get("butane")

            # Registro de fecha y hora
            timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

            # Guardado de datos archivo CSV
            with open(CSV_FILE, mode="a", newline="") as file:
                writer = csv.writer(file)
                writer.writerow([timestamp, temperature, humidity, ethylene, co2,
methane, butane])

            # Confirmación de recepción
            print(f"Datos recibidos y guardados: {data}")
            return jsonify({"status": "success", "message": "Data received and
saved successfully."}), 200
```

Figura 6: Cesta modificada para la medición



Fuente: Elaboración propia

Finalmente, para la medición se debe apoyar de un sistema que permita el almacenamiento y respiración de frutos, además de que las ventanas o aberturas estratégicas que permitan una circulación de corrientes ayudará a que los gases se muevan hacia los sensores. La fig. 7 muestra un sistema de almacenamiento, donde un compartimento de mayor tamaño almacena las frutas, mientras que una pequeña rejilla funciona como trampa para posicionar estratégicamente los sensores y atrapar los gases.

Recolección de Datos Externos para el Modelo Predictivo

En este caso, para contar con una fuente confiable de información sobre los precios y disponer de estos datos para generar correctamente el modelo predictivo, se realizó una búsqueda en una de las fuentes más veraces a nivel nacional de información y estadística: la plataforma del DANE (Departamento Administrativo Nacional de Estadística). Esta plataforma proporcionó artículos y documentos extensos sobre el historial de precios (mayoristas), de los cuales se extrajo la información específica. Fueron estos datos lo que permitió crear un conjunto de información adecuado para su análisis y posterior predicción. La entrada específica que se utilizó para la obtención de la información fue “Sistema de Información de Precios y Abastecimiento del Sector Agropecuario (SIPSA)” este en su componente “Precios mayoristas” que divide cada uno de sus boletines de manera diaria, semanal y mensual, para la facilidad del dataset y su correcta implementación se tomó el boletín mensual de todo el año 2020 hasta Agosto del 2024, esto



de manera manual se descargó y se procesó la información a través de Google Sheet (Excel de Google) y desde allí se realizaron las validaciones de qué información tenemos, y qué información utilizaremos.

Figura 7: Columnas/Variables de los boletines

Fecha	Grupo	Producto	Fuente	Precio
ene-20	VERDURAS Y HORTALIZAS	Acelga	Armenia, Mercar	1295
ene-20	VERDURAS Y HORTALIZAS	Acelga	Barranquilla, Barranquillita	3544
ene-20	VERDURAS Y HORTALIZAS	Acelga	Bogota, DC, Corabastos	900
ene-20	VERDURAS Y HORTALIZAS	Acelga	Bucaramanga, Centroabastos	1253
ene-20	VERDURAS Y HORTALIZAS	Acelga	Chiquinquirá (Boyaca)	1408
ene-20	VERDURAS Y HORTALIZAS	Acelga	Duitama (Boyaca)	1000
ene-20	VERDURAS Y HORTALIZAS	Acelga	Florencia (Caqueta)	1550
ene-20	VERDURAS Y HORTALIZAS	Acelga	Ibague, Plaza La 21	1258
ene-20	VERDURAS Y HORTALIZAS	Acelga	Ipiales (Narino), Ipiales somos todos	1547
ene-20	VERDURAS Y HORTALIZAS	Acelga	Medellin, Central Mayorista de Antioquia	1049
ene-20	VERDURAS Y HORTALIZAS	Acelga	Neiva, Surabastos	1775
ene-20	VERDURAS Y HORTALIZAS	Acelga	Pamplona (Norte de Santander)	2000
ene-20	VERDURAS Y HORTALIZAS	Acelga	Pasto, El Potrerillo	2462
ene-20	VERDURAS Y HORTALIZAS	Acelga	Sogamoso (Boyaca)	1167
ene-20	VERDURAS Y HORTALIZAS	Acelga	Tunja, Complejo de Servicios del Sur	1288
ene-20	VERDURAS Y HORTALIZAS	Acelga	Ubate (Cundinamarca)	769
ene-20	VERDURAS Y HORTALIZAS	Acelga	Villavicencio, CAV	1404
ene-20	VERDURAS Y HORTALIZAS	Ahuyama	Arauca (Arauca)	857
ene-20	VERDURAS Y HORTALIZAS	Ahuyama	Armenia, Mercar	986
ene-20	VERDURAS Y HORTALIZAS	Ahuyama	Barranquilla, Barranquillita	825
ene-20	VERDURAS Y HORTALIZAS	Ahuyama	Barranquilla, Granabastos	781
ene-20	VERDURAS Y HORTALIZAS	Ahuyama	Bogota, DC, Corabastos	1329
ene-20	VERDURAS Y HORTALIZAS	Ahuyama	Bucaramanga, Centroabastos	899
ene-20	VERDURAS Y HORTALIZAS	Ahuyama	Buenaventura (Valle del Cauca)	1467
ene-20	VERDURAS Y HORTALIZAS	Ahuyama	Cali, Cavasa	1177
ene-20	VERDURAS Y HORTALIZAS	Ahuyama	Cali, Santa Helena	1196
ene-20	VERDURAS Y HORTALIZAS	Ahuyama	Cartagena, Bazurto	952
ene-20	VERDURAS Y HORTALIZAS	Ahuyama	Cartago (Valle del Cauca)	1111
--	---	---	---	---

Fuente: Elaboración propia/Datos obtenidos de <https://www.dane.gov.co/index.php/estadisticas-por-tema/agropecuario/sistema-de-informacion-de-precios-sipsa/mayoristas-boletin-mensual-1>

Al revisar el boletín se decidió guardar el dataset en esta manera tal como lo comparte el DANE ya que, en sí, la variable "Fecha" resulta esencial, ya que permite capturar la variabilidad temporal de los precios. En modelos de predicción de precios, las fluctuaciones pueden depender de factores estacionales, tendencias o eventos específicos de tiempo. La inclusión de esta variable posibilita que el modelo aprenda patrones temporales, ajustando el precio a las condiciones del mercado en diferentes momentos.

Por otro lado, la variable "Grupo" agrupa los productos en categorías o segmentos. Esta información es



fundamental para identificar comportamientos o tendencias específicas dentro de diferentes grupos de productos. De esta manera, el modelo puede prever cómo el precio de un producto determinado podría comportarse en relación con otros productos dentro del mismo grupo, mejorando la precisión de las predicciones. Con esto podemos decir que la "Fuente" hace referencia a la procedencia de los datos o las condiciones del mercado de donde se obtienen los precios. Dependiendo de la fuente, los precios pueden variar debido a factores como ubicación geográfica, el tipo de proveedor o las condiciones del mercado en el momento de la medición. Incluir esta variable permite al modelo captar las diferencias en los precios derivados de fuentes diversas y ajustar las predicciones conforme a estas variaciones. Concluyendo, es esencial el "Precio", ya que es la variable objetivo del modelo, es decir, la variable que el modelo de aprendizaje automático intentará predecir. Al ser la variable dependiente, se ajusta en función de las características de las demás columnas (Fecha, Grupo, Producto y Fuente). La presencia de estas variables es fundamental para que el modelo pueda hacer predicciones precisas sobre el comportamiento del precio.

Los boletines del DANE tienen una particularidad y es que, desde la fecha inicial, enero del 2020 hasta el boletín más actual, se hace uso de la misma plantilla de Columnas facilitando así la preparación del dataset final.



Figura 8: Columnas/VARIABLES de los boletines

1	Fecha	Grupo	Producto	Fuente	Precio
253692	ago-24	PROCESADOS	Sardinas en lata	Barranquilla, Granabastos	11840
253693	ago-24	PROCESADOS	Sardinas en lata	Bogota, DC, Corabastos	15786
253694	ago-24	PROCESADOS	Sardinas en lata	Bogota, DC, Plaza Espana	16777
253695	ago-24	PROCESADOS	Sardinas en lata	Cali, La Floresta	13664
253696	ago-24	PROCESADOS	Sardinas en lata	Cali, Santa Elena	17059
253697	ago-24	PROCESADOS	Sardinas en lata	Cali, Siloe	13950
253698	ago-24	PROCESADOS	Sardinas en lata	Cartagena, Bazurto	11308
253699	ago-24	PROCESADOS	Sardinas en lata	Cartago (Valle del Cauca)	17158
253700	ago-24	PROCESADOS	Sardinas en lata	Cucuta, Cenabastos	16480
253701	ago-24	PROCESADOS	Sardinas en lata	Cucuta, La Nueva Sexta	16869
253702	ago-24	PROCESADOS	Sardinas en lata	Manizales, Centro Galerias	17678
253703	ago-24	PROCESADOS	Sardinas en lata	Medellin, Central Mayorista de Antioquia	9276
253704	ago-24	PROCESADOS	Sardinas en lata	Medellin, Plaza Minorista "Jose Maria Villa"	9309
253705	ago-24	PROCESADOS	Sardinas en lata	Monteria, Mercado del Sur	12418
253706	ago-24	PROCESADOS	Sardinas en lata	Palmira (Valle del Cauca)	19203
253707	ago-24	PROCESADOS	Sardinas en lata	Pamplona (Norte de Santander)	16471
253708	ago-24	PROCESADOS	Sardinas en lata	Popayan, Plaza de mercado del barrio Bolivar	14239
253709	ago-24	PROCESADOS	Sardinas en lata	Santa Marta (Magdalena)	10395
253710	ago-24	PROCESADOS	Sardinas en lata	Sincelejo, Nuevo Mercado	9600
253711	ago-24	PROCESADOS	Sardinas en lata	Tulua (Valle del Cauca)	16613
253712	ago-24	PROCESADOS	Sardinas en lata	Villavicencio, CAV	16865
253713	ago-24	PROCESADOS	Sopa de pollo (caja)	Bogota, DC, Corabastos	24148
253714	ago-24	PROCESADOS	Sopa de pollo (caja)	Bogota, DC, Plaza Espana	24921
253715	ago-24	PROCESADOS	Vinagre	Barranquilla, Barranquillita	1985
253716	ago-24	PROCESADOS	Vinagre	Barranquilla, Granabastos	1940
253717	ago-24	PROCESADOS	Vinagre	Bucaramanga, Centroabastos	2155
253718	ago-24	PROCESADOS	Vinagre	Bucaramanga, Mercados del centro	3156
253719	ago-24	PROCESADOS	Vinagre	Cartagena, Bazurto	1492
253720	ago-24	PROCESADOS	Vinagre	Cucuta, Cenabastos	2383
253721	ago-24	PROCESADOS	Vinagre	Cucuta, La Nueva Sexta	2369
253722	ago-24	PROCESADOS	Vinagre	Monteria, Mercado del Sur	1708
253723	ago-24	PROCESADOS	Vinagre	Pamplona (Norte de Santander)	1833
253724	ago-24	PROCESADOS	Vinagre	Santa Marta (Magdalena)	1519
253725	ago-24	PROCESADOS	Vinagre	Sincelejo, Nuevo Mercado	1328
253726	ago-24	PROCESADOS	Vinagre	Valledupar, Mercado Nuevo	1397

Fuente: Elaboración propia/Datos obtenidos de <https://www.dane.gov.co/index.php/estadisticas-por-tema/agropecuario/sistema-de-informacion-de-precios-sipsa/mayoristas-boletin-mensual-1>

Finalmente obtenemos el documento anteriormente presentado, en donde vemos que después de recopilar toda la información contamos con un dataset con una cantidad de 253.725 datos (Omitiendo la fila 1 que corresponde a las columnas/variables), suficiente para procesar y aplicar en el modelo de aprendizaje automático.

En la revisión manual se encontró que no hay datos nulos en todo el dataset, sin embargo, el dataset será evaluado en estos aspectos y en caso de determinar valores nulos se eliminarán para evitar el ruido (variabilidad o información irrelevante presente en los datos que dificulta que el modelo identifique los



patrones subyacentes correctamente) en el dataset y el aprendizaje del modelo.

Procesamiento y programación del Modelo Predictivo

Con el dataset establecido podemos dar pie al desarrollo del modelo que nos permitirá predecir más adelante los precios estimados de los productos.

Lo primero que se realiza es la descarga del dataset desde Google Sheet a un archivo externo en formato .csv (comma-separated value – Valores separados por coma), suelen importarse en este tipo de archivos debido a su simplicidad, ligereza y compatibilidad. Este formato organiza los datos en filas y columnas separadas por delimitadores como comas, siendo fácil de interpretar por herramientas y lenguajes de programación. Su bajo peso y estructura de texto plano lo hacen eficiente para almacenar grandes volúmenes de información, como en nuestro caso con el dataset de 253.725 registros. Si bien carecen de elementos gráficos o jerárquicos, su popularidad se debe a su facilidad de manejo y estandarización en el intercambio de datos tabulares.

Todo el modelo se desarrolló en Google Colab, un servicio en la nube que permite el uso de lenguajes de programación enfocados en el Machine Learning. Por facilidad del modelo que se empleó de regresión lineal se hizo uso del lenguaje de programación R, que es un lenguaje enfocado en la analítica de datos y Machine Learning. Siendo así, una de las primeras líneas de código que ejecutamos es:

`install.packages("caret")` esto para tener todas las funciones necesarias para desarrollar el modelo correspondiente. Caret significa “Classification And REgression Training” y este paquete es el más versátil simplificar y unificar los procesos de modelado predictivo, tanto para tareas de clasificación como de regresión. Su principal objetivo es proporcionar herramientas consistentes y fáciles de usar para entrenar, evaluar y comparar modelos de aprendizaje supervisado.

Ahora bien, con el dataset cargado en Colab y el paquete “caret” descargado e implementado, podemos empezar a visualizar la información, la línea que nos permitirá traer los datos es:



```
precios_histor <- read.csv('dataset_precios.csv')
```

```
str(precios_histor)
```

Dándonos el siguiente Output/Salida:

```
'data.frame': 253725 obs. of 5 variables:
 $ Fecha : chr "ene-20" "ene-20" "ene-20" "ene-20" ...
 $ Grupo : chr "VERDURAS Y HORTALIZAS" "VERDURAS Y HORTALIZAS" "VERDURAS Y HORTALIZAS" "VERDURAS Y HORTALIZAS" ...
 $ Producto: chr "Acelga" "Acelga" "Acelga" "Acelga" ...
 $ Fuente : chr "Armenia, Mercar" "Barranquilla, Barranquillita" "Bogota, DC, Corabastos" "Bucaramanga, Centroabastos" ...
 $ Precio : num 1295 3544 900 1253 1408 ...
```

Correspondiente al dataset que se creó y mostró anteriormente desde Google Sheet, la función "str" nos da la visual general del dataset, las columnas/variables, el tipo de dato que manejan las variables, los registros que coinciden con el dataset de excel, 253.725 y una visual de los datos que se encontrarán en cada una de las variables.

En R, al implementar una regresión lineal, es necesario convertir las variables categóricas a factores debido a cómo este tipo de modelos interpreta los datos. Las variables categóricas, como "Grupo", "Producto" y "Fuente" en el ejemplo, están originalmente almacenadas como cadenas de texto (chr), lo que no permite que el modelo las reconozca correctamente como categorías. Al transformarlas a factores, se asignan niveles únicos a cada categoría, permitiendo que el modelo las procese adecuadamente como variables explicativas.

```
#Conversión a factores
precios_histor$Grupo <- as.factor(precios_histor$Grupo)
precios_histor$Producto <- as.factor(precios_histor$Producto)
precios_histor$Fuente <- as.factor(precios_histor$Fuente)
```

```
#Verificar si hay valores NA (Nulos)
sum(is.na(precios_histor))
```

```
# Eliminar los datos NA (Nulos)
na.omit(precios_histor)
```

Además, se verifica la existencia de valores faltantes (NA) porque estos pueden interferir en el ajuste del modelo. Para garantizar que los datos sean adecuados para el análisis, se eliminan los valores faltantes mediante `na.omit()`. Esto asegura que la regresión lineal utilice únicamente observaciones completas y válidas. En resumen, estos pasos son necesarios para preparar y limpiar los datos, asegurando la correcta



interpretación de las variables y un análisis confiable.

Se realiza nuevamente la validación de los datos con la función “str” y confirmar que los datos se convirtieron en factores correctamente como lo esperábamos.

```
'data.frame': 253725 obs. of 5 variables:
 $ Fecha : chr "ene-20" "ene-20" "ene-20" "ene-20" ...
 $ Grupo : Factor w/ 16 levels "Carnes","CARNES",...: 16 16 16 16 16 16 16 16 16 ...
 $ Producto: Factor w/ 364 levels "Aceite de palma",...: 6 6 6 6 6 6 6 6 6 ...
 $ Fuente : Factor w/ 94 levels "Ancuya (Narino)",...: 3 5 8 15 30 34 38 40 43 53 ...
 $ Precio : num 1295 3544 900 1253 1408 ...
```

Con esto podemos confirmar que efectivamente las variables “Grupo”, “Producto” y “Fuente” actualizaron su tipo de dato a “Factor” con sus “niveles” correspondientes a cada una de las categorías descritas en cada columna.

Con esto podemos hacer uso del siguiente código:

```
precios_histor <- precios_histor[, c("Grupo", "Fuente", "Producto", "Precio")]
```

Que en sí nos permite especificar las columnas que utilizaremos para el entrenamiento del modelo y que explicamos el motivo de cada variable anteriormente.

```
set.seed(123)
```

```
train_index <- createDataPartition(precios_histor$Precio, p=0.7, list=FALSE)
train_data <- precios_histor[train_index,]
test_data <- precios_histor[-train_index,]
```

Este fragmento de código en R realiza la división del conjunto de datos en dos subconjuntos:

entrenamiento y prueba. Se explica de la siguiente manera; *set.seed(123)*: Establece una semilla para el generador de números aleatorios, lo que garantiza que la división del conjunto de datos sea reproducible.

Es decir, al ejecutar el código varias veces con la misma semilla, se obtendrán los mismos subconjuntos.

Se utiliza la función *createDataPartition* de la librería *caret* para crear índices que dividen el conjunto de datos de forma estratificada. Esto significa que la proporción de valores en la variable objetivo (Precio) se mantiene consistente entre los subconjuntos. El parámetro *p=0.7* especifica que el 70% de los datos se destinarán al conjunto de entrenamiento. La opción *list=FALSE* asegura que el resultado sea un vector de índices, no una lista. Finalmente se toman las filas correspondientes al conjunto de entrenamiento y son



almacenadas en el objeto *train_data*.

`test_data <- precios_histor[-train_index,]`: Selecciona las filas restantes (El 30% restante), que no están incluidas en el conjunto de entrenamiento, y las almacena en el objeto *test_data*, creando así el conjunto de prueba.

Ahora bien, para asegurar la integridad de los datos y que estos sean coincidentes se asegura que los factores de la variable *Producto* en los conjuntos de entrenamiento y prueba sean consistentes, evitando niveles que no están presentes en ambos conjuntos. Se explica de la siguiente manera:

```
niveles_comunes <- intersect(levels(train_data$Producto), levels(test_data$Producto))
```

Esta línea identifica los niveles comunes entre los factores de la variable *Producto* en los conjuntos de entrenamiento (*train_data*) y prueba (*test_data*) utilizando la función `intersect`. Esto crea un vector que contiene solo los niveles compartidos entre ambos conjuntos.

```
train_data <- train_data[train_data$Producto %in% niveles_comunes, ]
```

Después se filtra el conjunto de entrenamiento para incluir únicamente las filas donde el nivel del factor *Producto* esté presente en los niveles comunes identificados previamente.

```
test_data <- test_data[test_data$Producto %in% niveles_comunes, ]
```

De la misma manera se aplica el mismo filtro al conjunto de prueba, manteniendo solo las filas con niveles de *Producto* que coincidan con los niveles comunes.

```
train_data$Producto <- droplevels(train_data$Producto)
```

Teniendo esto, se reajusta los niveles del factor *Producto* en el conjunto de entrenamiento, eliminando los niveles que ya no están presentes después del filtrado.

```
test_data$Producto <- droplevels(test_data$Producto)
```

Y finalmente se realiza el mismo reajuste en el conjunto de prueba, asegurándose de que los factores contengan únicamente los niveles activos.

Con todo el dataset configurado, preprocesado y definido con los datos coincidentes, correctos y coherentes, se da pie a la implementación del modelo de regresión lineal para entrenar el modelo.



```
linear_model <- glm(Precio ~ Grupo + Fuente + Producto, data = train_data, family = "gaussian")
```

El código anterior crea un modelo lineal generalizado utilizando la función `glm`, donde la variable dependiente **Precio** se modela como una combinación lineal de las variables independientes **Grupo**, **Fuente** y **Producto**. Los datos para ajustar el modelo provienen del conjunto de entrenamiento especificado como `train_data`, y se utiliza una distribución gaussiana para los errores, lo que corresponde a una regresión lineal tradicional. Este tipo de modelo se prioriza por su simplicidad y claridad, ya que permite interpretar directamente el impacto de cada variable independiente en la variable dependiente a través de los coeficientes estimados. Además, es adecuado cuando existe una relación lineal entre las variables, ya que es capaz de capturar esta relación sin riesgo de sobreajuste.

Los modelos lineales se utilizan frecuentemente como base para análisis exploratorios, permitiendo evaluar inicialmente las relaciones entre las variables antes de probar con modelos más complejos. También son una opción robusta cuando los datos son limitados o ruidosos, ya que modelos más avanzados requieren un mayor volumen de datos para generalizar correctamente. Por estas razones, se elige el modelo lineal como primera alternativa en este caso. Con el modelo ya entrenado se realiza una validación de los datos para que las etiquetas/valores correspondientes a las variables que se usan para predecir el precio sean coincidentes y evitar errores de predicción esto con el siguiente código:

```
test_data$Grupo <- factor(test_data$Grupo, levels = levels(train_data$Grupo))
test_data$Fuente <- factor(test_data$Fuente, levels = levels(train_data$Fuente))
test_data$Producto <- factor(test_data$Producto, levels =
levels(train_data$Producto))
```

Posteriormente, se utilizan estos datos para realizar predicciones con el modelo ajustado previamente (`linear_model`) mediante la función `predict`.

```
predictions_continuous <- predict(linear_model, test_data)
```

Finalmente, se visualizan las primeras predicciones con la función `head` para obtener una vista

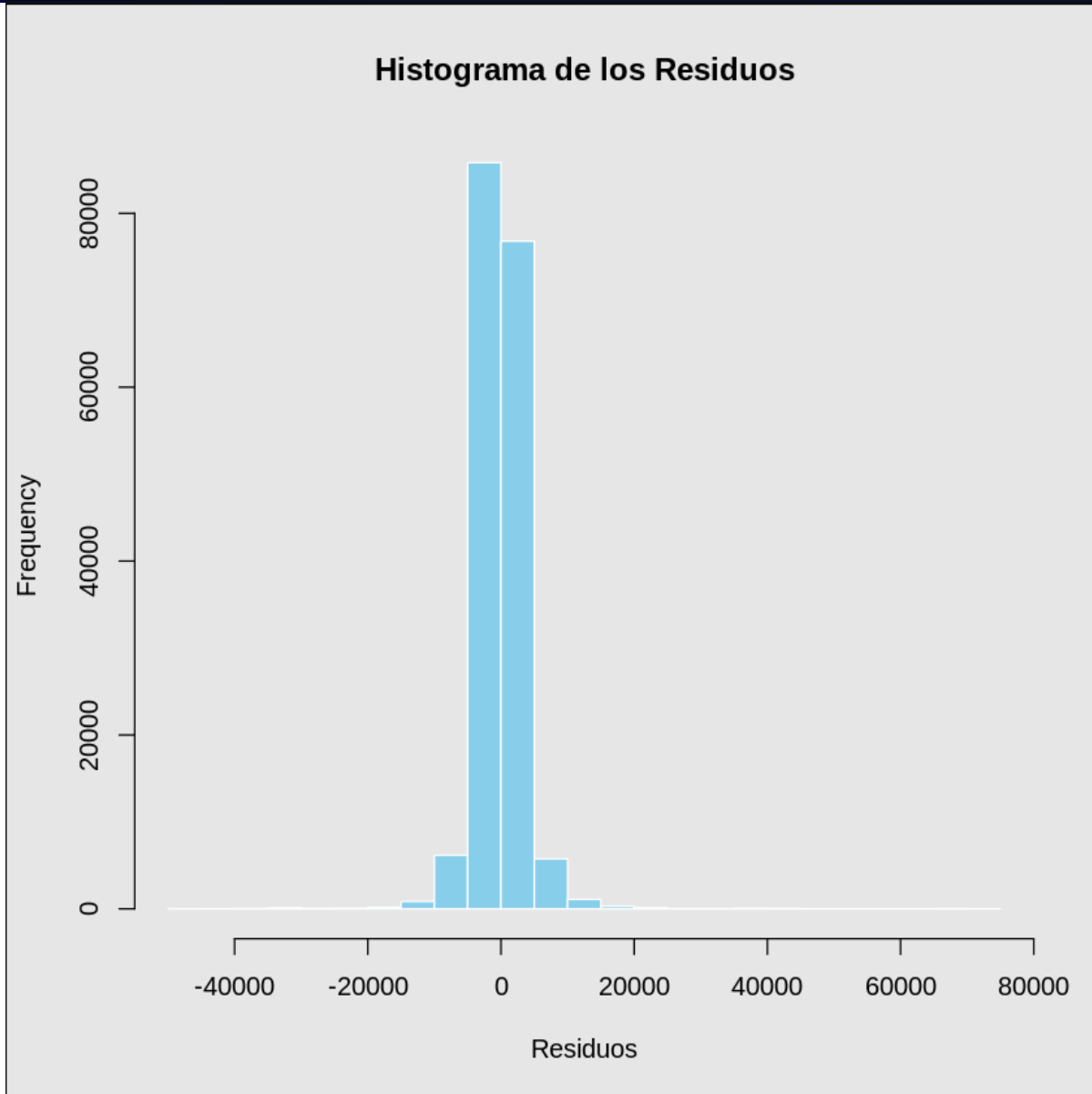


preliminar de los resultados. Este procedimiento asegura que el modelo pueda aplicar las reglas aprendidas en el entrenamiento sin errores derivados de niveles categóricos no reconocidos.

```
head(predictions_continuous)
```

Para validar que el modelo funciona se puede ir directamente al dataset y buscar manualmente precios y confirmar que tienen coherencia con lo que nos indica el modelo generado. Sin embargo, este proceso manual no se debe ni se puede realizar siempre ya que es un proceso tedioso que más que ayudar va a ralentizar los procesos, siendo así, podemos determinar a través de gráficas y métricas, la efectividad del modelo desarrollado.

Uno de estos gráficos es el histograma de residuos:



Sirve para verificar si los residuos tienen una distribución aproximadamente normal, como asume la regresión lineal. En nuestro caso y gracias a la validación de los datos residuales de nuestro modelo, podemos ver que efectivamente se visualiza una distribución normal esperada en un modelo correctamente entrenado y sus datos consistentes.



Por otro lado, contamos con el RMSE y R^2

```

# RMSE y R2
rmse <- sqrt(mean(residuals^2))
cat("RMSE del modelo:", rmse, "\n")

r_squared <- 1 - (sum(residuals^2) / sum((train_data$Precio - mean(train_data$Precio))^2))
cat("R2 del modelo:", r_squared, "\n")

```

RMSE del modelo: 316.789
R² del modelo: 0.9078237

El RMSE calculado, indica un error promedio relativamente bajo considerando el rango de los datos, que en sí son los precios y el error estimado del modelo es de aproximadamente \$ 315,789, y por otro lado nuestra métrica R^2 explica aproximadamente el 90.78% de la varianza de la variable dependiente, lo cual es bastante bueno.

Con esto podemos determinar que el modelo generado nos permitirá predecir con 3 datos, Grupo, Producto y Fuente el precio estimado de un alimento. Tener en consideración que si bien hacemos el ejemplo anteriormente ingresando los datos manualmente existen herramientas o recursos, como un código de barras, que pueden traer esta información y solo al leerlo podrán predecir el precio correspondiente sin un ingreso manual de los datos.

Este algoritmo funciona en dos etapas: La primera, ajuste basado en intervalos de tiempo de degradación conocidos y la segunda ajuste mediante sensores cuando la degradación es incierta o requiere seguimiento en tiempo real. Posteriormente el modelo revisa los productos, comparando la fecha de ingreso con el intervalo de tiempo de frescura almacenado en la base de datos. Si el intervalo de tiempo permite estimar con certeza el inicio de la degradación, el sistema ajusta el precio de manera progresiva para al final ser comparado con los precios con los de las bases de datos para ver si cumple con una proyección realista.

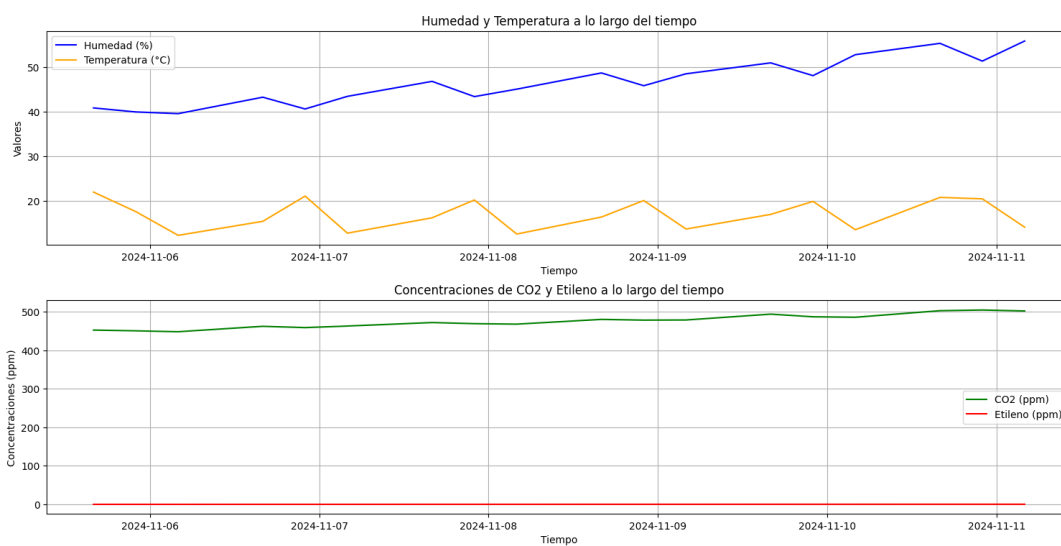


Prueba del sistema en entorno controlado

Validación del Sistema de monitoreo

En esta fase, el sistema fue implementado en un entorno de prueba controlado, donde se le encargó la medida de manzanas verdes. El objetivo fue validar la hipótesis mediante la ejecución del sistema de medición en tiempo real y evaluar la fiabilidad de los datos, así como establecer los umbrales de sensibilidad. Durante esta fase, se recopilaban métricas sobre la frescura de las manzanas durante 7 días. Algunos de los retos presentados fue la conexión de múltiples sensores al mismo tiempo, especialmente si los datos necesitan ser procesados en tiempo real y enviados al instante, esto genera saturación recibiendo y procesando de señales, lo que entorpece el microcontrolador y puede provocar fallos. Parte de este problema se pudo solucionar al incluir un condensador de desacoplamiento, reduciendo el ruido e interferencia. Por otro lado, se encontró redundancia en las señales de entrada, lo que afecta las lecturas entre sensores debido a la sensibilidad cruzada y en últimas hace más compleja la ya difícil tarea de calibración. Ante esto se descartó el uso del MQ7 y YL9, el MQ2 se encontró más preciso con la detección de monóxido de carbono y cuenta con más gases dentro de su curva de sensibilidad.

Figura 8: Datos censados por el sistema de medición



Fuente: Elaboración propia



Figura 9: Prototipo en etapa de muestreo



Finalmente, estos resultados muestran la evolución de la fruta a lo largo de una semana, si bien la muestra se encuentra en un lapso de maduración su variación no fue drástica y parte de los cambios en variables como temperatura, se pueden asociar a factores ambientales. Sin embargo, la tendencia en el aumento de CO₂ y de temperatura gradual reflejan una mayor actividad respiratoria entre las muestras, producto de su maduración.

Etapa	Etileno (ppm)	CO ₂ (ppm)	Temperatura (C)
Frescura	< 0.1	<600	±1 -1.2 con respecto a la temperatura ambiente
Maduración	0.1- 1	600-1200	+1 - 3 con respecto a la temperatura ambiente
Degradación	>1	>1200	+ 1 - 5 con respecto a la temperatura ambiente

Validación del Sistema de Predicción

Con el modelo ya entrenado anteriormente podemos hacer uso de este, por ejemplo y de manera



sencilla:

```
new_product <- data.frame(  
  Grupo = factor("PROCESADOS", levels = levels(precios_histor$Grupo)),  
  Fuente = factor("Sincelejo, Nuevo Mercado", levels =  
  levels(precios_histor$Fuente)),  
  Producto = factor("Salsa de tomate doy pack", levels =  
  levels(precios_histor$Producto))  
)
```

Se crea un nuevo producto especificando cada una de las variables que definirán su precio a través del modelo predictivo, e implementando este:

```
predicted_price <- predict(linear_model, new_product, type = "response")  
cat("Precio estimado para el nuevo producto:", round(predicted_price, 2), "\n")
```

Podemos visualizar que obtenemos el siguiente Output/salida:

Precio estimado para el nuevo producto: 12136.48

Y dado el análisis realizado con respecto a la veracidad del modelo, podemos tener certeza que este valor es una estimación real con respecto al histórico de precios y sus variables independientes.

Simulación de Escenarios: Realizar simulaciones en el sistema de monitoreo para verificar el ajuste de precios y el cambio de fases en diferentes condiciones de frescura y patrones de consumo.

Ahora bien, con el modelo desarrollado, y teniendo en cuenta los umbrales definidos anteriormente generamos las condicionales correspondientes en R y definimos la función `ajustar_precio <- funcion(precio, CO2, Humedad, Temperatura, Etileno)` que recibe el precio predicho con el modelo, y a través de los parámetros obtenidos con cada uno de los sensores podemos modificar, actualizar y tener una aproximación de precio más realista y dinámico con respecto al estado del producto. Determinado esto, tenemos la función definida:



```
# Definir función para ajustar el precio
ajustar_precio <- function(precio, CO2, Humedad, Temperatura, Etileno) {
  # Ajustar el precio según el CO2
  if (CO2 < 600) {
    # Frescura
    precio <- precio * 1.05 # Aumentar el precio un 5% por frescura
    estado <- "Fresca"
  } else if (CO2 >= 600 && CO2 < 1200) {
    # Maduración
    precio <- precio * 1.02 # Aumentar el precio un 2% por maduración
    estado <- "Maduración"
  } else if (CO2 >= 1200) {
    # Degradación
    precio <- precio * 0.95 # Reducir el precio un 5% por degradación
    estado <- "Degradada"
  }

  # Ajustar según humedad
  if (Humedad < 50) {
    precio <- precio * 0.98 # Reducir precio si humedad baja
  } else if (Humedad > 80) {
    precio <- precio * 0.97 # Reducir precio si humedad alta
  }

  # Ajustar según temperatura (por ejemplo, mayor temperatura puede acelerar la maduración)
  if (Temperatura > 30) {
    precio <- precio * 0.95 # Reducir precio si temperatura muy alta
  }

  # Ajustar según etileno (más etileno puede indicar más maduración)
  if (Etileno > 10) {
    precio <- precio * 0.98 # Reducir precio si el etileno es alto
  }

  # Devolver el precio ajustado y el estado
  return(list(precio_ajustado = precio, estado = estado))
}
```

Con las siguientes condiciones:

```
[65] new_product <- data.frame(
  Grupo = factor("PROCESADOS", levels = levels(precios_histor$Grupo)),
  Fuente = factor("Sincelejo, Nuevo Mercado", levels = levels(precios_histor$Fuente)),
  Producto = factor("Salsa de tomate doy pack", levels = levels(precios_histor$Producto))
)

[70] datos_nuevos <- data.frame(
  Humedad = 44, # Ejemplo de humedad en %
  Temperatura = 25, # Ejemplo de temperatura en °C
  CO2 = 800, # Ejemplo de CO2 en ppm
  Etileno = 0.14 # Ejemplo de etileno en µL/L
)
```



Obtenemos un precio predicho de aproximadamente de \$ 12.136,48 esto como valor inicial, más sin embargo a la hora de ejecutar la función con las condiciones propuestas tenemos que el valor actualizado al estado de la fruta es de \$ 12,131.63 teniendo en cuenta que su estado es de Maduración sin un cambio drástico en su composición y en un estado más bien óptimo. Si hacemos una prueba, pero ahora buscando que el producto esté en descomposición de la siguiente manera

```
[75] datos_nuevos <- data.frame(  
  Humedad = 22,      # Ejemplo de humedad en %  
  Temperatura = 35, # Ejemplo de temperatura en °C  
  CO2 = 1222,       # Ejemplo de CO2 en ppm  
  Etileno = 11      # Ejemplo de etileno en µL/L  
)
```

Obtenemos que dada las condicionales planteadas buscando la simulación de degradación, obtenemos un valor aproximado y actualizado de \$ 10.519,43.

```
"Precio ajustado: 10519.4314306463"  
"Estado de la fruta: Degradada"
```

Con esto vemos el dinamismo del código creado, viendo cómo un mismo producto bajo condiciones de estado diferentes se adecuan y el precio cambia con respecto a esto, mostrando la funcionalidad en si del sistema y su dinamismo.

Se hace la salvedad que los datos son leídos además de manera automática a través de los sensores y sistema físico, y buscar con esto la actualización en tiempo real del precio teniendo en cuenta sus condiciones reales.



11. Análisis de costos

Los costos directos corresponden a los elementos esenciales que intervienen en la fabricación y desarrollo del sistema y están directamente relacionados con el volumen de producción. Para el caso del proyecto actual, estos incluyen la adquisición e instalación de sensores especializados, como los sensores de gases MQ2, así como el sensor de temperatura y humedad DHT11. Además, los costos directos abarcan el desarrollo de un modelo predictivo basado en el historial de precios que permite ajustar automáticamente el valor de los productos en función de sus condiciones de conservación. También se incluye aquí el diseño de una interfaz gráfica de usuario (GUI), la cual visualiza los datos en tiempo real y facilita la interacción con el sistema. Estos costos variables fluctúan de acuerdo con la cantidad de productos que se fabriquen o instalen, lo que permite ajustar la inversión inicial según las necesidades del proyecto.

En contraste, los costos fijos se refieren a los elementos de infraestructura del sistema que permanecen constantes, independientemente del volumen de producción nivel de actividad. Estos engloban los sensores de planta que se integrarán en los sistemas de almacenamiento y refrigeración de los establecimientos comerciales, además de la red de comunicación necesaria para conectar estos sensores al sistema de monitoreo central. Asimismo, se contempla la adquisición de microcontroladores (Raspberry, por ejemplo), los cuales procesan y envían los datos recopilados por los sensores. Este conjunto de costos fijos asegura la operatividad continua del sistema y facilita la recolección de datos de manera precisa y confiable.

Por otra parte, el proyecto también considera los gastos generales, que abarcan aquellos costos indirectos



necesarios para el funcionamiento y mantenimiento del sistema. Estos incluyen actividades de reemplazo y calibración de sensores, las cuales se realizarán semestralmente para garantizar la precisión de las mediciones. Asimismo, se estima el consumo anual de energía eléctrica asociado a la operación de los dispositivos y sistemas de monitoreo. Estos gastos generales constituyen una parte esencial de los recursos requeridos para el mantenimiento adecuado del sistema y su eficiencia operativa a lo largo del tiempo.

Para la implementación y despliegue del sistema se requiere, además, una inversión inicial. Esta inversión incluye una serie de costos directos de adquisición, instalación y configuración de los equipos y materiales, como sensores, microcontroladores y otros dispositivos necesarios. El análisis de costos considera, también, la necesidad de un capital de trabajo, que se destinará a cubrir los gastos operativos iniciales de la empresa, como la compra de materias primas y el pago de salarios, hasta que el sistema comience a generar ingresos sostenibles.

Este esquema de costos permite estimar de manera detallada el margen de rentabilidad del negocio, considerando tanto los costos de producción como la inversión inicial. El proyecto se presenta como una alternativa económica viable para el sector minorista, ya que optimiza la gestión de inventarios mediante la implementación de sensores y un algoritmo predictivo, lo que contribuye a la reducción del desperdicio de alimentos perecederos. Con este modelo de costos, el sistema no solo representa una solución tecnológica innovadora, sino que también promete un retorno de inversión atractivo para los posibles inversionistas, quienes podrán valorar la rentabilidad y el impacto económico positivo del proyecto en el ámbito comercial.

TIPO DE GASTO	GASTO	VALOR	DESCRIPCIÓN
Gasto Fijo	Sensores de planta	\$50,000 -	El costo de los sensores varía según la precisión y el rango de medición, sensores que ofrecen mayor exactitud o que pueden medir en rangos más amplios requieren tecnologías más



		\$200,000 COP.	sofisticadas y procesos de calibración más precisos. Así mismo, la sensibilidad del sensor influye considerablemente en el precio, ya que sensores con una mayor sensibilidad son capaces de detectar niveles más bajos de una variable.
	Red de Comunicación	\$0 – \$40.000 COP.	El costo de la infraestructura de comunicación entre sensores y computadoras dependerá del tipo de comunicación es uno de los factores más influyentes: tecnologías como el Wi-Fi, Bluetooth, Zigbee o LoRa, cada una con sus propias características en términos de velocidad, alcance y consumo energético, tienen diferentes costos asociados. Si los dispositivos ya tienen módulos integrados se puede obviar.
	Micro controladores	\$32.500- \$620.000 COP.	Un microcontrolador con una mayor frecuencia de reloj, memoria RAM e interfaces de comunicación será más costoso. La opción más barata con estas características es el ESP32, pero puede aplicarse otros de mayor capacidad de cómputo como la tarjeta Raspberry Pi.
Gasto Directo	Desarrollo del algoritmo de precios	\$ 4'000,000 - \$ 7'000,000	Se basa en el análisis del tiempo, recursos técnicos y humanos requeridos para su programación y calibración. Este cálculo toma en cuenta las horas de desarrollo y pruebas, el costo por hora, así como las herramientas de desarrollo necesarias. La estimación de precio se obtiene multiplicando las horas de trabajo proyectadas por el costo por hora, sumando los gastos de herramientas.
	Software de control para usuarios	\$ 5'000,000 - \$ 10'000,000	Se determina tomando en cuenta el tiempo requerido para diseñar la interfaz gráfica, programar sus funcionalidades, y realizar pruebas de uso. El costo estimado se basa en el número de horas de diseño y programación proyectadas, aplicando igualmente un costo de referencia por hora de desarrollo. Este cálculo permite reflejar el esfuerzo invertido en el desarrollo de la interfaz de usuario y en asegurar su operatividad, incluyendo pruebas y ajustes necesarios.
Gasto General	Reemplazo y Calibración de Sensores (semestral)	\$50,000 - \$200,000 COP	La calibración de un sensor requiere equipos especializados y personal capacitado para garantizar que las mediciones sean precisas.
	Gastos de energía eléctrica (Anual)	\$343,280 COP	Basado en el consumo del sistema de monitoreo y microcontrolador, el envío de datos y computadora para una tienda con tarifa de 739 Kw/h.

El proyecto requiere una inversión inicial considerable para instalar sensores, infraestructura de datos y



desarrollo del software predictivo. Sin embargo, puede amortizarse en el tiempo con la reducción de pérdidas de alimentos perecederos y la optimización en la rotación de inventario, proporcionando ahorros significativos a largo plazo en desperdicio y costos operativos.

12. Conclusiones

Inicialmente, el sistema de control de precios dinámico implementado para alimentos perecederos



fue evaluado con diversos productos (Frutas y verduras), por ejemplo, mediante el monitoreo en tiempo real de dos manzanas con diferentes condiciones de frescura. Durante las mediciones, se analizaron las siguientes variables: temperatura, humedad y concentración de gas CO₂, las cuales afectan directamente el estado y vida útil de los productos.

En el caso de la manzana fresca, con una temperatura ambiente de 15°C, humedad de 20% y una concentración de gas CO₂ de 0.055 PPM, el precio se mantuvo en \$9,990 por kilogramo (Precio más reciente del valor por kilogramo de manzana, datos DANE 2024), el valor estimado para el día de la medición, dado que las condiciones de la fruta no indicaban signos de deterioro.

Por otro lado, la manzana en descomposición, con una temperatura de 18°C, humedad de 26% y concentración de CO₂ de 0.043 PPM, presentó una reducción en el precio, ajustándose a aproximadamente \$7,800 por kilogramo. Esta disminución en el precio fue el resultado del análisis de las variables de la fruta en combinación con el historial de precios, lo que permitió una evaluación dinámica y precisa de la condición del producto. Este comportamiento evidencia cómo el sistema ajusta los precios de manera efectiva, respondiendo a las condiciones del producto y optimizando la gestión de inventarios para mejorar tanto la rentabilidad como la sostenibilidad en la distribución de alimentos perecederos.

Con esto vemos que a lo largo de esta investigación se logró abordar la necesidad de un sistema de control de precios dinámico en un entorno controlado y diseñado por el equipo de desarrollo, permitiendo avances hacia la optimización de la gestión de precios e inventarios mediante tecnologías emergentes. Se cumplieron los objetivos propuestos, destacando la viabilidad de implementar un sistema basado en sensores y algoritmos predictivos para responder de manera eficiente a las condiciones de los productos perecederos en tiempo real. Tras evaluar los sistemas de control de precios y gestión de inventarios empleados actualmente en Colombia, se evidenció una necesidad de actualización tecnológica. Los métodos tradicionales presentan limitaciones que afectan la eficiencia y rentabilidad, y los métodos



emergentes ofrecen mejoras notables en precisión y respuesta, especialmente al incorporar sensores y tecnologías de monitoreo.

El algoritmo desarrollado permite ajustar los precios de los alimentos perecederos de forma dinámica, basándose en patrones de consumo y condiciones de inventario. Este enfoque predictivo contribuye a reducir desperdicios y maximizar ingresos al ajustar los precios en función del tiempo de vida del producto y su demanda, mostrando potencial para mejorar la sostenibilidad y rentabilidad en el sector minorista. El prototipo de monitoreo en tiempo real, basado en sensores, demostró ser efectivo para detectar cambios en las condiciones de los alimentos perecederos, como temperatura, emisión de gases y humedad, variables cruciales para evitar el desperdicio alimentario y a su vez la pérdida económica que trae consigo esto. Esto permite una mejor toma de decisiones en la gestión de inventarios y precios, alineándose con los principios de eficiencia y frescura en la distribución alimentaria.

13. Bibliografía

Gaviria, S. & Departamento de Planeación. (2016). PÉRDIDA Y DESPERDICIO DE ALIMENTOS EN COLOMBIA. En Departamento Nacional de Planeación. Recuperado 13 de septiembre de 2024, de



<https://colaboracion.dnp.gov.co/CDT/Prensa/Publicaciones/P%C3%A9rdida%20y%20desperdicio%20de%20alimentos%20en%20colombia.pdf>

Hambre en Colombia, entre la realidad y la contienda electoral. (2022, 11 febrero). Universidad Externado de Colombia. <https://www.uexternado.edu.co/economia/hambre-en-colombia-entre-la-realidad-y-la-contienda-electoral/>

Colonius, F., Helmke, U., Jordan, J., Kawan, C., Sailer, R., & Wirth, F. (2014). Analysis of Networked Systems. En Springer eBooks (pp. 31-79). https://doi.org/10.1007/978-3-319-01131-8_2

Pineda Pertuz, C. M. (2021). Aprendizaje automático y profundo en python: una mirada hacia la inteligencia artificial. Ediciones de la U. <https://www-ebooks7-24-com.bdbiblioteca.universidadean.edu.co/?il=18081>

Mora Fallas H. (2017). “Diseño de un dispositivo electrónico para medir el daño mecánico en melón haciendo uso de tecnologías de código abierto. Universidad de Costa Rica

Secretaría de Desarrollo Económico de Bogotá. (2022). La Secretaría de Desarrollo Económico seleccionó las mejores iniciativas que fomentan la reducción de pérdidas y desperdicios de alimentos en Bogotá. <https://desarrolloeconomico.gov.co/la-secretaria-de-desarrollo-economico-selecciono-las-mejores-iniciativas-que-fomentan-la-reduccion-de-perdidas-y-desperdicios-de-alimentos-en-bogota/>

Secretaría Distrital de Planeación de Bogotá. (2020). Distrito promueve acciones para evitar el desperdicio de alimentos. <https://www.sdp.gov.co/noticias/distrito-promueve-acciones-evitar-desperdicio-de-alimentos>

BBC News Mundo. (2021). El desperdicio de alimentos: ¿cómo es posible que se tire el 17% de la comida en el mundo y al mismo tiempo haya 690 millones de personas que pasan hambre? <https://www.bbc.com/mundo/noticias-56322961>

Sebesta, R. W. (2016). Concepts of programming languages (11th ed.). Pearson.



Fielding, R. T., & Taylor, R. N. (2000). Architectural styles and the design of network-based software architectures (Doctoral dissertation, University of California, Irvine).

Connolly, T., & Begg, C. (2015). Database systems: A practical approach to design, implementation, and management (6th ed.). Pearson.

Tanenbaum, A. S., & Wetherall, D. J. (2010). Computer networks (5th ed.). Pearson.

Bishop, C. M. (2016). Pattern recognition and machine learning (2nd ed.). Springer.

Laudon, K. C., & Laudon, J. P. (2016). Management Information Systems: Managing the Digital Firm (14th ed.). Pearson.

Render, B., Stair, R. M., & Hanna, M. E. (2017). Quantitative Analysis for Management (12th ed.). Pearson.

El Tiempo. (2022). Bogotá: al día se desecha comida que alimentaría a más de 900.000 personas. <https://www.eltiempo.com/bogota/bogota-al-dia-se-desecha-comida-que-alimentaria-a-mas-de-900-000-personas-672350>

Wang, W., Yang, W., Liu, Y., Wang, Z., & Yan, Z. (2022). A Research of Neural Network Optimization Technology for Apple Freshness Recognition Based on Gas Sensor Array. *Scientific Programming*, 2022, 1-11. <https://doi.org/10.1155/2022/5861326>

Departamento Administrativo Nacional de Estadística (DANE). (2023). Comportamiento de la economía: IV trimestre de 2023 [Informe]. <https://www.dane.gov.co/files/operaciones/EEVV/cp-EEVV-IVtrim2023.pdf>

Departamento Administrativo Nacional de Estadística - DANE. (2022). Inseguridad alimentaria en Colombia. Nota estadística. Bogotá: DANE. <https://www.dane.gov.co/files/investigaciones/notas-estadisticas/NotaEstadistica-FIES-DANE-FAO.pdf>

Departamento Nacional de Planeación (DNP). (2016). Pérdida y desperdicio de alimentos en Colombia. <https://colaboracion.dnp.gov.co/CDT/Prensa/Publicaciones/P%C3%A9rdida%20y%20desperdicio%20de>



%20alimentos%20en%20colombia.pdf

Bean, R. (2024). How Kroger is using data and AI to drive innovation in the grocery industry. *Forbes*. <https://www.forbes.com/sites/randybean/2024/08/26/how-kroger-is-using-data-ai-to-drive-innovation-in-the-grocery-industry/>

Ministerio de Ambiente y Desarrollo Sostenible. (2022). *Residuos de aparato eléctricos y electrónicos - RAEE* -. <https://www.minambiente.gov.co/asuntos-ambientales-sectorial-y-urbana/residuos-de-aparato-electricos-y-electronicos-raee/>

Perez-Rodriguez, F. (Ed.). (2019). *Predictive modelling in food* (1st ed.). Cambridge Scholars Publishing.

Mathotaarachchi, K. V., Hasan, R., & Mahmood, S. (2024). Advanced Machine Learning Techniques for Predictive Modeling of Property Prices. *Information*, 15(6), 295-.
<https://doi.org/10.3390/info15060295>

De, B. (2023). Introduction to APIs. In: *API Management*. Apress, Berkeley, CA.
https://doi.org/10.1007/979-8-8688-0054-2_1

Toscano, R. (2024). Machine Learning. In: *Solving Optimization Problems with the Heuristic Kalman Algorithm*. Springer Optimization and Its Applications, vol 212. Springer, Cham.
https://doi.org/10.1007/978-3-031-52459-2_7

A. Varghese, E. Lacap, Jr., I. Sajath, M. Kumar, & S. Kolamban (2023). *Introduction to Machine Learning*. IGI Global. <https://doi.org/10.4018/978-1-7998-8343-2.ch008>

Heizer, J., Render, B., Munson, C., Zamora, B. E., De Mendieta, H. A., Román, M. L., & Morales



Castro, A. (2021). Principios de administración de operaciones (1a edición.). Pearson.

Herrera Ramírez, M. M., & Orjuela Castro, J. A. (2014). Perspectiva de trazabilidad en la cadena de suministros de frutas: un enfoque desde la dinámica de sistemas. *Ingeniería (0121-750X)*, 19(2), 63–84.

Espejo González, M. (2022). Gestión de inventarios: métodos cuantitativos: (1 ed.). Marge Books. <https://elibro-net.bdbiblioteca.universidadean.edu.co/es/ereader/bibliotecaean/217920?page=64>

Hena-Baena, C. A., Zuluaga-Zuluaga, B., Galeano-Castro, J., Marín-García, E. J., & Calvo-Salcedo, A. F. (2024). Methodology for Inventory Management in Neighborhood Stores Using Machine Learning and Integer Linear Programming. *Ingeniería (Bogotá, Colombia : 1993)*, 29(1), e19423–e19423. <https://doi.org/10.14483/23448393.19423>

Pantano, E., Timmermans, H., & Clodfelter, R. (2020). *Point-of-sale technologies at retail stores*. In E. Pantano (Ed.), *Technology and Innovation for Marketing* (pp. 1-16). IGI Global. <https://doi.org/10.4018/978-1-60960-738-8.ch001>

Jaycon-Dev. (2023, 28 septiembre). *Gas Sensor Technology Unveiled: What You Need to Know*. Jaycon. <https://www.jaycon.com/understanding-a-gas-sensor/#h-results>

HANWEI ELETRONICS CO. (s. f.). TECHNICAL DATA MQ-2 GAS SENSOR. En *HWSensor* (N.º MQ2). Recuperado 14 de noviembre de 2024, de <https://www.mouser.com/datasheet/2/321/605-00008-MQ-2-Datasheet-370464.pdf>

Datasheet

