Pontificia Universidad Católica del Ecuador

Facultad de Ingeniería

TEMA:

Desarrollo de un modelo predictivo basado en Machine Learning para anticipar emergencias por especialidad y estaciones de Bomberos del Distrito Metropolitano de Quito

AUTOR:

MANUEL EDUARDO MOINA CAMPOS

TUTOR:

MSc. DAMIAN ANIBAL NICOLALDE RODRIGUEZ

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE MAGISTER EN SISTEMAS DE INFORMACIÓN MENCIÓN DATA SCIENCE

Quito, noviembre 2024

DEDICATORIA

Este trabajo de titulación está dedicado a mi padre, Manuel Moina, un gran hombre que me enseño el valor del trabajo duro, la perseverancia, la importancia de no rendirme en los momentos difíciles, sin duda gran parte de lo que soy se lo debo a él, su recuerdo me da la confianza que necesito para seguir cumpliendo todas las metas que me proponga.

“Estoy seguro de que desde donde estés, te sientes orgulloso de este logro”

AGRADECIMIENTO

Quiero expresar mi más profundo agradecimiento a todas las personas que han contribuido directa o indirectamente a concluir este proyecto. En especial a mis hijas Valentina, Milena a mi mamá Luz Victoria Campos, quienes sin saberlo me motivan y me dan la fuerza que necesito, sin ustedes este proceso hubiera sido mucho más difícil. De igual manera agradezco a mi tutor de tesis por su tiempo, guía, paciencia y dedicación durante este proceso de titulación. Me siento afortunado por haber contado con su apoyo.

RESUMEN

Este trabajo presenta el desarrollo de un modelo predictivo basado en técnicas de Machine Learning cuyo objetivo es anticipar la cantidad de emergencias atendidas por Bomberos del Distrito Metropolitano de Quito. La motivación principal radica en la complejidad en la diversidad de especialidades de emergencias que atiende Bomberos: incendios, rescates, materiales peligrosos, atención prehospitalaria y eventos hídricos y meteorológicos, los cuales se distribuyen de manera desigual en el territorio y demandan una respuesta efectiva.

Para abordar este problema, se recopilaron y analizaron datos históricos de los años 2022, 2023 y 2024, procedentes de los registros oficiales de Bomberos del Distrito Metropolitano de Quito. Estos datos incluyen la fecha y hora de la emergencia, la estación que brindó la atención, la especialidad de la emergencia entre otras variables relevantes. Previo al modelado, se realizó un proceso de limpieza de datos, eliminando columnas que no aportaban valor a la predicción y transformando variables categóricas para que pudieran procesarse correctamente. Por otra parte, se integraron nuevas variables como el día de la semana, se agruparon las emergencias por especialidad, fechay día del més y se agregó la variable número de emergencias con el fin de capturar posibles patrones temporales.

Como parte de la metodología se integró CRISP-DM, partiendo de la comprensión del negocio continuando con la preparación de datos y finalizando con la validación y evaluación del modelo. El algoritmo escogido fue Random Forest,

seleccionado por su robustez ante valores atípicos, su capacidad de manejar variables heterogéneas y su facilidad para capturar relaciones no lineales.

Los resultados obtenidos evidencian que el modelo es capaz de predecir con un error moderado la frecuencia de emergencias en las distintas estaciones, identificando tendencias con cierto grado de confiabilidad. Sin embargo, aún se observan limitaciones en escenarios puntuales como fines de semana largos debido a feriados o eventos meteorológicos extremos. Estas situaciones exigen variables adicionales como datos climáticos.

En términos de aplicación práctica, los hallazgos podrían permitir a Bomberos preparar con mayor anticipación la asignación de recursos operativos, reduciendo los tiempos de respuesta y, contribuyendo en la misión institucional de Salvar Vidas y Proteger Bienes.

ABSTRACT

This work presents the development of a predictive model based on Machine Learning techniques aimed at forecasting the number of emergencies handled by the Fire Department of the Metropolitan District of Quito. The main motivation stems from the complexity and diversity of the emergencies they address—fires, rescues, hazardous materials, pre-hospital care, and hydrometeorological events—which are unevenly distributed and require an effective response.

To tackle this issue, historical data from 2022, 2023, and 2024 were collected and analyzed using official records from the Fire Department of the Metropolitan District of Quito. This dataset includes the date and time of the emergency, the station providing assistance, the emergency specialty, among other relevant variables. Before modeling, a data-cleaning process was conducted, removing columns that did not contribute to the prediction and converting categorical variables into a workable format. Additionally, new variables were integrated—such as the day of the week—and the emergencies were grouped by specialty, date, and day of the month. A “number of emergencies” variable was also introduced to capture potential temporal patterns.

As part of the methodology, CRISP-DM was applied, starting with a thorough understanding of the business, continuing with data preparation, and concluding with model validation and evaluation. Random Forest was the chosen algorithm, selected for its robustness against outliers, its ability to handle heterogeneous variables, and its ease in capturing non-linear relationships.

The results indicate that the model can predict, with moderate error, the frequency of emergencies at various stations, reliably identifying trends. However, certain limitations remain in specific scenarios—such as long weekends due to public holidays or extreme weather events—which call for additional variables like climate data.

In practical terms, these findings could allow the Fire Department to prepare operational resources well in advance, reducing response times and contributing to the institutional mission of saving lives and protecting property.

Contenido

ÍNDICE DE FIGURAS

CAPÍTULO I: INTRODUCCIÓN

1. Introducción

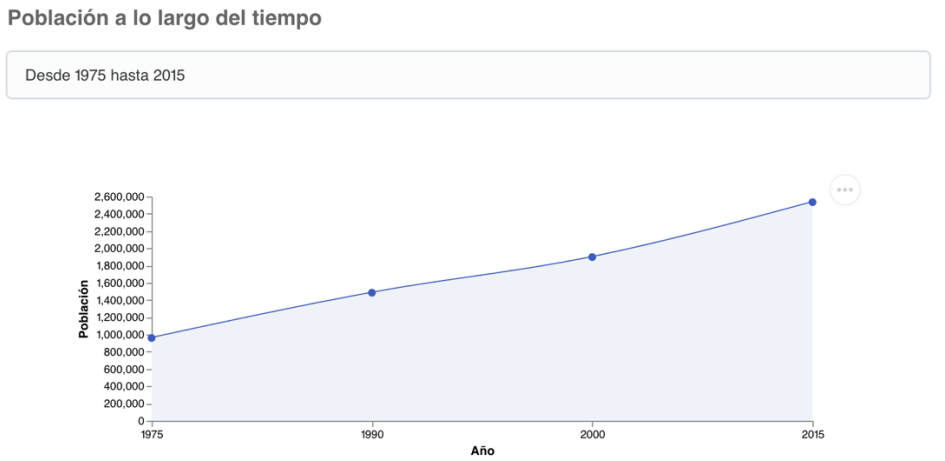
1.1 Planteamiento del problema

La gestión de emergencias en el Distrito Metropolitano de Quito enfrenta múltiples desafíos. Uno de ellos es la diversidad de eventos que atiende el Cuerpo de Bomberos: incendios, rescates, manejo de materiales peligrosos, atención prehospitalaria y respuesta a eventos hídricos y meteorológicos. Estos eventos varían significativamente en su frecuencia y se distribuyen de manera desigual a lo largo del territorio. Esta situación dificulta la planificación eficiente y la respuesta oportuna. Además, la geografía compleja del Distrito Metropolitano, con zonas urbanas densas, áreas rurales y regiones montañosas, incrementa los retos logísticos y operativos, agregando aún más complejidad a la gestión de emergencias.

En la actualidad, el Cuerpo de Bomberos de Quito se basa principalmente en análisis retrospectivos y en la experiencia acumulada para planificar y asignar recursos. Este enfoque limita la capacidad de anticipar demandas futuras con precisión. Como resultado, la asignación de recursos no siempre coincide con las necesidades reales de cada estación. Esto afecta la eficiencia operativa y los tiempos de respuesta ante situaciones inesperadas. La falta de herramientas predictivas que proporcionen

información anticipada sobre el número y tipo de emergencias por estación y especialidad impide diseñar estrategias proactivas de preparación y mitigación.

Este problema se agrava debido a factores externos. Uno de ellos es el crecimiento poblacional acelerado y los efectos del cambio climático que se traducen en variaciones considerables de temperatura, incrementos de lluvias intensas y periodos de sequía lo cual agrega más incertidumbre. Ante esta coyuntura, surge la necesidad de herramientas predictivas que, basadas en datos históricos, permitan anticipar el número y especialidad de emergencias por estación de bomberos para mejorar la planificación y reducir tiempos de respuesta, a continuación, se presenta la Figura 1 que muestra el crecimiento poblacional del DMQ.



Datos	1975	1990	2000	2015
Población	959 529	1 484 407	1 897 523	2 534 575

Figura 1 Crecimiento poblacional del Distrito Metropolitano de Quito desde 1975 hasta 2015, (City-Facts. (n.d.))

El desarrollo urbano desordenado y el cambio climático también han influido en el incremento de la frecuencia y complejidad de ciertas emergencias. El cambio

climático ha generado un aumento en la temperatura promedio de entre 1.2°C y 1.4°C entre 1981 y 1999, intensificando el efecto de isla de calor urbana debido a la expansión urbana y la reducción de áreas verdes (Venegas Zapata, J. G. (2020)). Además, se han registrado eventos climáticos extremos, como lluvias intensas y sequías, junto con la pérdida de 83,809 hectáreas de cobertura vegetal en 23 años (Venegas Zapata, J. G. (2020)). Estos fenómenos elevan el riesgo de inundaciones, escasez de agua y la aparición de temporadas secas propensas a incendios forestales. Las lluvias intensas también provocan inundaciones y deslizamientos. Sin herramientas que permitan anticipar estos eventos, las estaciones de bomberos no cuentan con una preparación óptima para gestionar los recursos necesarios durante las emergencias.

En este contexto, se hace necesaria la creación de un modelo predictivo basado en técnicas de Machine Learning. Este modelo permitirá anticipar el número de emergencias por especialidad y estación. Además, ayudará a identificar patrones en la frecuencia de las emergencias, las especialidades involucradas y las estaciones afectadas. De esta forma, el Cuerpo de Bomberos pasará de un enfoque operativo predominantemente reactivo a uno proactivo. Con ello, será posible planificar y distribuir los recursos con mayor eficiencia, mejorar la preparación operativa y disminuir los tiempos de respuesta. Este avance repercutirá directamente en la seguridad y el bienestar de la población.

Por último, la carencia de un enfoque predictivo no es exclusiva del Distrito Metropolitano de Quito. Muchas ciudades metropolitanas enfrentan dificultades similares en la gestión de emergencias. Por tanto, el desarrollo de este modelo no solo

beneficiará a Quito, sino que también podría convertirse en una referencia valiosa para otras ciudades con características similares. De esta manera, otras urbes interesadas en optimizar sus estrategias de respuesta podrán adoptar el modelo propuesto y fortalecer su capacidad de preparación y reacción ante emergencias.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un modelo predictivo basado en Machine Learning para anticipar el número de emergencias según la especialidad y estaciones de Bomberos del Distrito Metropolitano de Quito, con el fin de que los resultados obtenidos del modelo aporten en la planificación de la gestión de emergencias reduciendo tiempos de respuesta mediante una asignación más efectiva de recursos operativos.

1.2.2 Objetivos Específicos

1. Aplicar una metodología para desarrollo de proyectos de Ciencia de Datos que asegure un enfoque sistemático y organizado y permita cumplir con el ciclo de vida del proyecto
2. Recopilar, analizar y procesar datos históricos de emergencias atendidas por estaciones de Bomberos del Distrito Metropolitano de Quito.
3. Implementar técnicas de análisis exploratorio para identificar patrones en la ocurrencia de emergencias

4. Diseñar y entrenar un modelo predictivo basado en Machine Learning que pueda anticipar la cantidad de emergencias por especialidad y estación de Bomberos en el Distrito Metropolitano.
5. Evaluar el desempeño del modelo con datos reales para determinar la precisión y confiabilidad de los resultados obtenidos

1.3 Justificación

La capacidad de anticipar con un alto grado de precisión el número de posible de emergencias es una necesidad que trasciende fronteras locales y nacionales, ya que permite una gestión más efectiva en la atención de emergencias en cualquier contexto urbano o metropolitano. A nivel local, el Distrito Metropolitano de Quito enfrenta esta misma necesidad debido a las características particulares de su entorno. Por ello, se propone la construcción de un modelo predictivo que no solo identifique el número de posibles emergencias por especialidad y estación de bomberos, sino que también transforme el enfoque operativo de uno predominantemente reactivo a uno proactivo y basado en la preparación anticipada. Este cambio permitirá mejorar significativamente la capacidad de respuesta operativa y optimizar los recursos disponibles.

El Distrito Metropolitano de Quito se encuentra en un entorno complejo en cuanto a la gestión de emergencias. Su diversidad geográfica, que incluye zonas urbanas densamente pobladas, áreas rurales y zonas montañosas, crea un desafío logístico y operativo para la distribución y atención de emergencias. A esto se suma la creciente densidad poblacional y la variabilidad de tipos de emergencias que enfrentan

las estaciones de bomberos en la ciudad. Estas emergencias abarcan diversas especialidades, como:

- Incendios
- Rescates
- Manejo de materiales peligrosos
- Atención prehospitalaria
- Eventos hídricos y meteorológicos

Estas emergencias no solo ocurren de forma desigual en términos geográficos, sino que también presentan variaciones estacionales y temporales. Esto exige un análisis detallado que permita anticipar los eventos según su especialidad, frecuencia y estación de bomberos.

Actualmente no existen herramientas predictivas específicas para el Distrito metropolitano de Quito que tome en cuentas sus particularidades y que permitan proyectar con un alto grado de precisión el número y especialidad de emergencias esperadas en cada estación de Bomberos.

Este proyecto busca desarrollar un modelo predictivo basado en Machine Learning, que permita anticipar la cantidad de emergencias por especialidad y estación de Bomberos, los resultados encontrados pueden servir como insumo para fortalecer la planificación y la gestión operativa del Cuerpo de Bomberos del Distrito Metropolitano de Quito. Aunque el enfoque principal del proyecto no es la asignación directa de recursos, los resultados del modelo servirán como insumo estratégico para optimizar el

uso de personal, vehículos, equipos y herramientas y toda la gestión logística operativa necesaria en la atención de emergencias.

Para la construcción del modelo, se ha seleccionado el algoritmo Random Forest. Esta selección técnica se fundamenta en las capacidades del algoritmo para manejar datos heterogéneos y complejos, su resiliencia ante el sobreajuste y su robustez frente a valores atípicos o incompletos. El Modelo Random Forest al combinar múltiples árboles de decisión, puede capturar relaciones no lineales entre variables de diversa índole como especialidades, subespecialidad de emergencia, días de la semana, estación que atendió la emergencia entre otras, lo que se adapta a la naturaleza de las emergencias en el Distrito Metropolitano de Quito. Además, su capacidad para generar una medida de importancia de variables resulta valiosa, pues facilita la identificación de los factores más influyentes en la ocurrencia de diferentes tipos de emergencias y contribuye a una toma de decisiones más informada. En comparación con otros métodos de Machine Learning, Random Forest presenta un equilibrio sólido entre precisión, capacidad de generalización e interpretabilidad relativa, lo que lo convierte en una opción confiable para la predicción y análisis en entornos con alta variabilidad e incertidumbre.

El modelo desarrollado para el Distrito Metropolitano de Quito podría servir como referencia o aplicarse en otras ciudades que compartan condiciones similares. Estas condiciones incluyen una amplia variedad de emergencias a atender como: incendios, rescates, manejo de materiales peligrosos, atención prehospitalaria y eventos hidrometeorológico dentro de entornos geográficos y climáticos complejos, con áreas

urbanas densamente pobladas, zonas rurales y regiones de difícil acceso. Además, la ciudad debe mostrar un crecimiento urbano acelerado, estar expuesta a efectos visibles del cambio climático y presentar una distribución desigual en la ocurrencia de emergencias. Asimismo, es necesario contar con un sistema organizado de estaciones de bomberos, registros históricos de datos suficientes para el análisis, infraestructura tecnológica adecuada y la capacidad de enfrentar retos logísticos comparables. Finalmente, se requiere un enfoque innovador que favorezca la implementación de nuevas tecnologías.

1.4 Alcance

El presente trabajo de titulación se centra en el desarrollo de un modelo predictivo basado en técnicas de Machine Learning para anticipar emergencias por especialidad y estación de Bomberos en el Distrito Metropolitano de Quito. Este alcance incluye el análisis y uso de datos históricos proporcionados por el CB-DMQ, clasificados en especialidades como incendios, rescates, manejo de materiales peligrosos, emergencias prehospitalarias y eventos hídricos y meteorológicos. La investigación abarca las 25 estaciones de Bomberos de Quito.

El modelo predictivo estará diseñado para identificar tendencias y anticipar el número de emergencias en cada especialidad y estación de Bomberos. A través del algoritmo Random Forest se busca procesar grandes volúmenes de datos históricos y validar su precisión mediante técnicas como validación cruzada y métricas como MAE (Mean Absolute Error) y RMSE (Root Mean Squared Error). La metodología CRISP-DM

guiará el desarrollo del modelo, garantizando un enfoque sistemático desde la comprensión del problema hasta la validación y evaluación de resultados.

En el proceso de preparación de los datos, se abordará el tratamiento de información incompleta o inconsistente mediante técnicas de limpieza y depuración de la base de datos, así como estrategias de imputación de valores faltantes que aseguren la coherencia y relevancia de las variables. Esto incluirá la detección y corrección de valores anómalos, eliminación o relleno de datos faltantes utilizando métodos estadísticos o modelos de imputación y la normalización de formatos para garantizar la uniformidad del conjunto de datos. Dichas medidas permitirán contar con una base de información robusta, minimizando el sesgo y los errores asociados al entrenamiento del modelo predictivo (Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2020)).

Aunque no se contempla la implementación del modelo en el Cuerpo de Bomberos del Distrito Metropolitano de Quito, los resultados se presentarán como insumos estratégicos para fortalecer la planificación operativa. Estos insumos permitirán optimizar la asignación de recursos como personal, vehículos, equipos y herramientas, contribuyendo a una gestión proactiva en la atención de emergencias. Sin embargo, la falta de una implementación en un entorno de producción limita la capacidad de evaluar el desempeño del modelo bajo condiciones reales, controlar la evolución del rendimiento a lo largo del tiempo y gestionar factores externos como la deriva de datos o el cambio en patrones estacionales. Además, no se podrán considerar aspectos propios de un sistema productivo tales como la escalabilidad, la latencia en la respuesta y la integración con sistemas internos y externos (Ashmore, R.,

Calinescu, R., & Paterson, C. (2019)). En consecuencia, el alcance se limita al desarrollo y validación del modelo en entornos simulados, sin la posibilidad de observar su impacto directo en la toma de decisiones operativas a gran escala.

En cuanto a su aplicabilidad, esta investigación no solo tiene importancia a nivel local, sino que también podría ser implementada en otras ciudades con características similares. Al establecer los fundamentos para una gestión de emergencias proactiva, este proyecto pretende posicionar al CB-DMQ como una institución pionera en el uso de herramientas tecnológicas avanzadas para la toma de decisiones informadas.

CAPÍTULO II: MARCO TEÓRICO Y CONCEPTUAL

2. Marco Teórico y Conceptual

2.1 Marco Teórico

Para garantizar que el presente trabajo de titulación de Ciencia de Datos sea eficiente, estructurado y enfocado en resultados prácticos será desarrollado utilizando la metodología CRISP-DM, el enfoque iterativo, flexible y orientado al negocio permitirá resolver problemas complejos de manera efectiva y maximizar el valor de los datos disponibles. La Metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) es un marco ampliamente utilizado para proyectos de minería de datos y análisis predictivo. Desarrollada en 1996 por un consorcio liderado por IBM, proporciona un enfoque iterativo y flexible, diseñado para alinear los resultados

analíticos con los objetivos estratégicos de diferentes industrias (Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000)).

A continuación, en la Figura 1 se presenta el flujo de trabajo de la metodología CRISP-DM:

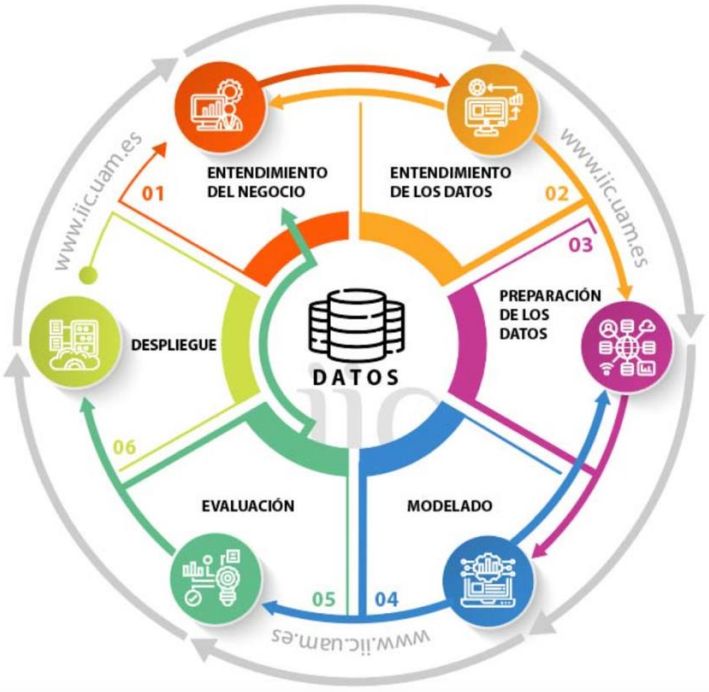


Figura 2 Flujo de trabajo de la metodología CRISP-DM Instituto de Ingeniería del Conocimiento (IIC). (2021).

La imagen ilustra un ciclo centrado en los datos que representa las seis etapas de la metodología CRISP-DM: comprensión del negocio, comprensión de los datos, preparación de datos, modelado, evaluación y despliegue. Cada fase se presenta de forma circular y conectada por flechas, reflejando un proceso iterativo y flexible en el que es posible volver a etapas anteriores según se vaya adquiriendo mayor entendimiento del problema o se requieran ajustes en la estrategia analítica. En el centro están los datos, componente fundamental del proceso, y alrededor se muestran

las fases, cada una con su propio ícono, enfatizando que la metodología no es lineal, sino que permite refinamientos y mejoras continuas a lo largo del desarrollo del proyecto (Instituto de Ingeniería del Conocimiento (IIC). (2021)).

Respecto al análisis de datos útil para identificar patrones es el Análisis Exploratorio de Datos (EDA) es un enfoque inicial en el análisis de datos que se centra en resumir y comprender las características principales de un conjunto de datos. Su objetivo principal es explorar patrones, relaciones y tendencias antes de aplicar técnicas de modelado más avanzadas. Para la predicción del número de emergencias clasificadas por especialidad y estación de Bomberos, el EDA puede:

- Comprender cómo varía la demanda de emergencias en relación con factores externos.
- Preparar y limpiar los datos, manejando datos faltantes y valores atípicos.
- Informar sobre la selección de variables y el enfoque de modelado.

El presente trabajo de titulación plantea el desarrollo de un modelo predictivo, un modelo predictivo es una herramienta estadística y analítica que permite anticipar resultados futuros basados en datos históricos y patrones identificados. En el contexto del presente trabajo de titulación sobre la predicción de la cantidad de emergencias por especialidad y estaciones del Cuerpo de Bomberos del Distrito Metropolitano de Quito, un modelo predictivo puede ofrecer varias ventajas como:

- Anticipación a la demanda
- Toma de decisiones basadas en datos

- Identificación de patrones y tendencias

El modelo predictivo utiliza datos históricos para este contexto podrían ser el número de emergencias por estación para prever la demanda de atención de emergencias para cada estación de bomberos en el caso de que el modelo indique que una determinada estación probablemente enfrentará un aumento en las emergencias debido a un evento próximo, los resultados se podrían tomar en cuenta para reasignar personal de otras estaciones mejorando la respuesta y reduciendo el riesgo de tiempos de respuesta prolongados.

Los resultados del modelo de predicción de la cantidad posible de emergencias en estaciones de Bomberos pueden ser considerados para la toma de decisiones informadas y gestionar de forma proactiva los recursos necesarios. La Gestión de Recursos en emergencias se refiere al proceso de planificación, organización, y control de recursos necesarios para enfrentar situaciones de emergencia. Este concepto abarca una serie de actividades y estrategias que garantizan que los recursos humanos estén disponibles y se utilicen de manera eficiente durante una emergencia para minimizar el impacto y asegurar una respuesta efectiva.

Otra técnica ampliamente utilizada para predicción e identificar patrones complejos a partir de datos son Las redes neuronales artificiales (RNA) las cuales se han convertido en herramientas necesarias para la predicción de emergencias, debido a su capacidad para identificar patrones complejos con grandes volúmenes de datos. En el ámbito de la salud, las redes neuronales se han utilizado para predecir brotes de enfermedades infecciosas, aprovechando datos históricos y factores ambientales, un

ejemplo de esto es el estudio que empleó redes neuronales para predecir el brote de dengue en Brasil encontró que el modelo podía anticipar el número de casos semanales con alta precisión, al incluir variables como temperatura, humedad y precipitaciones (Baquero, O. S., Santana, L. M. R., & Chiaravalloti-Neto, F. (2018)). Esta capacidad predictiva facilita la planificación de recursos y la implementación de medidas preventivas en sistemas de salud pública, permitiendo una respuesta más rápida y eficiente.

En el ámbito de la seguridad pública, las redes neuronales también se han aplicado en la predicción de crímenes, un ejemplo de ello se lo realizó Estados Unidos, un sistema basado en RNA fue implementado para predecir delitos en Chicago, logrando identificar áreas de alto riesgo con una precisión del 90% (Yang, D., Liu, Q., & Zhao, X. (2021)). Este sistema permite que la policía local enfoque sus recursos en zonas críticas, mejorando la prevención y la respuesta ante crímenes. Las RNA en este contexto no solo ayudan a reducir el tiempo de respuesta, sino que también optimizan el uso de los recursos disponibles para la seguridad ciudadana.

Otro método de predicción basada en datos es La regresión lineal, este método estadístico es utilizado para modelar la relación entre una variable dependiente continua y una o más variables independientes mediante una ecuación lineal. Su objetivo es encontrar la mejor línea que minimice la suma de los errores al cuadrado entre las predicciones del modelo y los valores reales. En el presente plan de tesis la técnica de regresión lineal puede ser utilizada para predecir la cantidad de emergencias posibles en función de datos históricos de atención de emergencias en el Distrito

Metropolitano, como el tipo de emergencia, la hora del día, y la ubicación geográfica. Esta técnica permite entender la influencia de cada variable en la necesidad de recursos (Montgomery, D. C., & Runger, G. C. (2006)). En el sector de salud pública, los modelos de regresión han sido empleados para predecir la demanda de servicios médicos en eventos de emergencia, como los picos de hospitalización durante epidemias. Por ejemplo, en la pandemia de COVID-19 los modelos de regresión lineal y no lineal permitieron estimar la cantidad de camas de hospital y respiradores necesarios, ayudando a los sistemas de salud a planificar y gestionar sus recursos de manera anticipada (Kumar, A., Sharma, K., & Yadav, A. K. (2021)). Estos modelos han demostrado ser altamente precisos al utilizar datos históricos y tasas de infección en tiempo real, lo cual permitió a los gobiernos tomar decisiones informadas y reducir la mortalidad en varios países.

En el campo de la seguridad pública, la regresión logística es utilizada para prever incidentes como accidentes de tráfico en zonas urbanas congestionadas. Un estudio realizado en Japón utilizó un modelo de regresión para identificar factores de riesgo en zonas específicas de tráfico, lo cual ayudó a implementar medidas preventivas en los horarios y áreas de mayor incidencia de accidentes (Kawasaki, T., & Saito, H. (2019)). Esta información es valiosa para las autoridades de tránsito, que pueden ajustar sus operaciones y enfocar sus recursos en mitigar accidentes en puntos críticos.

También los árboles de decisión son modelos particularmente útiles en situaciones de emergencia debido a su capacidad para clasificar eventos y generar

reglas de decisión interpretables, en el campo de la salud pública los árboles de decisión se han usado para predecir emergencias médicas en pacientes de alto riesgo, como aquellos con enfermedades cardíacas. Un ejemplo claro de esto es el estudio en hospitales de Alemania utilizó árboles de decisión para anticipar emergencias cardíacas en pacientes, permitiendo una intervención rápida en el caso de los pacientes con mayor probabilidad de sufrir complicaciones (Meissner, A., Körner, M., & Boettcher, B. (2018)). Este enfoque ayuda al personal médico a priorizar los casos de alto riesgo, mejorando la eficiencia y efectividad en la atención hospitalaria.

En seguridad pública, los árboles de decisión han demostrado ser efectivos en la clasificación de incidentes de incendios. Un estudio en Australia utilizó un modelo de árbol de decisión para clasificar incendios forestales según su severidad, con variables como temperatura, velocidad del viento y humedad (Sharples, J. J., McRae, R. H., & Weber, R. O. (2019)). Los resultados permitieron una predicción precisa de la expansión de incendios, facilitando una respuesta oportuna y adecuada en recursos para mitigar el impacto en áreas vulnerables.

Para el desarrollo del presente trabajo de titulación se ha optado el uso de Random Forest debido a su capacidad para manejar grandes conjuntos de datos, variables heterogéneas y relaciones no lineales. A diferencia de métodos más simples como la regresión lineal, que asume una relación lineal entre las variables independientes y la variable dependiente, Random Forest combina múltiples árboles de decisión para capturar patrones complejos y reducir la varianza del modelo (Hastie, T., Tibshirani, R., & Friedman, J. (2009)). Asimismo, en contraposición a las redes

neuronales, que suelen requerir un mayor esfuerzo de ajuste de hiperparámetros y un volumen considerable de datos para alcanzar un desempeño óptimo, Random Forest ofrece una configuración más intuitiva y un entrenamiento menos costoso computacionalmente. Además, a diferencia de otros algoritmos altamente interpretables como árboles de decisión simples, que pueden sufrir de alta varianza, Random Forest mitiga este problema al promediar las predicciones de múltiples árboles entrenados sobre diferentes subconjuntos de datos. Esto mejora la generalización del modelo y lo convierte en una alternativa robusta, equilibrando interpretabilidad y capacidad predictiva. Por estos motivos, Random Forest es una elección consistente para el presente trabajo, ya que permite aprovechar las ventajas de otras técnicas minimizando sus desventajas, ofreciendo así un balance adecuado entre complejidad, precisión e interpretabilidad.

A continuación, se presenta la tabla comparativa entre diferentes algoritmos de predicción vs. Random Forest, destacando sus principales características, ventajas y desventajas (Hastie, T., Tibshirani, R., & Friedman, J. (2009).:

Algoritmo	**Ventajas**	**Desventajas**	**Comparación con Random forest**
Regresión Lineal	Fácil interpretación	Asume relación lineal	Random forest captura relaciones no lineales y maneja mejor la presencia de outliers.
	Rápido de entrenar	Sensible a valores atípicos	

Redes Neuronales	Capaces de modelar relaciones complejas	Mayor complejidad en ajuste de hiperparámetros	Random forest requiere menos ajuste de parámetros y es más rápido de entrenar en la mayoría de los casos.
	Alta precisión con grandes volúmenes de datos	Alto costo computacional	
Regresión Logística	Interpretación estadística clara	Asume relaciones lineales en el logit	Random forest puede capturar relaciones más complejas y no lineales, ofreciendo mayor flexibilidad.
	Adecuada para problemas de clasificación binaria	Menos efectiva con datos muy complejos	
Árboles de Decisión Simples	Fácil de interpretar	Alta varianza (sobreajuste)	Random forest promedia múltiples árboles, reduciendo el sobreajuste y mejorando la precisión del modelo.
	Rapidez en entrenamiento	Menor precisión al no promediar múltiples árboles	
Máquinas de Soporte Vectorial (SVM)	Efectivas en espacios de alta dimensión	Alto costo computacional	Random forest generalmente es más fácil de ajustar y entrenar con grandes conjuntos de datos, ofreciendo un rendimiento competitivo.
	Flexibles con kernels	Dificultad para ajustar hiperparámetros	

Figura 3 Comparación entre diferentes algoritmos de predicción vs. random forest

Los modelos predictivos deben ser validados para garantizar su capacidad de generalización y su precisión en datos no vistos. Para realizar esta validación, se deben emplear técnicas específicas y métricas que evalúen el desempeño del modelo (Raschka, S., & Mirjalili, V. (2019)). A continuación, se describen algunas técnicas utilizadas para la validación de modelos predictivos:

1. División del Conjunto de Datos: Se separan los datos en conjuntos de entrenamiento, validación y prueba (hold-out), con el fin de entrenar y evaluar el modelo en datos diferentes para evitar sobreajuste.
2. Técnicas de Validación:
    a. Validación Cruzada (Cross-Validation): Consiste en dividir los datos en k pliegues, entrenar el modelo en todos menos uno, y validar en el restante, repitiendo el proceso para cada pliegue. Es útil para evaluar modelos de forma robusta.
    b. Validación Cruzada Estratificada: Variante de la validación cruzada que conserva la distribución de las clases en los pliegues, ideal para datos desbalanceados.
    c. Validación Temporal: Se aplica en series temporales respetando la secuencia cronológica.
3. Métricas de Evaluación:
    a. Para clasificación: Precisión, recall, F1-score, matriz de confusión, y área bajo la curva ROC (AUC-ROC).
    b. Para regresión: MAE (Error Absoluto Medio), RMSE (Raíz del Error Cuadrático Medio), y R^2 (Coeficiente de Determinación).

4. Optimización de Hiperparámetros:
    a. Técnicas como Grid Search o Random Search combinadas con validación cruzada ayudan a encontrar los mejores parámetros para el modelo.
5. Curvas de Aprendizaje y Evaluación:
    a. Gráficos como las curvas ROC y curvas de aprendizaje permiten identificar problemas de sobreajuste o subajuste y mejorar el modelo.

Estas técnicas aseguran que el modelo predictivo sea confiable, generalizable y adecuado para resolver el problema que se plantee.

2.2 Marco Conceptual

Para comprender de mejor manera la forma en la que se va a desarrollar el presente trabajo de titulación se ha generado un diagrama conceptual (Figura 7) de la forma en la que se relaciona cada uno de los elementos a utilizar:

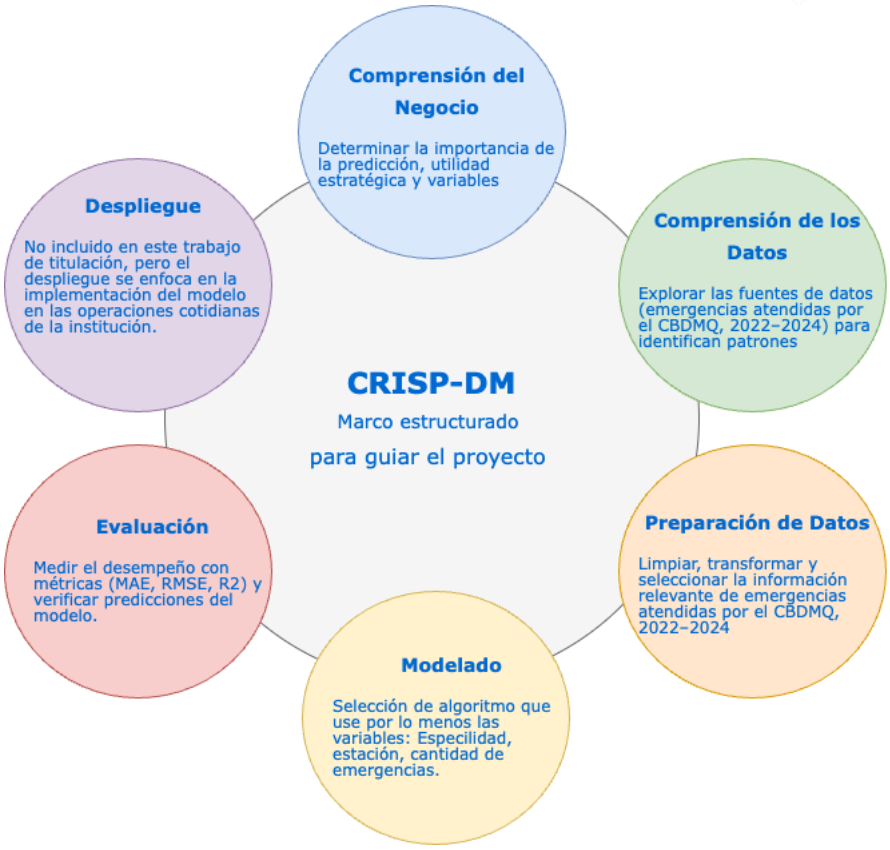


Figura 4 Diagrama conceptual del proceso estructurado para el desarrollo de un modelo predictivo basado en Machine Learning para anticipar el número de emergencias.

El diagrama muestra la estructura que se contempla para desarrollar el modelo predictivo basado en Machine Learning. El enfoque se estructura mediante la metodología CRISP-DM, asegurando un proceso organizado y alineado con las necesidades estratégicas del CB-DMQ. Para la estructura se ha considerado:

1. Metodología CRISP-DM: proporciona el marco estructural para guiar las etapas del proyecto, desde la recolección de datos hasta la implementación del modelo y su evaluación.
2. Comprensión del Negocio: Define la meta de predecir cuántas emergencias ocurrirán en cada estación y por qué es importante diferenciar el tipo de

especialidad (incendios, rescates, etc.). Ajusta los objetivos del proyecto con las necesidades reales de la institución, asegurando que la predicción sea útil para planificar recursos y mejorar tiempos de respuesta.

3. Comprensión de los Datos: Examina la calidad y relevancia de la información disponible, revisando la distribución de las emergencias por especialidad y estación, así como tendencias históricas. Verifica si existen inconsistencias, faltantes o duplicados que puedan afectar la precisión del modelo.
4. Preparación: Limpia y transforma los datos en un formato adecuado para el modelado, eliminando variables irrelevantes y lidiando con valores nulos. Genera variables adicionales por ejemplo, agrupaciones por día que facilitan la detección de patrones en la frecuencia de emergencias.
5. Modelado: Selecciona y entrena el algoritmo de Machine Learning incorporando hiperparámetros óptimos y estrategias de validación. Aquí, la especialidad, estación y cantidad de emergencias se combinan para generar pronósticos sobre la demanda futura.
6. Evaluación: Mide el desempeño del modelo comparando sus predicciones con los datos reales, empleando métricas como MAE, RMSE o R^2. Sirve para analizar en qué medida se aciertan las estimaciones y si es necesario refinar parámetros o reestructurar variables.
7. Despliegue: Aun que el despliegue no es parte del alcance de este trabajo de titulación, este punto se refiere a integra el modelo en la operación diaria, aprovechando sus resultados para ajustar los recursos operativos.

El trabajo de titulación no solo aborda un desafío local del Distrito Metropolitano de Quito, sino que también sienta las bases para replicar esta solución en otras ciudades con características similares, marcando un precedente en la gestión eficiente de emergencias mediante tecnologías predictivas.

La demanda de emergencias refleja la cantidad y especialidades que requieren atención en un tiempo específico, sirviendo como base para la estimación de recursos necesarios. Mediante un modelo predictivo basado en aprendizaje automático, se anticipa esta demanda utilizando datos históricos, lo que permite una planificación anticipada (Akkihal, A. R. (2006)).

La Regresión Lineal es una técnica estadística y de aprendizaje automático que busca establecer una relación lineal entre una variable dependiente y una o más variables independientes. En la predicción de emergencias, se utiliza para analizar cómo factores específicos (como la temperatura en la predicción de incendios) afectan la ocurrencia de eventos, facilitando así la toma de decisiones basadas en tendencias observadas (Kumar, A., Sharma, K., & Yadav, A. K. (2021)).

Los Modelos predictivos utilizan técnicas estadísticas y de machine learning para hacer predicciones basadas en datos históricos y actuales. Estos modelos analizan patrones y tendencias para anticipar eventos futuros.

Las herramientas de Big Data son plataformas que permiten almacenar, procesar y analizar grandes volúmenes de datos entre las cuales se incluye bases de

datos NoSQL, Hadoop y Spark, estas herramientas facilitan la integración y el análisis de datos en tiempo real para mejorar la toma de decisiones.

2.3 Metodología CRISP-DM

Como se indica en el marco teórico, la metodología CRISP-DM es ampliamente utilizada en proyectos de minería de datos y es reconocida por su significativa contribución en la gestión de proyectos de ciencia de datos. Su estructura proporciona un marco sistemático que guía a los equipos desde la comprensión inicial del negocio hasta la implementación de soluciones basadas en datos, facilitando una planificación y ejecución ordenada (Danalytics, 2023).

Un ejemplo práctico de la aplicación de la metodología CRISP-DM en proyectos de Ciencia de Datos en los cuales se consiguieron resultados exitosos fue realizado por el Banco Santander en España. El banco aplicó CRISP-DM para desarrollar modelos predictivos que mejoraran la detección de fraudes y optimizaran las campañas de marketing dirigidas. Mediante el análisis de grandes volúmenes de datos transaccionales y de clientes, el equipo de ciencia de datos pudo identificar patrones de comportamiento asociados con actividades fraudulentas y segmentar de manera más efectiva su base de clientes (Martínez, J., & Pérez, L. (2018)).

La implementación de CRISP-DM permitió al Banco Santander seguir un enfoque estructurado y sistemático. Durante la fase de comprensión del negocio, se establecieron claramente los objetivos que fueron: reducir las pérdidas por fraude y aumentar la efectividad de las campañas de marketing. En la fase de comprensión de

los datos, se analizaron fuentes de datos internas y externas para identificar variables relevantes. La preparación de los datos involucró la limpieza y transformación de los datos para asegurar su calidad y adecuación para el modelado.

En la fase de modelado, se utilizaron técnicas avanzadas de machine learning, como árboles de decisión y redes neuronales, para construir modelos predictivos. La fase de evaluación permitió validar los modelos y asegurarse de que cumplían con los objetivos establecidos. Finalmente, en la fase de despliegue, los modelos se integraron a la arquitectura empresarial del banco, permitiendo una detección de fraudes en tiempo real y campañas de marketing más efectivas.

Los resultados fueron significativos, se redujo el impacto financiero del fraude en un 20% y se incrementó la tasa de respuesta a las campañas de marketing en un 15%. Este proyecto demostró cómo CRISP-DM facilita la gestión eficiente de proyectos de ciencia de datos y contribuye al logro de resultados tangibles que benefician al negocio.

A continuación, se presenta un cronograma de tareas en cada una de las fases definidas por la metodología CRISP-DM aplicadas al presente proyecto:

Fase	**Tareas**	**Nov 2024**	**Dic 2024**	**Ene 2025**	**Feb 2025**	**Mar 2025**
	Definir objetivos, alcance,	X				

1. Comprensión del Negocio	métricas de éxito del modelo					
	Identificar actores involucrados y requisitos	X				
2. Comprensión de los Datos	Recopilar datos históricos del CB-DMQ		X			
	Analizar calidad, disponibilidad y relevancia de variables		X			
3. Preparación de Datos	Limpieza de datos (eliminación de duplicados, manejo de faltantes)		X	X		
	Generación de nuevas variables y transformaciones			X		

	Normalización y selección de características			X		
4. Modelado	Selección de algoritmo (Random Forest) y ajuste de hiperparámetros			X	X	
	Entrenamiento con conjunto de entrenamiento y validación cruzada			X	X	
5. Evaluación	Evaluar modelo usando MAE, RMSE				X	
	Ajustes finales y comparación con otros modelos				X	
6. Despliegue	Presentar resultados y conclusiones					X

	Generar reportes e informes técnicos					X

Figura 5 Cronograma de tareas en cada una de las fases definidas por la metodología CRISP-DM

Nota: El despliegue o implementación en producción no se contempla en este trabajo de titulación, solo la presentación de resultados como insumo para la toma de decisiones.

Durante la fase de evaluación del modelo, se utilizarán métricas apropiadas para modelos predictivos continuos, tales como el MAE (Mean Absolute Error) y el RMSE (Root Mean Squared Error). Estas métricas se calculan tras entrenar el modelo con los datos históricos, aplicándolo a un conjunto de prueba o a través de técnicas de validación cruzada para asegurar la representatividad de los resultados.

MAE (Mean Absolute Error)

Indica, en promedio, qué tan lejos están las predicciones del modelo de los valores reales. Un MAE bajo sugiere mayor precisión.

RMSE (Root Mean Squared Error)

Eleva al cuadrado los errores antes de promediar, dando más peso a errores grandes. Un RMSE bajo indica una mejor capacidad de predicción, especialmente en presencia de valores atípicos.

La comparación de MAE y RMSE permite no solo entender el nivel de error promedio, sino también identificar si existen errores puntuales muy grandes que impacten más la predicción algo que el RMSE evidenciará. Si estas métricas no

cumplen con los objetivos planteados inicialmente en la fase de comprensión del negocio, se podrá iterar nuevamente a etapas previas como la preparación de datos o el modelado con el fin de mejorar el desempeño del modelo.

2.4 Análisis Exploratorio de Datos (EDA)

El Análisis Exploratorio de Datos (AED) es usado con alta frecuencia en proyectos de ciencia de datos que implican examinar y visualizar conjuntos de datos para resumir sus características principales. Su objetivo es descubrir patrones, detectar anomalías, probar hipótesis y verificar supuestos mediante métodos gráficos y estadísticos (Shmueli, G., Bruce, P. C., Yahav, I., Patel, N. R., & Lichtendahl, K. C. (2017)).

El Análisis Exploratorio de Datos permite a los científicos de datos comprender la estructura de los datos, familiarizarse con las variables y entender las relaciones entre ellas. Esta comprensión facilita la formulación de hipótesis y orienta el análisis posterior de manera efectiva (Wickham, H., & Grolemund, G. (2017)). Además, el AED contribuye a identificar anomalías y valores atípicos que podrían distorsionar los resultados si no se manejan adecuadamente. Detectar estos elementos tempranamente en el proceso ayuda a garantizar la integridad y calidad de los datos utilizados en el modelado (Aggarwal, C. C. (2017)).

Otra contribución importante es la verificación de supuestos estadísticos. El AED permite evaluar si los datos cumplen con los supuestos necesarios para ciertos métodos estadísticos, como normalidad y homocedasticidad, lo cual determina la

validez de los modelos (Field et al., 2012). También facilita la selección y validación de modelos adecuados según las características de los datos (Geron, A. (2019)).

Un ejemplo práctico de la aplicación de Análisis Exploratorio de Datos es el estudio realizado con el conjunto de datos del Biobanco del Reino Unido (UK Biobank). Este proyecto recopila información genética, de salud y estilo de vida de más de 500,000 participantes, proporcionando un recurso invaluable para la investigación médica.

A través del AED, los investigadores exploraron este vasto conjunto de datos para identificar patrones y correlaciones entre marcadores genéticos, factores ambientales y diversas enfermedades. El AED permitió descubrir asociaciones previamente desconocidas y generar nuevas hipótesis sobre el desarrollo y progresión de enfermedades complejas como la diabetes, enfermedades cardiovasculares y trastornos neurodegenerativos. Este enfoque facilitó la visualización y comprensión de relaciones multifactoriales en un conjunto de datos de gran escala, contribuyendo al avance de la medicina personalizada (Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., ... & Collins, R. (2015)).

El uso efectivo del AED en este proyecto demostró cómo esta técnica extrae insights de grandes volúmenes de datos. Los hallazgos obtenidos a través del AED en el UK Biobank han influido en políticas de salud y en el diseño de intervenciones clínicas, evidenciando el impacto positivo de esta metodología en proyectos de ciencia de datos a gran escala.

2.5 Modelo predictivo basados en Machine Learning

Como ya se indicó en el marco teórico los modelos predictivos basados en Machine Learning son algoritmos que utilizan datos históricos para predecir resultados futuros. En proyectos de ciencia de datos, estos modelos contribuyen significativamente al análisis y comprensión de grandes volúmenes de información. Al automatizar el proceso de aprendizaje, los modelos de Machine Learning pueden manejar complejidades y variaciones en los datos que serían difíciles de captar mediante técnicas estadísticas tradicionales (Domingos, P. (2015)). Esto incluye la detección de patrones no lineales, interacciones entre variables y tendencias ocultas. Como resultado, es posible extraer insights valiosos y tomar decisiones informadas basadas en predicciones confiables, lo que es esencial en entornos competitivos y en rápida evolución.

Estos modelos fomentan la innovación al abrir posibilidades en campos emergentes como la conducción autónoma, donde los vehículos aprenden a navegar y tomar decisiones en tiempo real basándose en datos sensoriales (Goodfellow, I., Bengio, Y., & Courville, A. (2016)). También juegan un papel importante en la gestión de recursos naturales y la predicción de fenómenos climáticos, contribuyendo a soluciones sostenibles y a la mitigación de desastres naturales.

2.6 Validación del Modelo Predictivo

La validación del modelo es un paso imprescindible en la metodología científica aplicada a la Ciencia de Datos. El objetivo es asegurar que el modelo no solo funcione

adecuadamente con la información con la cual fue entrenado, sino que también sea capaz de predecir con precisión para nuevos conjuntos de datos. Este proceso contribuye a medir la robustez y la capacidad de generalización del modelo, evitando problemas de sobreajuste (overfitting).

Validación Cruzada en Datos Temporales

La validación cruzada es una técnica estándar que consiste en dividir el conjunto de datos en múltiples partes (folds) y entrenar y evaluar el modelo sucesivas veces con diferentes particiones, proporcionando una estimación más estable de su desempeño. Sin embargo, cuando se trabaja con datos temporales, la dependencia cronológica entre observaciones imposibilita el uso directo de la validación cruzada aleatoria típica.

En su lugar, se emplean variantes de validación cruzada diseñadas para series temporales (time series cross-validation), donde las particiones respetan la secuencia cronológica de los datos. Por ejemplo, se pueden crear subconjuntos de entrenamiento siempre anteriores en el tiempo a los datos de validación, avanzando iterativamente en el horizonte temporal. Esta aproximación permite evaluar el modelo en escenarios más realistas, emulando la predicción futura a partir de datos estrictamente pasados (Hyndman & Athanasopoulos, 2018).

Un esquema simplificado es:

- Dividir la serie temporal en un intervalo inicial para entrenamiento y uno siguiente para validación.
- Entrenar el modelo en el primer intervalo y validar en el segundo.

- Ampliar la ventana de entrenamiento agregando más datos recientes y volver a validar con el siguiente intervalo futuro.
- Repetir el proceso varias veces para obtener una medida robusta del desempeño promedio del modelo en distintas condiciones temporales.

Métricas y Visualizaciones en la Validación del Modelo

Para entender a fondo el desempeño del modelo, es necesario emplear métricas apropiadas y visualizaciones que ayuden a interpretar sus resultados:

Curva de Error (Learning Curve)

Este gráfico muestra cómo evoluciona el error o la puntuación del modelo a medida que se incrementa el número de muestras de entrenamiento. Permite evaluar si el modelo se está sobre ajustando overfitting o subajustando underfitting, así como determinar si más datos podrían mejorar el desempeño. Si las curvas de entrenamiento y validación convergen, generalmente significa que el modelo ha logrado un buen ajuste.

Ejemplo de gráfico Curvas ROC:

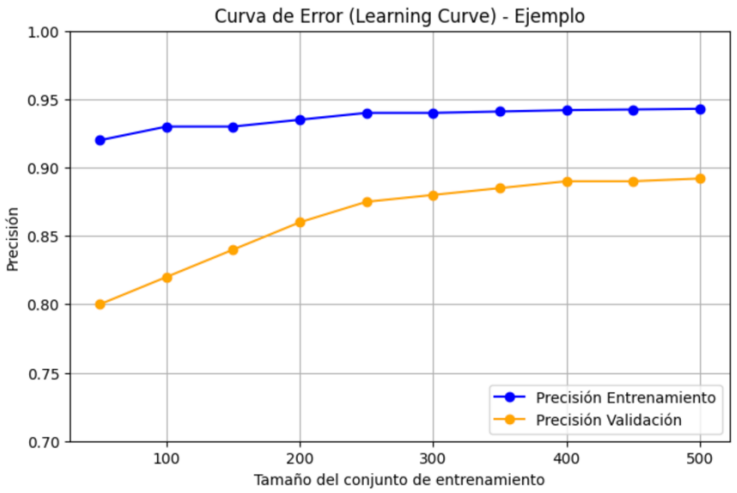


Figura 6 Gráfico curvo de error (Learning curve)

Gráfico predicción vs. valor real

Este gráfico, útil sobre todo en regresión, muestra en el eje X los valores reales y en el eje Y las predicciones del modelo. Una línea diagonal y = x indica predicciones perfectas. La dispersión de los puntos en torno a esta línea da una idea del error y la precisión.

Ejemplo de gráfico predicción vs. valor real:

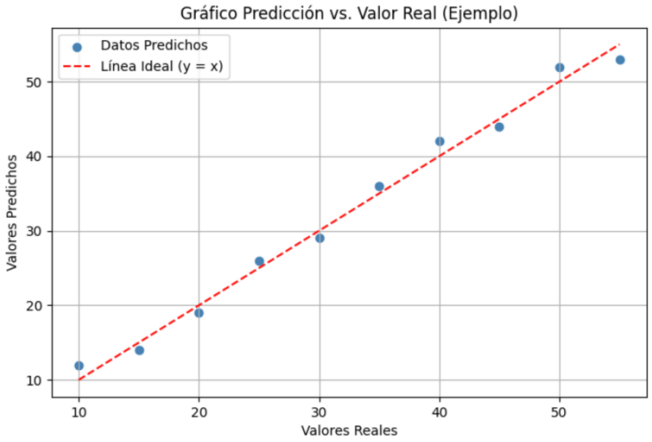


Figura 7 Gráfico predicción vs. valor real

Distribución de Errores (Error Distribution Plot)

Un gráfico que muestre la distribución de los errores de predicción (por ejemplo, residuales en regresión) ayuda a evaluar el sesgo del modelo. Una distribución centrada alrededor de cero con dispersión reducida indica un modelo equilibrado. Las asimetrías o colas largas pueden sugerir que el modelo sistemáticamente subestima o sobreestima el valor real.

Ejemplo de gráfico de distribución de errores

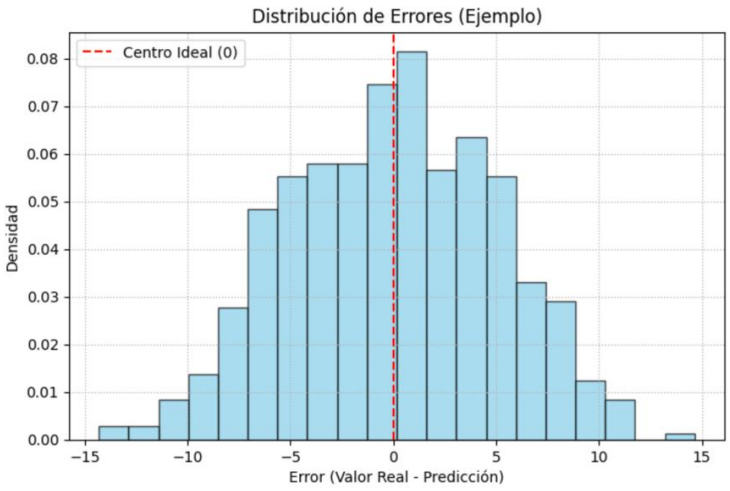


Figura 8 Gráfico distribución de errores

Curvas ROC (Receiver Operating Characteristic)

En problemas de clasificación, la curva ROC muestra la relación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) a medida que se varía el umbral de decisión. El área bajo la curva (AUC) es una métrica común que resume el desempeño del clasificador. Una curva ROC más cercana al vértice superior izquierdo indica un mejor rendimiento.

Ejemplo de gráfico Curvas ROC:

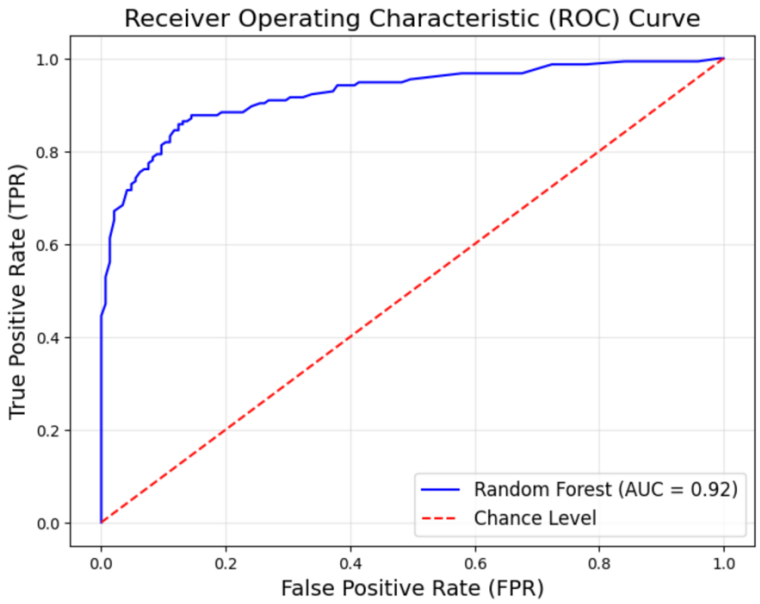


Figura 9 Gráfico de la curva ROC con ejes TPR vs. FPR, destacando el AUC

Curvas de Error (MAE, RMSE) para Modelos de Regresión

En el caso de modelos que predicen valores numéricos regresión, métricas como el MAE Mean Absolute Error y el RMSE Root Mean Squared Error permiten cuantificar el error de predicción. Gráficos de error en función del tiempo o del número de iteraciones de entrenamiento ayudan a visualizar cómo mejora o empeora el desempeño del modelo con ajustes sucesivos.

Ejemplo de gráfico de Curvas de Error

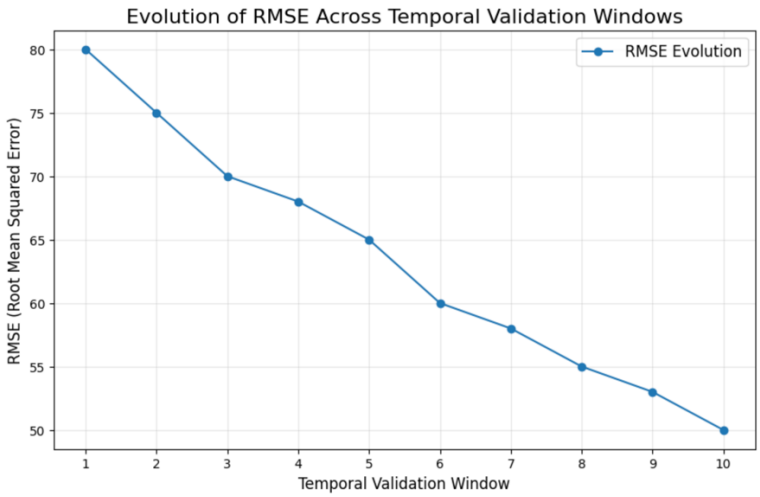


Figura 10 Gráfico de línea que muestre la evolución del RMSE a lo largo de diferentes ventanas de validación temporal, evidenciando si el error se reduce conforme se avanza en las etapas del modelado.

Estas visualizaciones y métricas, integradas en el proceso de validación, facilitan la toma de decisiones sobre la adecuación del modelo, la necesidad de ajustes en los parámetros o la incorporación de nuevas variables. Así, la validación del modelo predictivo no solo es un punto de control obligatorio, sino también una herramienta para guiar la mejora continua del modelo y, en última instancia, la calidad de las predicciones.

Finalmente, luego de la revisión de los fundamentos teóricos, el presente trabajo de titulación se orienta a la aplicación de la metodología CRISP-DM sobre datos de emergencias atendidas por las estaciones de Bomberos Quito en el Distrito Metropolitano. En el siguiente capítulo, se describen los aspectos metodológicos que regirán la recolección de datos, la preparación de los mismos y la implementación práctica del modelo de Machine Learning.

CAPÍTULO III: MARCO METODOLÓGICO

3. Marco Metodológico

3.1 Marco Metodológico de Investigación

Para el desarrollo de este estudio, se ha adoptado un enfoque metodológico estructurado que permite abordar de manera sistemática la predicción de emergencias en el Distrito Metropolitano de Quito. Dado que el objetivo principal es desarrollar un modelo basado en Machine Learning, se combinarán métodos descriptivos, exploratorios, correlacionales y experimentales, de acuerdo con cada fase del proceso.

El análisis de datos se basará en información histórica de emergencias atendidas por Bomberos del Distrito Metropolitano de Quito. Esta información incluye registros de emergencias clasificadas por fecha y hora de emergencia, especialidad, subespecialidad, estación que atendió la emergencia, recursos utilizados, entre otras variables relevantes.

Para la recopilación y procesamiento de los datos, se emplearán herramientas especializadas, entre ellas Python y sus bibliotecas para análisis de datos (Pandas, NumPy, Scikit-learn). Los datos obtenidos serán almacenados en una estructura temporal, lo que permitirá manipular y prepararlos sin afectar los registros originales de emergencias.

Una vez que se cuente con los datos, los mismos pasarán por un proceso de limpieza y transformación, asegurando que la información esté libre de inconsistencias

y duplicaciones. Posteriormente, se aplicará un análisis exploratorio utilizando herramientas de visualización como Matplotlib y Seaborn, con el objetivo de identificar patrones de ocurrencia de emergencias en las distintas estaciones de bomberos.

Los resultados de esta exploración serán clave para definir las características que alimentarán el modelo predictivo. Finalmente, se empleará Jupyter Notebook para presentar visualmente los hallazgos, facilitando la visualización e interpretación de los datos.

3.2 Marco Metodológico en Ciencia de Datos

Para el desarrollo del modelo predictivo, se ha optado por seguir la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining), un enfoque ampliamente reconocido en proyectos de ciencia de datos como se ha indicado en el marco teórico de este trabajo de titulación.

3.2.1 Comprensión del Negocio

En esta primera fase, se profundizará en las necesidades del CB-DMQ en términos de planificación y respuesta ante emergencias. Se busca entender cómo la predicción de incidentes puede contribuir a una mejor distribución de recursos orientada a una reducción en los tiempos de respuesta. Para la comprensión del negocio se realizaron entrevistas con personal encargado de la planificación de recursos operativos de Bomberos, en estas entrevistas se identificó el objetivo de que el modelo debe estimar la cantidad de emergencias desglosadas por especialidad

esperadas en cada estación, con un periodo de predicción de días, también surgieron las siguientes preguntas:

- ¿Cuáles son los patrones de ocurrencia de emergencias en las distintas estaciones?
- ¿En qué períodos del año se presenta mayor concentración de emergencias?
- ¿Cómo pueden optimizarse los recursos disponibles en función de estas predicciones?

3.2.2 Comprensión de los Datos

Una vez comprendido el problema, el siguiente paso es el análisis detallado de los datos históricos. En esta fase se recopilarán registros de los últimos años, considerando aspectos como:

- Especialidad de emergencia (incendios, rescates, manejo de materiales peligrosos, emergencias prehospitalarias, eventos hídricos y meteorológicos).
- Fecha y hora de la emergencia, para detectar patrones temporales.
- Estación de bomberos que atendió la emergencia.

Además, se analizará la calidad de los datos, verificando la presencia de valores faltantes, datos atípicos o inconsistencias que puedan afectar el desempeño del modelo.

3.2.3 Preparación de los Datos

En esta etapa, se trabajará en la limpieza y transformación de los datos para garantizar su adecuación en el proceso de modelado. Algunas de las acciones que se llevarán a cabo incluyen:

- Eliminación de registros duplicados.
- Manejo de valores nulos mediante técnicas de imputación.
- Normalización de variables numéricas para mejorar el rendimiento del modelo.
- Codificación de variables categóricas mediante técnicas como One-Hot Encoding, asegurando que puedan ser interpretadas correctamente por los algoritmos de Machine Learning.

Además, se explorará la creación de nuevas variables a partir de la combinación de datos existentes, con el fin de mejorar la capacidad predictiva del modelo.

3.2.4 Modelado

Para la implementación del modelo predictivo, se ha seleccionado Random Forest, un algoritmo que ha demostrado ser altamente efectivo en problemas de predicción con datos estructurados. Este modelo permite manejar una gran cantidad de variables y tiene la ventaja de ser resistente al sobreajuste, lo que mejora su capacidad de generalización.

El proceso de modelado incluirá:

- División del conjunto de datos en entrenamiento (80%) y prueba (20%).

- Entrenamiento del modelo utilizando validación cruzada, con el fin de evaluar su rendimiento en distintos subconjuntos de datos.
- Ajuste de hiperparámetros mediante Grid Search, optimizando la profundidad de los árboles, el número de estimadores y otros parámetros clave.
- Comparación con otros modelos, como regresión lineal y redes neuronales, para evaluar cuál ofrece mejores resultados.

3.2.5 Evaluación del Modelo

Una vez entrenado el modelo, será sometido a pruebas rigurosas para determinar su precisión y confiabilidad. Para ello, se emplearán métricas como:

- MAE (Mean Absolute Error): Medirá la diferencia media entre las predicciones del modelo y los valores reales.
- RMSE (Root Mean Squared Error): Permitirá evaluar la dispersión del error, dando mayor peso a los errores más grandes.
- Curvas de aprendizaje: Se analizará el comportamiento del modelo a medida que se incrementa la cantidad de datos de entrenamiento, permitiendo detectar posibles problemas de sobreajuste o subajuste.

Además, se realizarán gráficos de dispersión y matrices de error para visualizar las diferencias entre las predicciones y los valores reales, lo que permitirá identificar posibles patrones de error y ajustar el modelo en consecuencia.

3.2.6 Presentación de Resultados y Aplicabilidad

Si bien el modelo no será implementado de manera operativa en el CB-DMQ dentro del alcance de esta investigación, los resultados serán presentados de manera que puedan servir como insumo estratégico para la toma de decisiones.

En esta fase se llevarán a cabo las siguientes acciones:

- Elaboración de informes con las predicciones del modelo.
- Creación de visualizaciones para facilitar la interpretación de los resultados.
- Documentación del código y las metodologías utilizadas, permitiendo futuras mejoras.

Con estos resultados, se espera generar un impacto positivo en la gestión de emergencias en el DMQ, proporcionando herramientas que permitan anticipar la demanda y optimizar el uso de recursos.

CAPÍTULO IV: RESULTADOS

4. Aplicación de técnicas de minería de Datos

4.1 Comprensión de los datos

De acuerdo con lo indicado en el capítulo III Marco Metodológico, se identifica que los datos de emergencias atendidas por Bomberos del Distrito Metropolitano de Quito son datos que contiene fecha y hora de emergencia, especialidad, subespecialidad, estación que atendió la emergencia, recursos utilizados, entre otras variables relevantes.

A continuación, se presentan las 13 variables con las que se va a trabajar para el desarrollo del modelo predictivo:

- numero_parte
- anio
- mes
- diadelmes
- codigo_especialidad
- especialidad
- codigo_subespecilidad
- subespecilidad
- forma_aviso
- numero_vehiculos
- numero_personal_operativo
- estacion
- canton

4.2 Recopilación de los datos

Los datos de atención de emergencias por especialidad y estación fueron obtenidos de las bases de datos oficiales de Bomberos del Distrito Metropolitano de Quito, los datos son parte de reportes operativos, se ha considerado para este estudio los datos de los años 2022, 2023 y 2024 los mismos serán suficiente para identificar patrones significativos en la ocurrencia de emergencias.

Cada registro contiene detalles como la fecha y hora de emergencia, Especialidad y subespecialidad de la emergencia, el código de la estación de bomberos que atendió la emergencia. Los datos fueron integrados y normalizados para garantizar su consistencia antes del análisis exploratorio además se verificó que entre las variables no consten registros de datos personales de las víctimas atendidas esto con el fin de no contravenir la Ley Orgánica de Protección de Datos.

4.2 Descripción de los datos

A continuación, se describen las variables que se utilizarán para el modelo predictivo:

- **numero_parte:** Esta variable contiene un identificador único de la emergencia y es de tipo numérico
- **anio:** Esta variable contiene el año de la emergencia atendida y es de tipo numérico
- **mes:** Esta variable contiene el mes de la emergencia atendida y es de tipo numérico
- **diadelmes:** Esta variable contiene el día del mes de la emergencia atendida y es de tipo numérico
- **codigo_especialidad:** Esta variable contiene el año de la emergencia atendida y es de tipo numérico
- **especialidad:** Esta variable contiene el nombre de la especialidad de la emergencia atendida y es de tipo varchar

- **codigo_subespecilidad:** Esta variable contiene el código de la especialidad de la emergencia atendida y es de tipo numérico
- **subespecilidad:** Esta variable contiene el nombre de la subespecialidad de la emergencia atendida y es de tipo varchar
- **forma_aviso:** Esta variable contiene la forma de aviso de la emergencia atendida y es de tipo varchar
- **numero_vehiculos:** Esta variable contiene el número de vehículos que asiste a la emergencia y es de tipo numérico
- **numero_personal_operativo:** Esta variable contiene el número de personal operativo que asiste a la emergencia y es de tipo numérico
- **estacion:** Esta variable contiene el código de la estación que atendió la emergencia y es de tipo varchar
- **canton:** Esta variable contiene el nombre del cantón en el cual se atendió la emergencia y es de tipo varchar

4.3 Exploración y verificación calidad de los datos

Para comprender la distribución y comportamiento de los datos, se llevó a cabo un análisis exploratorio de datos (EDA). Para realizar el análisis exploratorio de los datos se utilizó Google Colab para visualizar gráficamente el comportamiento de las variables que son parte de este estudio. A continuación, se destacan los principales hallazgos:

Descripción general de variables numéricas:

Variable	Media (mean)	Mínimo (min)	25% (Q1)	Mediana (Q2)	75% (Q3)	Máximo (max)	Observaciones
numero_parte	445,563.85	381,747	411,554.75	444,841.50	481,343.50	506,653	Serie numérica que identifica a cada emergencia.
anio	2022.99	2022	2022	2023	2024	2024	Contiene datos de emergencias de los años 2022, 2023 y 2024.
mes	6.63	1	4	7	9	12	Se visualiza una distribución homogénea en la atención de emergencias en todos los meses.
diadelmes	15.75	1	8	16	23	31	Se mantiene una distribución homogenia, sin concentraciones de emergencias visibles.
codigo_ subespecilidad	310.40	287	287	296	336	403	Es un listados de códigos numpericos de subespecialidad de las emergencias.

numero_ vehiculos	1.24	1	1	1	1	30	En esta variable se puede ver que en la mayoría de emergencias se moviliza un solo vehículo, pero hay casos excepcionales con hasta 30 vehículos.
numero_personal_ operativo	3.15	0	2	3	4	45	Se identifica que normalmente asisten a las emergencias entre 2 y 4 bomberos sin embargo, se presentan emergencias en las que asisten hasta 45.

Figura 11 Descripción general de variables numéricas de emergencias atendidas por el CBDMQ, 2022-2024

Descripción general de variables categóricas:

Variable	**Valores Únicos (unique)**	**Valor más frecuente (top)**	**Frecuencia (freq)**	**Observaciones**

Especialidad	7	Prehospitalaria	50,846	La mayoría de las emergencias están relacionadas con atención prehospitalária.
Subespecialidad	30	Trauma	29,731	Trauma es la subespecialidad más común entre las emergencias atendidas.
Forma de Aviso	16	CIU	58,207	La mayoría de emergencias son reportadas con el código CIU Ciudadanía (ECU911).
Estación	26	X6	7,318	La estación X6 atiende la mayor cantidad de emergencias registradas en Quito.
Cantón	13	Quito	76,007	La mayoría de los registros son de emergencias atendidas en Quito.

Figura 12 Descripción general de variables categóricas de emergencias atendidas por el CBDMQ, 2022-2024

Comprobación de valores faltantes en el conjunto de datos:

```
#Verificar valores faltantes:
df.isnull().sum()
```

	0
numero_parte	0
anio	0
mes	0
diadelmes	0
codigo_especialidad	0
especialidad	0
codigo_subespecilidad	0
subespecilidad	0
forma_aviso	4
numero_vehiculos	12
numero_personal_operativo	12
estacion	0
canton	0

dtype: int64

Figura 13 Verificación de valores faltantes en conjunto de datos de emergencias antendidas por el CBDMQ, 2022-2024

Representación gráfica de la distribución de emergencias:

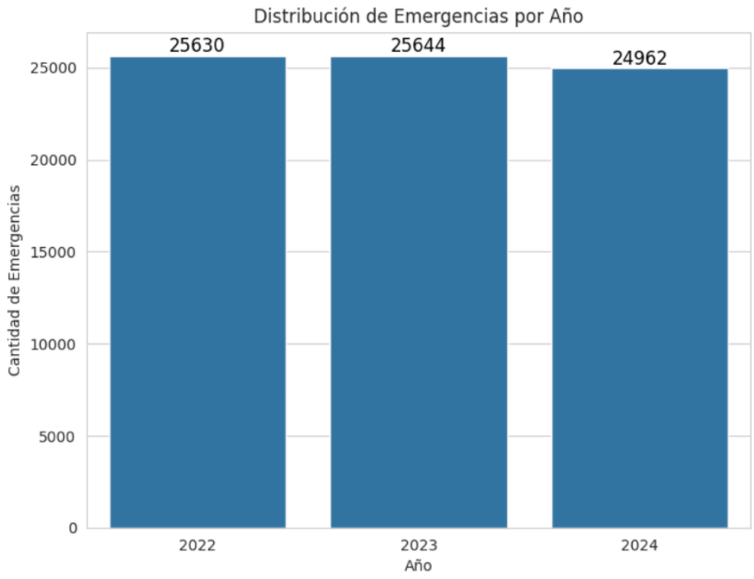


Figura 14 Distribución anual de emergencias atendidas por el CBDMQ, 2022-2024.

En la gráfica de barras se presenta la cantidad de emergencias atendidas por Bomberos del Distrito Metropolitano de Quito en los años 2022, 2023 y 2024. Como conclusiones al ver la gráfica se puede decir que el número de emergencias se ha mantenido estable entre 2022 y 2023, con una pequeña diferencia de 14 atenciones, en 2024 las emergencias han disminuido ligeramente respecto al año anterior la diferencia es de 682 emergencias que equivale al 2.66% de emergencias atendidas finalmente se evidencia que no existe una variación significativa en los años 2022, 2023 y 2024, lo que sugiere que el comportamiento de las emergencias es constante año a año.

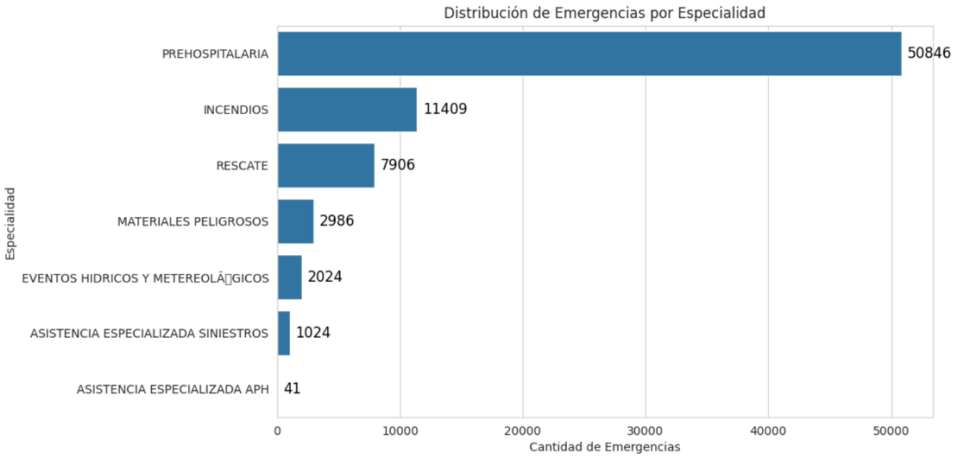


Figura 15 Distribución de emergencias por especialidad atendidas por el CBDMQ, 2022-2024

El gráfico muestra que la mayoría de las emergencias atendidas por Bomberos del Distrito Metropolitano de Quito corresponden a la especialidad Prehospitalaria, representando más del 66% del total de emergencias atendidas. Los incendios y rescates también tienen una presencia significativa, aunque en menor proporción, lo que sugiere que los eventos estructurales y vehiculares requieren intervención frecuente, pero no predominan sobre las emergencias de salud. Por otro lado las especialidades Materiales peligrosos, Eventos hídricos y meteorológicos y asistencia especializada son menos comunes en el Distrito Metropolitano.

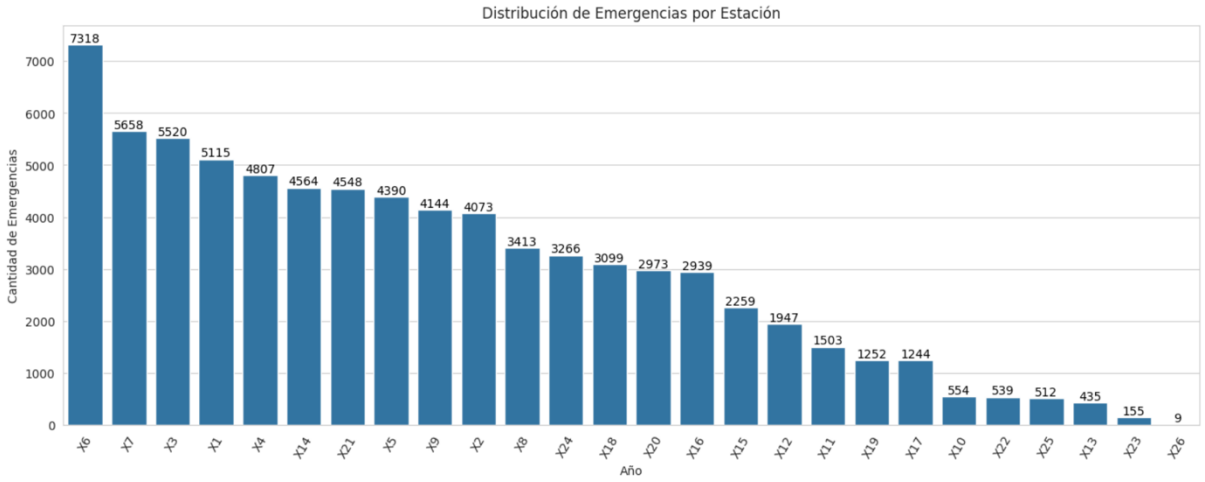


Figura 16 Distribución de emergencias por estación atendidas por el CBDMQ, 2022-2024

El gráfico muestra la distribución de emergencias atendidas por estación, destacando que algunas estaciones tienen una carga operativa mayor que otras. En este caso los datos muestran que la estación X6 atiende más emergencias con 7.318 atenciones, las estaciones que le sigue son X7, X3 y X1 todas con atención de más de 5.000 emergencias, lo que sugiere que estas estaciones cubren áreas con alta demanda. A medida que se recorre hacia la derecha la cantidad de emergencias disminuye rápidamente, con estaciones como X23 y X26 que registran menos de 500 emergencias, esto es explicado ya que la estación x23 está localizada en Nono que es un sector con menos población, en el caso de la estación x26 es una estación que se inauguró el 20 de diciembre 2024.

Durante esta etapa de verificación de calidad de los datos se constató que casi la totalidad de los registros con los que se cuenta tienen coherencia. La calidad de los datos se explica porque Bomberos del Distrito Metropolitano de Quito utiliza un Sistema Integrado Partes de emergencia en cual gestiona todos los recursos operativos durante

todo el proceso de atención de la emergencias y los datos son registrados en formularios que validad el registro atreves de reglas como ejemplo la hora que llegada a la emergencia no puede ser menor a la hora de despacho, gracias a este proceso, cualquier omisión o error en los campos se identifica y corrige de inmediato, impidiendo la generación de inconsistencias, datos nulos o duplicaciones.

La estandarización en los diferentes tipos de datos, por ejemplo: fechas, códigos de estación restringidos a una lista de selección, campos obligatorios, validación en cronología de las fechas del proceso de atención de la emergencia está controlada por el Sistema, de este modo se evita el registro de datos con inconsistencia y se asegura que los registros cumplan con los formatos para facilitar el análisis posterior.
Finalmente se puede evidenciar que el sistema Informático utilizado por Bomberos contribuye directamente a la integridad y fiabilidad de la información de la atención de emergencias, brindando una base sólida para aplicar técnicas de Ciencia de Datos.

4.4 Preparación y Limpieza de Datos

En esta fase, el objetivo principal fue adecuar la información de tal manera que resultara realmente útil para el modelo predictivo. Concretamente, se eliminaron variables que no aportaban valor a la predicción, y se organizaron los datos en un formato que facilitara el entrenamiento y la interpretación del modelo. A continuación, se detalla cada uno de los pasos realizados.

4.4.1 Tratar valores faltantes

Para garantizar la calidad de los datos y evitar sesgos en el modelo predictivo, fue necesario abordar los valores faltantes en algunas variables. La presencia de datos nulos puede afectar la precisión del modelo, por lo que se optó por aplicar técnicas de imputación basadas en la naturaleza de cada variable.

En el caso de las variables numero_vehiculos y numero_personal_operativo, que contienen valores numéricos continuos, se utilizó la imputación por media aritmética. Esta técnica permitió mantener la distribución original de los datos sin introducir sesgos significativos.

Por otro lado, la variable forma_aviso, al ser de tipo categórico, se trató mediante imputación con moda. Se identificó el valor más frecuente dentro de la columna y se utilizó como reemplazo para los datos faltantes. Esta estrategia resultó adecuada, ya que evita la distorsión de la distribución de categorías y permite mantener la coherencia del conjunto de datos.

A continuación se presenta el código con el cual se trató los datos faltantes en el dataset de emergencias.

```
# Tratar datos faltantes
# Imputación con la media en variables:
# 'numero_vehiculos' y 'numero_personal_operativo'
df['numero_vehiculos'].fillna(df['numero_vehiculos'].mean(), inplace=True)
df['numero_personal_operativo'].fillna(df['numero_personal_operativo'].mean(), inplace=True)

# Imputación con la moda en variable: 'forma_aviso'
moda_forma_aviso = df['forma_aviso'].mode()[0]
df['forma_aviso'].fillna(moda_forma_aviso, inplace=True)

# Verificamos nuevamente valores faltantes
print(df.isnull().sum())
```

```
numero_parte                0
anio                        0
mes                         0
diadelmes                   0
codigo_especialidad         0
especialidad                0
codigo_subespecilidad       0
subespecilidad              0
forma_aviso                 0
numero_vehiculos            0
numero_personal_operativo   0
estacion                    0
canton                      0
dtype: int64
```

Figura 17 Tratamiento de valores faltantes en conjunto de datos de emergencias mediante inputación con media y moda

Tras la imputación, se realizó una verificación para confirmar que no quedaran valores nulos en el dataset. Este proceso garantiza que el modelo trabajara con información completa y consistente, reduciendo el riesgo de errores en las etapas posteriores de entrenamiento y validación.

4.4.2 Filtrado de datos

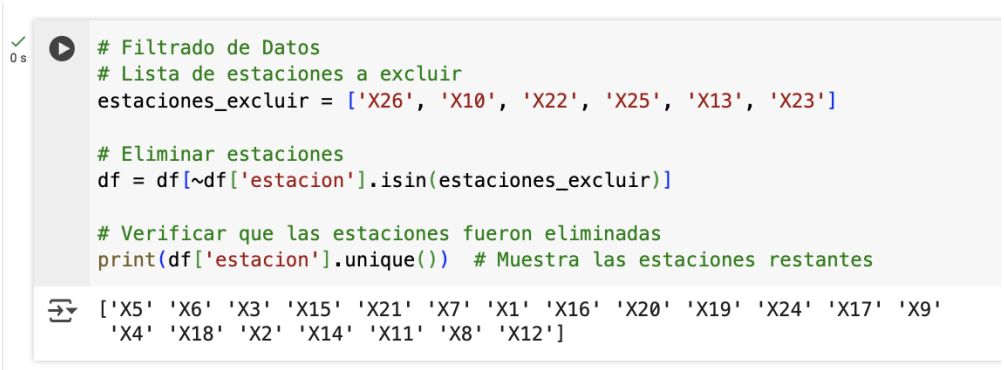
Como parte del proceso de preparación, se realizó un análisis de la distribución de emergencias el cual se puede ver en la “Figura 16 Distribución de emergencias por estación atendidas por el CBDMQ, 2022-2024”, la distribución por estación permitió identificar las estaciones que aportan información significativa al modelo predictivo. Por esta razón se decidió excluir ciertas estaciones con base en dos criterios principales:

1. Falta de datos históricos suficientes: La estación X26 fue inaugurada el 20 de diciembre de 2024, por lo que su registro de emergencias en el

periodo analizado (2022-2024) es prácticamente inexistente. Debido a la ausencia de datos históricos, no es posible extraer patrones confiables sobre su operación, lo que afectaría la capacidad del modelo para hacer predicciones en dicha estación.

2. Baja frecuencia de emergencias atendidas: Las estaciones X10, X22, X25, X13 y X23 muestran un número significativamente menor de emergencias en comparación con las demás estaciones. Como se observa en la “Figura 16 Distribución de emergencias por estación atendidas por el CBDMQ, 2022-2024”, estas unidades atendieron menos de 600 emergencias en tres años, lo que sugiere que su volumen de actividad es muy bajo. Incluir estas estaciones en el modelo podría introducir ruido en el entrenamiento, dado que el patrón de demanda de emergencias en estas unidades es irregular y estadísticamente menos representativo.

A continuación, se presenta el código con el cual se excluyó las estaciones del conjunto de datos de emergencia:



```
# Filtrado de Datos
# Lista de estaciones a excluir
estaciones_excluir = ['X26', 'X10', 'X22', 'X25', 'X13', 'X23']

# Eliminar estaciones
df = df[~df['estacion'].isin(estaciones_excluir)]

# Verificar que las estaciones fueron eliminadas
print(df['estacion'].unique())  # Muestra las estaciones restantes
```

```
['X5' 'X6' 'X3' 'X15' 'X21' 'X7' 'X1' 'X16' 'X20' 'X19' 'X24' 'X17' 'X9'
 'X4' 'X18' 'X2' 'X14' 'X11' 'X8' 'X12']
```

Figura 18 Filtrado de estaciones con baja frecuencia de atención de emergencias, 2022-2024

Luego de excluir las estaciones se puede garantizar que la predicción se basa en estaciones con registros suficientes y patrones de demanda más estables.

4.4.3 Eliminación de variables

El este paso se determinó cuáles son las columnas realmente necesarias para estimar el número de emergencias por estación y especialidad, y cuáles, en cambio, solo complicaban la construcción del modelo. Luego del análisis correspondiente se decidió descartar las siguientes variables:

- numero_parte: Es un identificador único de cada emergencia, pero no aporta información útil para el análisis de tendencias o patrones. Es más útil como referencia administrativa.
- codigo_especialidad: Dado que la columna especialidad ya contiene la descripción textual de la especialidad, el código es redundante y puede eliminarse sin pérdida de información.
- codigo_subespecilidad: Similar al caso anterior, subespecilidad ya describe la categoría de la emergencia, haciendo que el código numérico sea innecesario.
- subespecilidad: Puede eliminarse ya que no se requiere en este estudio llegar a predecir eventos de emergencia a un nivel de subespecialidad.
- forma_aviso: Este campo indica como se reportó la emergencia y se ve que no influye en la gravedad ni el tipo de la emergencia, por lo que puede eliminarse
- numero_vehiculos: No se requiere esta variable ya que el trabajo no plantea el análisis de recursos operativos utilizados en cada emergencia.

- numero_personal_operativo: Similar a la variable anterior, no se requiere esta variable ya que el trabajo no plantea el análisis de recursos operativos utilizados en cada emergencia.
- canton: Las emergencias en el conjunto de datos ocurren en el mismo cantón es por esta razón que la variable no es relevante para el análisis.

A pesar de que estas variables son útiles en la operación diaria del Bomberos por ejemplo, para saber cómo llega la alerta o cuántos vehículos se despacharon a una emergencia, se vio que no influyen de manera significativa a la hora de predecir la ocurrencia de emergencias.

Eliminar estas variables permitió simplificar la estructura del dataset y evitar que el modelo procesara información que no aporta al modelo predictivo, ahorrando así tiempo y recursos durante el entrenamiento.

4.4.4 Agrupación y creación de nuevas variables

Luego de la eliminación de variables se reorganizó el conjunto de datos mediante el agrupamiento de variables:

- anio
- mes
- diadelmes
- especialidad
- estación

El propósito de este reordenamiento fue concentrar en un mismo registro toda la información relacionada con la cantidad de emergencias atendidas cada día, en cada estación, y para cada tipo de especialidad. Tras aplicar la operación de agrupamiento (normalmente un “group by” en Python o en SQL), se generó una columna adicional:

- emergencias_atendidas: indica el total de casos registrados para una combinación específica de fecha (año, mes, día), tipo de emergencia (especialidad) y estación de bomberos.
- dia_de_la_semana: indica en números el día de la semana de acuerdo a la fecha de atención de la emergencia de acuerdo a: Lunes = 0 ….. Domingo = 6
- tipo_dia_festivo: esta variable identifica si el día de la atención de la emergencia es un día regular, feriado o puente festivo.

Gracias a este procedimiento, se obtuvo una vista más consolidada y sencilla de manejar, con las siguientes variables finales:

- anio
- mes
- diadelmes
- dia_de_la_semana
- especialidad
- estación
- emergencias_atendidas
- tipo_dia_festivo

Al trabajar con menos columnas y datos más organizados, se facilita la lectura y análisis por parte del modelo.

4.4.5 Transformación de variables categóricas

Como parte de los últimos ajustes realizados al conjunto de datos de emergencias se transformó las siguientes variables categóricas:

- especialidad: tipos de emergencias como: incendios, rescates, manejo de materiales peligrosos, emergencias prehospitalarias, eventos hídricos y meteorológicos.
- estacion: Se transformo a tipo categórica y se definió un número a cada estación de acuerdo a x1 = 1 ……. x26 = 26
- tipo_dia_festivo: tipo día en el que se atendió la emergencia clasificado por: regular, feriado, puente feriado

Los modelos de aprendizaje automático generalmente requieren que estas variables estén en formato numérico. Por ello, se reemplazaron los valores de texto por códigos por ejemplo se asignó números consecutivos y se utilizaron técnicas de codificación como “One-Hot Encoding” que consiste en la es la creación columnas binarias para cada categoría esta técnica fue aplicada a las variables especialidad y tipo_dia_festivo. De este modo, el modelo puede distinguir las diferentes especialidades, días festivos y estaciones sin malinterpretar el texto como un valor ordinal o una simple cadena.

4.4.6 Selección de los Datos

Como resultado de las actividades de limpieza y agrupación iniciales, se generó un nuevo conjunto de datos más compacto que el conjunto de datos original y alineado con los objetivos de predicción de emergencias por estación y especialidad. Este este nuevo conjunto de datos incluye las siguientes columnas:

- anio
- mes
- diadelmes
- estacion
- emergencias_atendidas
- dia_de_la_semana
- esp_ASISTENCIA ESPECIALIZADA APH
- esp_ASISTENCIA ESPECIALIZADA SINIESTROS
- esp_EVENTOS HIDRICOS Y METEOROLÓGICOS
- esp_INCENDIOS
- esp_MATERIALES PELIGROSOS
- esp_PREHOSPITALARIAfu
- esp_RESCATE
- festivo_Día regular
- festivo_Feriado
- festivo_Puente feriado

La selección de estas columnas se fundamenta en su relevancia estadística y su relación directa con la aparición de emergencias. Por un lado, anio, mes y diadelmes

capturan la estacionalidad y las tendencias temporales que afectan la frecuencia de emergencias, las variables especialidad y estación representan la naturaleza específica de la emergencia y la ubicación que brinda respuesta, ambos elementos son necesarios para pronosticar cuántas atenciones se pueden generar en un día determinado. Por último, la variable emergencias_atendidas es la variable objetivo dependiente que queremos predecir.

Luego de este proceso el nuevo dataset ya no contine variables que no aportan valor predictivo y concentra las variables esenciales. Este nuevo conjunto de datos más estructurado facilitará la etapa de modelado y también mejora la interpretación de los resultados, esto debido a que cada variable está directamente asociada con el análisis de ocurrencia y distribución de emergencias en el Distrito Metropolitano de Quito.

4.5 Desarrollo del modelo predictivo

4.5.1 Selección y entrenamiento del modelo

Una vez que contamos con un conjunto de datos depurado y organizado, el siguiente paso es seleccionar el algoritmo de Machine Learning más adecuado y entrenarlo para que pueda anticipar la cantidad de emergencias por día, estación y especialidad. En el Marco Teórico del presente trabajo de titulación se muestra una tabla comparativa de modelos predictivos y como resultado de esta comparación se optó por el modelo Random Forest debido a su capacidad para:

- Manejar múltiples variables numéricas y categóricas.
- Identificar patrones no lineales sin requerir grandes ajustes de hiperparámetros.

- Reducir el riesgo de sobreajuste debido a que promedia los resultados de varios árboles de decisión.

El siguiente paso es la división del conjunto de datos en entrenamiento y prueba (Train/Test Split) para comenzar, dividimos nuestro conjunto en dos partes: entrenamiento 80% de los registros y para pruebas el 20% restante. De esta forma el modelo aprende a partir de la mayoría de la información disponible, mientras reservamos un subconjunto independiente que el modelo predictivo no puede acceder durante la etapa de aprendizaje. Este conjunto de pruebas nos permite comprobar, qué tan bien predice sobre ejemplos que nunca antes se presentaron.

```
# Entrenammiento del modelo
# se seleccionas las variables dependiente e independientes
X = df_final.drop('emergencias_atendidas', axis=1)
y = df_final['emergencias_atendidas']

# Dividir en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=42
)
```

Figura 19 División de conjunto de datos en entrenamiento y pruebas

A diferencia de los modelos basados en series de tiempo, donde la secuencia temporal es fundamental, en este caso se optó por una partición aleatoria (utilizando la función `train_test_split`) porque la ocurrencia de emergencias no depende exclusivamente del tiempo, sino de múltiples factores externos como el clima, eventos masivos o la infraestructura de la ciudad. Esta decisión permite que el modelo aprenda patrones generales en lugar de limitarse a un periodo específico, mejorando su capacidad para identificar tendencias en distintos momentos.

Sin embargo, si en el futuro Bomberos requiere predecir emergencias en el tiempo, podría aplicar una validación temporal estricta, utilizando datos históricos para entrenar y registros recientes para probar también, podría explorar una validación cruzada para series de tiempo, lo que permitiría evaluar la estabilidad del modelo en diferentes periodos y ajustar su precisión en función de los cambios en la ciudad.

Continuamos con el ajuste de Hiperparámetros Parámetros como el número de árboles (n_estimators), la máxima profundidad (max_depth) o la cantidad mínima de muestras para dividir un nodo (min_samples_split) se afinan con cuidado para lograr un balance adecuado entre precisión y rendimiento computacional. Para esta tarea se emplea técnicas como GridSearchCV o RandomizedSearchCV, incluidas en scikit-learn, las cuales permiten evaluar múltiples configuraciones y encontrar aquella que ofrezca el mejor desempeño global.

Para encontrar la combinación óptima de estos valores, empleamos GridSearchCV, que prueba varias combinaciones y selecciona la mejor en base al error cuadrático medio (MSE). La búsqueda se realizó con los siguientes valores:

```
param_grid = {
    'n_estimators': [50, 100, 150],
    'max_depth': [10, 15, 20],
    'min_samples_split': [2, 4, 6]
}
```

Luego, se ejecutó la optimización con validación cruzada de 3 pliegues (cv=3), lo que significa que cada combinación de hiperparámetros se probó en diferentes subconjuntos del conjunto de entrenamiento:

```
grid_search = GridSearchCV(
    estimator=RandomForestRegressor(random_state=42),
    param_grid=param_grid,
    scoring='neg_mean_squared_error',
    cv=3,
    n_jobs=-1
)
grid_search.fit(X_train, y_train)
```

Después de completar la búsqueda, los mejores hiperparámetros encontrados fueron:

```
{'n_estimators': 100, 'max_depth': 15, 'min_samples_split': 4}
```

Una vez que determinamos la mejor combinación de hiperparámetros, entrenamos la versión definitiva del modelo con todo el conjunto de entrenamiento, maximizando así el uso de la información disponible y elevando la calidad de las predicciones.

```python
# Definición de modelo
rf = RandomForestRegressor(random_state=42)

param_grid = {
    'n_estimators': [50, 100, 150],
    'max_depth': [10, 15, 20],
    'min_samples_split': [2, 4, 6]
}

grid_search = GridSearchCV(estimator=rf,
                           param_grid=param_grid,
                           scoring='neg_mean_squared_error',
                           cv=3,
                           n_jobs=-1)

grid_search.fit(X_train, y_train)

print("Mejores Hiperparámetros:", grid_search.best_params_)

best_params = grid_search.best_params_
rf_final = RandomForestRegressor(
    n_estimators=best_params['n_estimators'],
    max_depth=best_params['max_depth'],
    min_samples_split=best_params['min_samples_split'],
    random_state=42
)

rf_final.fit(X_train, y_train)
```

Figura 20 Definición de modelo y configuración de hiperparámetros

Como línea base para comparar lo resultados obtenidos por el modelo Random forest se implementó los modelos:

- Regresión Lineal
- Redes Neuronales

A continuación, se presenta el código de implementación de los algoritmos citados anteriormente:

```python
# Generar algoritmo de regresión lineal como línea base
# para comparar resultados con modelo random forest
lr = LinearRegression()

# Entrenar modelo
lr.fit(X_train, y_train)

# Hacer predicciones
y_pred_lr = lr.predict(X_test)

# Evaluar el modelo
mae_lr = mean_absolute_error(y_test, y_pred_lr)
mse_lr = mean_squared_error(y_test, y_pred_lr)
rmse_lr = np.sqrt(mse_lr)
r2_lr = r2_score(y_test, y_pred_lr)

# Mostrar resultados
print("Resultados de Regresión Lineal:")
print(f"MAE: {mae_lr:.2f}")
print(f"MSE: {mse_lr:.2f}")
print(f"RMSE: {rmse_lr:.2f}")
print(f"R²: {r2_lr:.2f}")
```

```
Resultados de Regresión Lineal:
MAE: 0.89
MSE: 1.56
RMSE: 1.25
R²: 0.33
```

Figura 21 Código de implementación de algoritmo de regresión línea como línea base para evaluación de resultados con random forest

```
# Generar redes neuronales como línea base para comparar
# resultados con modelo random forest
# Escalar datos
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Definir el modelo de red neuronal
nn = MLPRegressor(hidden_layer_sizes=(50, 30), activation='relu',
                  solver='adam', max_iter=500, random_state=42)

# Entrenar modelo
nn.fit(X_train_scaled, y_train)

# Hacer predicciones
y_pred_nn = nn.predict(X_test_scaled)

# Evaluar el modelo
mae_nn = mean_absolute_error(y_test, y_pred_nn)
mse_nn = mean_squared_error(y_test, y_pred_nn)
rmse_nn = np.sqrt(mse_nn)
r2_nn = r2_score(y_test, y_pred_nn)

# Mostrar resultados
print("Resultados de Redes Neuronales:")
print(f"MAE: {mae_nn:.2f}")
print(f"MSE: {mse_nn:.2f}")
print(f"RMSE: {rmse_nn:.2f}")
print(f"R²: {r2_nn:.2f}")
```

```
Resultados de Redes Neuronales:
MAE: 0.80
MSE: 1.30
RMSE: 1.14
R²: 0.44
```

Figura 22 Código de implementación de algoritmo de redes neuronales como línea base para evaluación de resultados con random forest

4.6 Validación y Evaluación del Modelo

La validación y evaluación del modelo constituyen pasos esenciales para determinar su capacidad de generalización. Entre las métricas más utilizadas se encuentran:

1. **MAE (Mean Absolute Error)**

   Mide el error promedio entre predicciones y valores reales. Un MAE de 2 significa que, en promedio, el modelo se desvía 2 emergencias por día respecto a lo ocurrido.

2. **RMSE (Root Mean Squared Error)**

   Penaliza de forma más intensa los errores grandes, ya que se basa en el cuadrado de las diferencias. Un RMSE de 4 implica que, si bien el modelo puede acertar la mayoría de las predicciones con poca desviación, algunos escenarios podrían presentar diferencias mayores.

3. **R^2 Score (Coeficiente de Determinación)**

   Mide cuánta varianza de la variable objetivo (emergencias_atendidas) es explicada por el modelo. Un valor de R^2 cercano a 1 indica un excelente ajuste, mientras que valores cercanos a 0 sugieren que el modelo no explica gran parte de la variabilidad.

A continuación los resultados obtenidos de la evaluación del modelo random forest:

```
# Validación y evaluación del modelo
y_pred = rf_final.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("Resultados de la Evaluación del Modelo:")
print(f"MAE  (Mean Absolute Error):      {mae:.2f}")
print(f"MSE  (Mean Squared Error):       {mse:.2f}")
print(f"RMSE (Root Mean Squared Error):  {rmse:.2f}")
print(f"R^2  (Coeficiente de Determinación): {r2:.2f}")
```

```
Resultados de la Evaluación del Modelo:
MAE  (Mean Absolute Error):      0.72
MSE  (Mean Squared Error):       1.11
RMSE (Root Mean Squared Error):  1.05
R^2  (Coeficiente de Determinación): 0.50
```

Figura 23 Validación y evaluación del modelo random forest

4.6.1 Gráficos de Evaluación

Gráfico de Predicción vs. Real

Permite ver si las estimaciones se alinean con lo ocurrido. Idealmente, deberían quedar cerca de la diagonal y = x.

```
[64] # Interpretación de los resultados

     # Gráfico de Predicción vs. Valor Real
     plt.figure(figsize=(6,6))
     plt.scatter(y_test, y_pred, alpha=0.5)
     plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
     plt.xlabel('Valor Real')
     plt.ylabel('Predicción')
     plt.title('Predicción vs. Valor Real')
     plt.show()
```

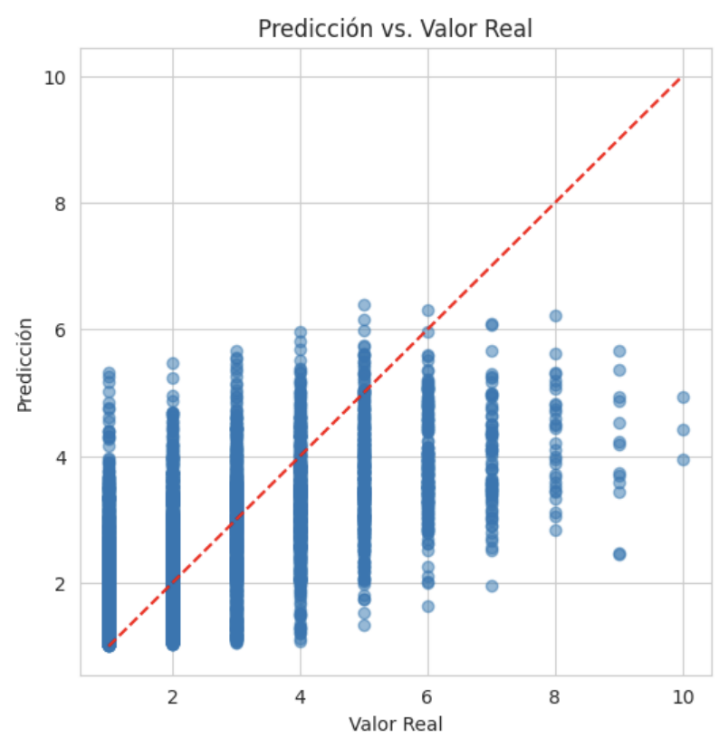


Figura 24 Gráfico de predicción vs. valor real del modelo random forest

En este gráfico, el eje horizontal muestra los valores reales de emergencias_atendidas, mientras que en el eje vertical se representan las predicciones

del modelo. La línea diagonal roja (y = x) representa la coincidencia perfecta entre predicción y realidad, cuanto más se acerquen los puntos a esta diagonal, más exacto será el resultado. Si la mayoría de puntos se encuentra alrededor de la línea, significa que el modelo está haciendo un buen trabajo al estimar la cantidad de emergencias.

Matriz de Errores / Distribución de Errores

Muestra cómo se dispersan los valores residuales (diferencia entre lo real y la predicción). Un sesgo constante o colas largas podrían señalar la necesidad de refinar el modelo o agregar más variables explicativas.

```
[65] # Distribución de Errores (Residuals)
     residuals = y_test - y_pred  # error = real - pred
     plt.figure(figsize=(6,4))
     sns.histplot(residuals, kde=True, color='blue', bins=30)
     plt.title('Distribución de los Errores (Residuals)')
     plt.xlabel('Error')
     plt.ylabel('Frecuencia')
     plt.show()
```

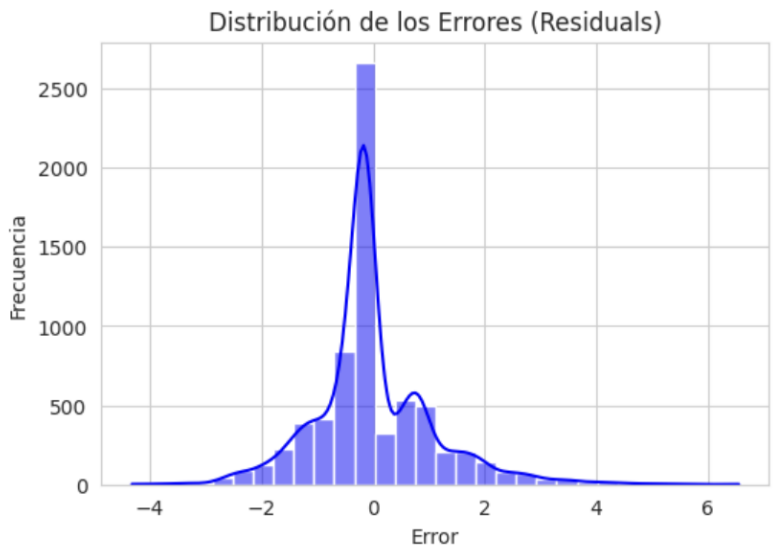


Figura 25 Gráfica de distribución de errores, muestra diferencia entre valores reales y la predicción

Los errores se obtienen al restar la predicción al valor real se representa con un histograma en el que se observa la frecuencia de cada error y una línea de densidad para resaltar su forma general. Lo ideal es que esta distribución se ubique alrededor de cero, lo que indica un modelo sin sesgo aparente y con errores mayormente equilibrados.

Residuos vs. Valores reales

Representa en el eje horizontal el valor real o el dato observado y en el vertical el residuo correspondiente la diferencia entre el valor real y la predicción del modelo, el gráfico muestra si los errores se reparten uniformemente o si se concentran en algún rango o patrón que se puede corregir agregando nuevas variables.

```
[66] # Análisis de Errores vs. Valores Reales
     plt.figure(figsize=(6,4))
     plt.scatter(y_test, residuals, alpha=0.5)
     plt.hlines(y=0, xmin=y_test.min(), xmax=y_test.max(), colors='r', linestyles='--')
     plt.xlabel('Valor Real')
     plt.ylabel('Residual (Real - Predicción)')
     plt.title('Residuales vs. Valor Real')
     plt.show()
```

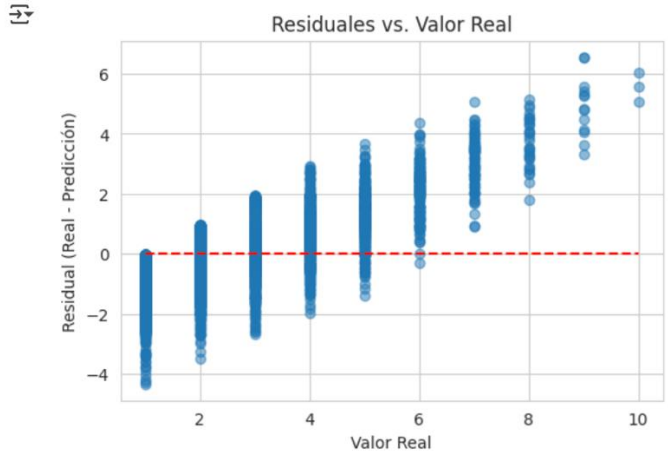


Figura 26 Gráfico de errores vs. valores reales, muestra el valor real y la predicción del modelo

Este diagrama ayuda a ver si el tamaño de los errores aumenta o disminuye en función del valor real de las emergencias atendidas. Si se llegaran apreciar errores más grandes a medida que se eleva el número de emergencias, podría indicar que el modelo necesita ajustarse mejor a situaciones extremas, ya sea añadiendo más variables o aplicando transformaciones que capten de manera adecuada eventos poco comunes.

Las visualizaciones presentadas permiten identificar patrones que pueden pasar desapercibidos si solo nos basamos en métricas numéricas como MAE, RMSE o R^2. Si se observa que los errores positivos se concentran en un rango específico de valores reales, esto podría indicar que el modelo suele subestimar las emergencias en ciertos niveles o por el contrario, sobreestimarlas en otros casos.

4.6.1 Interpretación de resultados

Luego de entrenar y evaluar el modelo random forest y los dos modelos que servirán de base para comprar los resultados se obtuvo cuatro métricas que reflejan qué tan cercanas son las predicciones, en ese sentido, a continuación se presenta la tabla con los resultados de la evaluación de los modelos:

Modelo	MAE	MSE	RMSE	R^2
Regresión Lineal	0.89	1.56	1.25	0.33
Redes Neuronales	0.80	1.30	1.14	0.44
Random Forest	0.74	1.16	1.08	0.50

Figura 27 Tabla de resultados de evaluación de modelos: regresión lineal, redes neuronales y random forest

La regresión lineal es la opción con menor desempeño, su error promedio (MAE: 0.89) indica que, en general, se equivoca en casi una emergencia atendida por día, lo que puede ser significativo en un análisis de este tipo además, su capacidad explicativa (R^2: 0.33) nos dice que solo el 33% de la variabilidad en los datos está siendo explicada por el modelo, lo que deja muchas predicciones sin una base sólida. Esto sugiere que la relación entre las variables no es completamente lineal.

Las redes neuronales muestran una mejora significativa respecto a la regresión lineal, con un error más bajo (MAE: 0.80) y una mejor capacidad de predicción (R^2: 0.44). Esto significa que el modelo es capaz de capturar más patrones en los datos, pero todavía no alcanza la precisión de Random Forest.

De los tres modelos evaluados, Random Forest tuvo el mejor desempeño en todos los aspectos, el error promedio (MAE: 0.74) fue el más bajo y su capacidad explicativa (R^2: 0.50) fue la más alta, lo que significa que explica mejor la variabilidad en los datos y genera predicciones más precisas. Esto se debe a que Random Forest es un modelo basado en múltiples árboles de decisión, lo que le permite identificar patrones más complejos y manejar mejor los valores atípicos.

El R^2 de 0.50 indica que el modelo explica el 50% de la variabilidad en los datos, lo que puede parecer moderado, pero no significa que sea un modelo deficiente. Hay varios factores que podrían estar limitando su precisión, por ejemplo, no se consideraron variables como el clima, eventos masivos, estas variables podrían influir en la cantidad de emergencias atendidas además, aun que Random Forest es un modelo robusto tiene sus limitaciones ya que no extrapola bien más allá de los datos de

entrenamiento, por lo que podría tener problemas al predecir eventos poco frecuentes. Además, la naturaleza impredecible de las emergencias, como accidentes o desastres naturales, introduce una variabilidad que ningún modelo puede anticipar completamente. En este sentido, más que ver el R^2 de 0.50 como un problema, es importante entenderlo como una señal de que el fenómeno estudiado es complejo y que hay factores externos influyendo en los resultados. Para mejorar la precisión, se podría considerar incluir más variables relevantes y explorar otros modelos que capturen relaciones más profundas en los datos.

Dado que Bomberos no cuenta con estudios previos sobre la predicción de emergencias mediante métodos de Machine Learning , no existe un punto de referencia con el cual comparar el MAE de 0.72, en ese caso lo más práctico es trabajar con los resultados obtenidos y evaluar directamente su impacto en la planificación operativa. El nivel de error de 0.72 emergencias por día y estación puede ser irrelevante en estaciones con alta demanda de atención, pero en estaciones con pocas emergencias diaria esta variabilidad podría ser más significativa, debido a que no existe un criterio establecido para medir la tolerancia de error lo mejor es utilizar estos resultados como una línea base para analizar su efecto y, si es necesario, hacer ajustes para mejorar la precisión del modelo según las necesidades reales.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5. Conclusiones y Recomendaciones

5.1 Conclusiones

1. El uso de un modelo predictivo basado en Machine Learning permitió pasar de una planificación reactiva a una más anticipada. Esto facilitó la identificación de patrones y tendencias en la ocurrencia de emergencias, brindando al Cuerpo de Bomberos del Distrito Metropolitano de Quito (CB-DMQ) la posibilidad de asignar recursos de manera más efectiva.
2. La metodología CRISP-DM demostró ser un enfoque sólido y organizado para manejar todo el ciclo de vida del proyecto, desde la definición del problema hasta la evaluación y validación del modelo. Su carácter iterativo contribuyó a refinar continuamente el análisis y mejorar los resultados.
3. La limpieza y la selección cuidadosa de variables, así como la transformación de datos, resultaron adecuadas para garantizar la calidad y la fiabilidad del modelo predictivo. El retirar información redundante y concentrarse en aquellas variables con mayor aporte estadístico optimizó tanto el entrenamiento como la interpretación final.
4. Pese a que el modelo alcanzó un desempeño moderado, existió un margen de error considerable en escenarios puntuales como días festivos o condiciones climáticas extremas. Esto pone de refleja la importancia de incorporar nuevas variables y ajustar de manera continua la infraestructura analítica para mejorar la precisión.
5. La experiencia obtenida en el Distrito Metropolitano de Quito es potencialmente escalable y replicable en otras ciudades con características similares,

contribuyendo así a la formación de un enfoque más proactivo en la gestión de emergencias.

5.2 Recomendaciones

1. Integrar nuevas fuentes de datos como factores climáticos y estacionales con mayor detalle, se recomienda incluir variables meteorológicas como temperatura, humedad relativa, radiación solar, velocidad del viento y precipitaciones en el DMQ, con ello, el modelo podría reducir su margen de error en escenarios atípicos.
2. Para que el modelo sea una herramienta útil para Bomberos el mismo debe mantenerse actualizado con datos más recientes de emergencias y crear un proceso de realimentación en el que las salidas del modelo se usen para mejorar la asignación de recursos, mientras se registra información de los resultados operativos reales ajustando parámetros y detectando posibles desviaciones en la precisión.
3. Tomando en cuenta que las necesidades de la ciudad pueden cambiar con el tiempo (demografía, urbanización, clima), se aconseja realizar evaluaciones periódicas y reentrenamientos del modelo para garantizar que mantenga su precisión, esto asegurará que el Bomberos cuente con un sistema de predicción robusto y alineado con las realidades actuales del DMQ.
4. Aunque Random Forest ofrece un rendimiento satisfactorio, puede ser útil evaluar otros modelos como XGBoost, LightGBM y contrastar sus resultados con el desempeño del Random Forest actual, de esta forma se podría mejorar los aciertos e identificar los casos más difíciles de predecir como emergencias con

baja frecuencia o eventos imprevistos. Un análisis comparativo permitiría a Bomberos seleccionar el mejor enfoque según sus necesidades operativas.

BIBLIOGRAFÍA

Aggarwal, C. C. (2017). Outlier analysis (2.a ed.). Springer. https://doi.org/10.1007/978-3-319-47578-3

Akkihal, A. R. (2006). Inventory pre-positioning for humanitarian operations (Tesis de maestría). Massachusetts Institute of Technology, recuperado de https://dspace.mit.edu/handle/1721.1/36318

Ashmore, R., Calinescu, R., & Paterson, C. (2019). Assuring the machine learning lifecycle: Desiderata, methods, and challenges. Reliability Engineering & System Safety, 196, 106740. https://doi.org/10.48550/arXiv.1905.04223

Baquero, O. S., Santana, L. M. R., & Chiaravalloti-Neto, F. (2018). Dengue forecasting in São Paulo city with generalized additive models, artificial neural networks and seasonal autoregressive integrated moving average models. PLOS ONE, 13(4), e0195065. https://doi.org/10.1371/journal.pone.0195065

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. CRISP-DM Consortium. https://www.crisp-dm.org/

City-Facts. (n.d.). Evolución de la población de Quito. Recuperado el 11 de diciembre de 2024, de https://es.city-facts.com/quito/population

Danalytics. (2023). Proyectos de analítica de datos con CRISP-DM. Recuperado de https://www.danalyticspro.com/proyectos-analitica-de-datos-con-crispdm/

Domingos, P. (2015). The master algorithm: How the quest for the ultimate learning machine will remake our world. Basic Books.

Geron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow (2.a ed.). O'Reilly Media.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction. Springer Science & Business Media.

Instituto de Ingeniería del Conocimiento (IIC). (2021). La metodología CRISP-DM en ciencia de datos. Recuperado de https://www.iic.uam.es/innovacion/metodologia-crisp-dm-ciencia-de-datos/

Kawasaki, T., & Saito, H. (2019). Predicting traffic accident frequency using logistic regression. Transportation Research Part F: Traffic Psychology and Behaviour, 62, 133-142.

Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2020). Fundamentals of machine learning for predictive data analytics: Algorithms, worked examples, and case studies (2.a ed.). MIT Press.

Kumar, A., Sharma, K., & Yadav, A. K. (2021). A review of COVID-19 prediction models based on machine learning approaches. International Journal of Data Science and Analysis, 7(2), 55-63.

Martínez, J., & Pérez, L. (2018). Aplicación de CRISP-DM en el sector bancario: Caso Banco Santander. Revista Española de Ciencia de Datos, 10(2), 45-60.

Meissner, A., Körner, M., & Boettcher, B. (2018). Application of decision tree-based prediction models in emergency care for patients with cardiac symptoms. Journal of Medical Systems, 42(8), 152.

Montgomery, D. C., & Runger, G. C. (2006). Introducción a la regresión lineal (2.a ed.). McGraw-Hill.

Raschka, S., & Mirjalili, V. (2019). Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing.

Sharples, J. J., McRae, R. H., & Weber, R. O. (2019). A decision tree model for predicting bushfire severity and spread. International Journal of Wildland Fire, 28(2), 1-12.

Shmueli, G., Bruce, P. C., Yahav, I., Patel, N. R., & Lichtendahl, K. C. (2017). Data mining for business analytics: Concepts, techniques, and applications in R. Wiley.

Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., ... & Collins, R. (2015). UK Biobank: An open access resource for identifying the causes of a wide range of complex diseases of middle and old age. PLOS Medicine, 12(3), e1001779. https://doi.org/10.1371/journal.pmed.1001779

Venegas Zapata, J. G. (2020). Análisis estadístico de datos meteorológicos mensuales y diarios en el periodo 2006-2018 para la determinación de variabilidad

climática y cambio climático en el Distrito Metropolitano de Quito (Tesis de maestría). Universidad Andina Simón Bolívar, Quito, Ecuador. Recuperado de https://repositorio.uasb.edu.ec/handle/10644/7482

Wickham, H., & Grolemund, G. (2017). R for data science. O'Reilly Media.

Yang, D., Liu, Q., & Zhao, X. (2021). Crime prediction and pattern analysis using artificial neural networks. Journal of Criminal Justice, 73, 101761.

ANEXO 1 IMPLEMENTACIÓN DE MODELO PREDICTIVO

```
# Importar librerías
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# Configuración de visualización
sns.set_style("whitegrid")
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LinearRegression
from sklearn.neural_network import MLPRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
```

```
# Cargar dataset
df = pd.read_csv('data_emergencias_2022-2023-2024.csv', encoding="latin1")

# Mostrar información básica del dataset
print(df.info())
# Mostrar primeras 5 filas
print(df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 76236 entries, 0 to 76235
Data columns (total 13 columns):
```

```
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   numero_parte                76236 non-null  int64
 1   anio                        76236 non-null  int64
 2   mes                         76236 non-null  int64
 3   diadelmes                   76236 non-null  int64
 4   codigo_especialidad         76236 non-null  object
 5   especialidad                76236 non-null  object
 6   codigo_subespecilidad       76236 non-null  int64
 7   subespecilidad              76236 non-null  object
 8   forma_aviso                 76232 non-null  object
 9   numero_vehiculos            76224 non-null  float64
 10  numero_personal_operativo   76224 non-null  float64
 11  estacion                    76236 non-null  object
 12  canton                      76236 non-null  object
dtypes: float64(2), int64(5), object(6)
memory usage: 7.6+ MB
None
   numero_parte  anio  mes  diadelmes   codigo_especialidad  \
0        506653  2024   12         31  40
1        506651  2024   12         31  50.
2        506649  2024   12         31  40
3        506648  2024   12         31  40
4        506647  2024   12         31  40

                       especialidad  codigo_subespecilidad  \
0                      PREHOSPITALARIA                    287
1  EVENTOS HIDRICOS Y METEREOLÃ□GICOS                    335
2                      PREHOSPITALARIA                    287
3                      PREHOSPITALARIA                    287
4                      PREHOSPITALARIA                    336

                             subespecilidad           forma_aviso  \
0                                   TRAUMA  CIU
1  LIMPIEZA DE OBSTÃ□CULOS EN VÃ□A PÃ□BLICA  CIU
2                                   TRAUMA  CIU
3                                   TRAUMA  MSP
4                      EMERGENCIA CLÃ□NICA  CIU

   numero_vehiculos  numero_personal_operativo estacion canton
0               1.0                        2.0       X5  QUITO
1               3.0                        4.0       X6  QUITO
2               1.0                        2.0       X3  QUITO
3               1.0                        2.0      X26  QUITO
4               1.0                        2.0      X15  QUITO
```

```
#Verificar valores faltantes:
df.isnull().sum()
```

	0
numero_parte	0
anio	0
mes	0
diadelmes	0
codigo_especialidad	0
especialidad	0
codigo_subespecilidad	0
subespecilidad	0
forma_aviso	4
numero_vehiculos	12
numero_personal_operativo	12
estacion	0
canton	0

dtype: int64

```
# Tratar datos faltantes
# Imputación con la media en variables:
# 'numero_vehiculos' y 'numero_personal_operativo'
df['numero_vehiculos'].fillna(df['numero_vehiculos'].mean(), inplace=True)
df['numero_personal_operativo'].fillna(df['numero_personal_operativo'].mea
n(), inplace=True)

# Imputación con la moda en variable: 'forma_aviso'
moda_forma_aviso = df['forma_aviso'].mode()[0]
df['forma_aviso'].fillna(moda_forma_aviso, inplace=True)

# Verificamos nuevamente valores faltantes
print(df.isnull().sum())
```

```
numero_parte          0
```

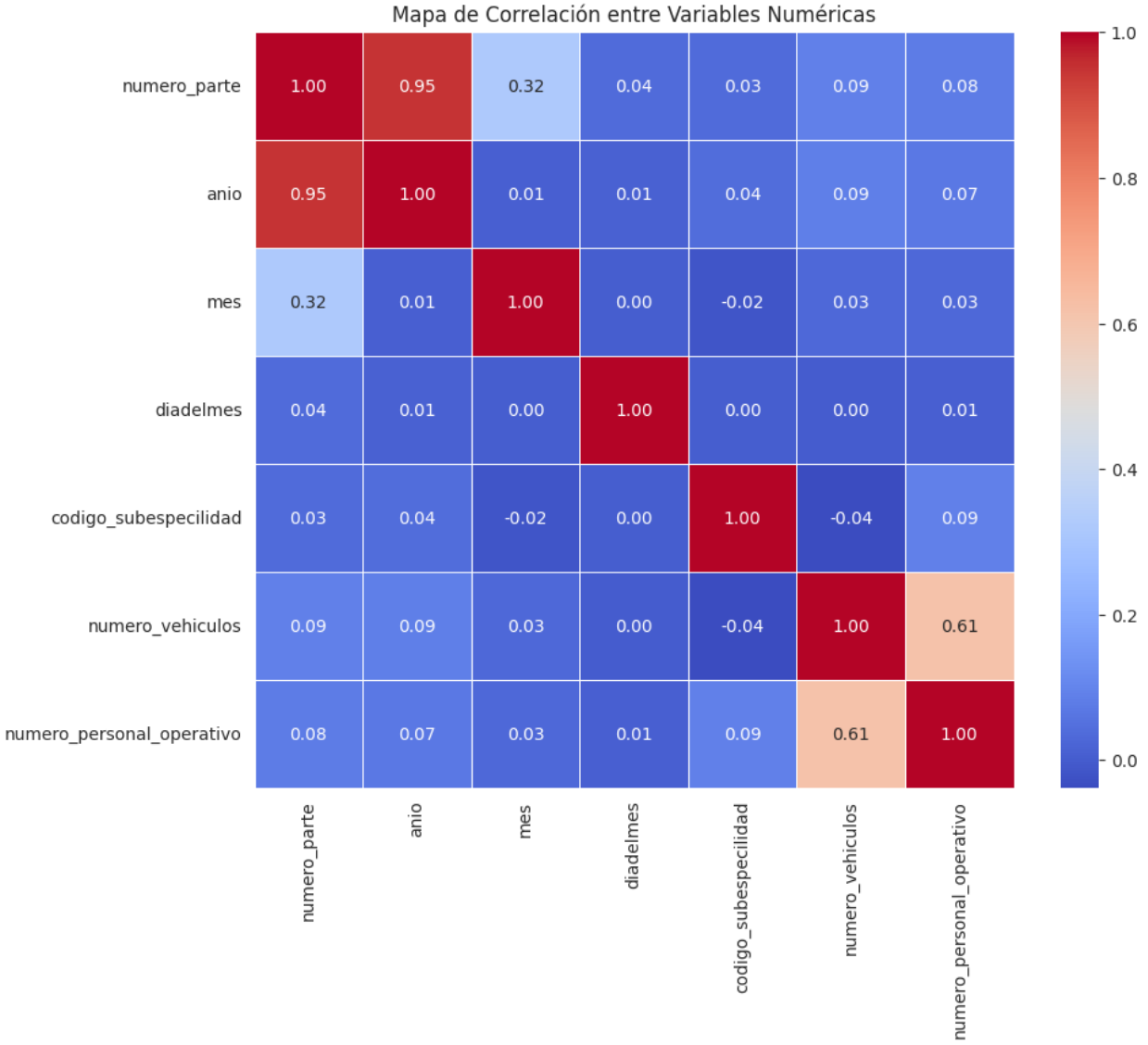
```
anio                    0
mes                     0
diadelmes               0
codigo_especialidad     0
especialidad            0
codigo_subespecilidad   0
subespecilidad          0
forma_aviso             0
numero_vehiculos        0
numero_personal_operativo  0
estacion                0
canton                  0
dtype: int64
```

```
# Análisis exploratorio de datos
print("Resumen general de los datos:")
print(df.describe(include="all"))
```

```
Resumen general de los datos:
         numero_parte          anio           mes     diadelmes  \
count    76236.000000  76236.000000  76236.000000  76236.000000
unique            NaN           NaN           NaN           NaN
top               NaN           NaN           NaN           NaN
freq              NaN           NaN           NaN           NaN
mean    445563.853665   2022.991238      6.636812     15.754683
std      38018.911257      0.814589      3.379808      8.819203
min     381747.000000   2022.000000      1.000000      1.000000
25%     411554.750000   2022.000000      4.000000      8.000000
50%     444841.500000   2023.000000      7.000000     16.000000
75%     481343.500000   2024.000000      9.000000     23.000000
max     506653.000000   2024.000000     12.000000     31.000000

         codigo_especialidad     especialidad  codigo_subespecilidad  \
count                  76236            76236           76236.000000
unique                     7                7                    NaN
top     40                    PREHOSPITALARIA                    NaN
freq                   50846            50846                    NaN
mean                     NaN              NaN             310.395679
std                      NaN              NaN              24.187376
min                      NaN              NaN             287.000000
25%                      NaN              NaN             287.000000
50%                      NaN              NaN             296.000000
75%                      NaN              NaN             336.000000
max                      NaN              NaN             403.000000

       subespecilidad           forma_aviso  numero_vehiculos  \
count           76236                 76236      76236.000000
unique             30                    16               NaN
top            TRAUMA  CIU                                NaN
freq            29731                 58211               NaN
mean              NaN                   NaN          1.245553
std               NaN                   NaN          0.708640
```

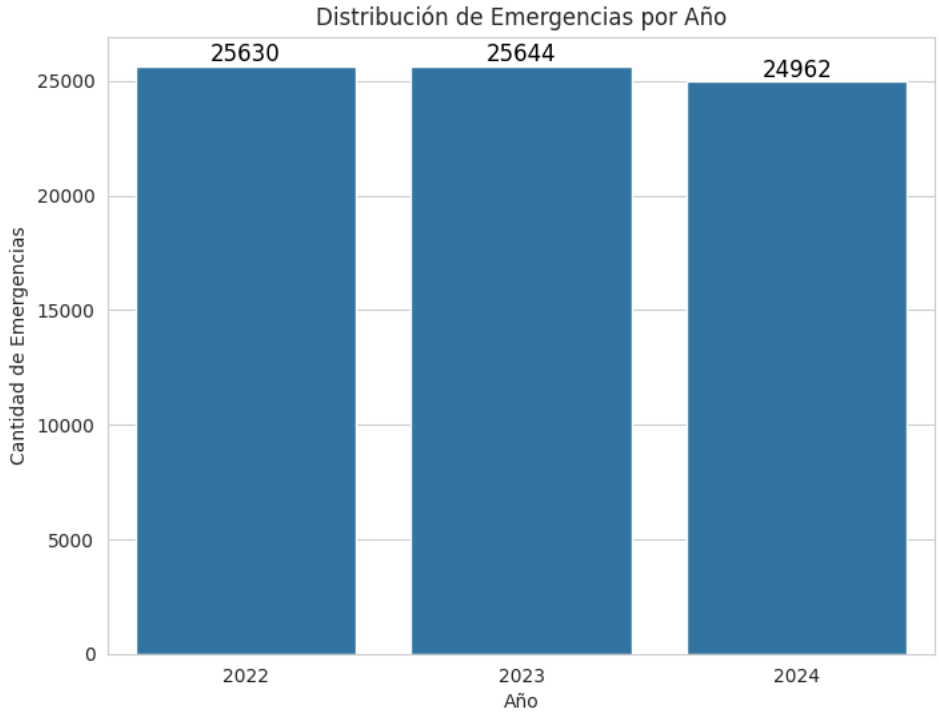
```
min                NaN                    NaN          1.000000
25%                NaN                    NaN          1.000000
50%                NaN                    NaN          1.000000
75%                NaN                    NaN          1.000000
max                NaN                    NaN         30.000000

        numero_personal_operativo estacion canton
count                76236.000000    76236  76236
unique                        NaN       26     13
top                           NaN       X6  QUITO
freq                          NaN     7318  76007
mean                     3.145925      NaN    NaN
std                      1.822920      NaN    NaN
min                      0.000000      NaN    NaN
25%                      2.000000      NaN    NaN
50%                      3.000000      NaN    NaN
75%                      4.000000      NaN    NaN
max                     45.000000      NaN    NaN
```

```
# Variables numericas para correlación
df_numeric = df.select_dtypes(include=['number'])

# Generar matriz de correlación
correlation_matrix = df_numeric.corr()

# Crear gráfico heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=0.5)
plt.title("Mapa de Correlación entre Variables Numéricas")
plt.show()
```

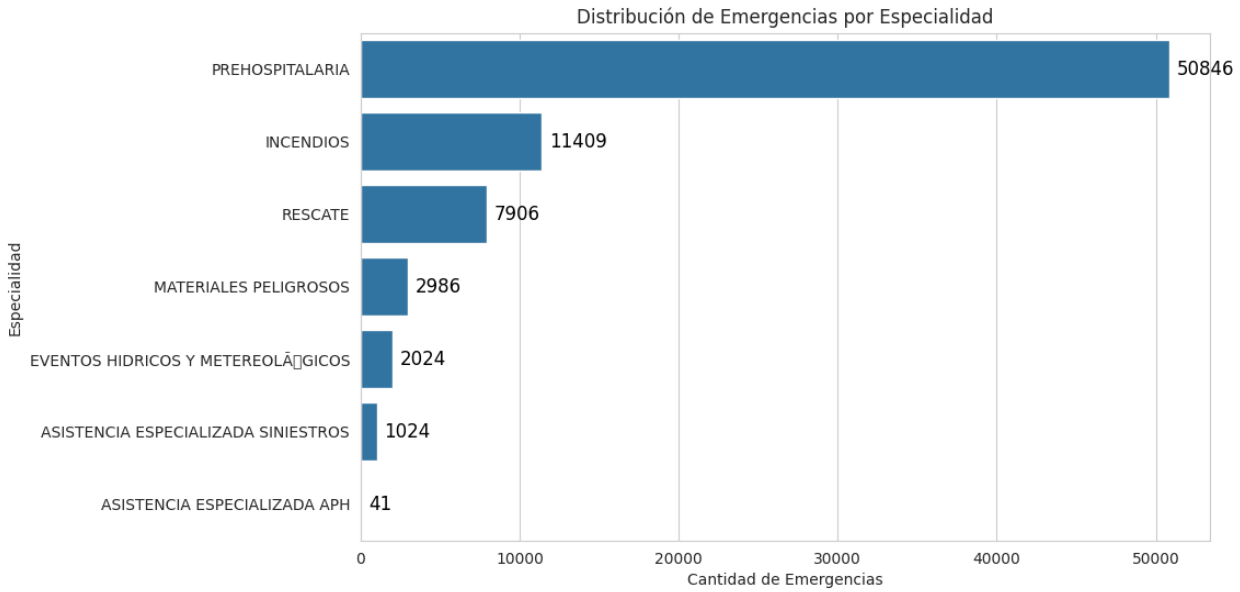



```
# Emergencias por año
plt.figure(figsize=(8, 6))
ax = sns.countplot(data=df, x="anio")
for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}',  # Texto con el valor de la
barra
                (p.get_x() + p.get_width() / 2, p.get_height()),  #
Posición del texto
                ha='center', va='bottom', fontsize=12, color='black',
fontweight='normal')

plt.title("Distribución de Emergencias por Año")
plt.xlabel("Año")
plt.ylabel("Cantidad de Emergencias")
```

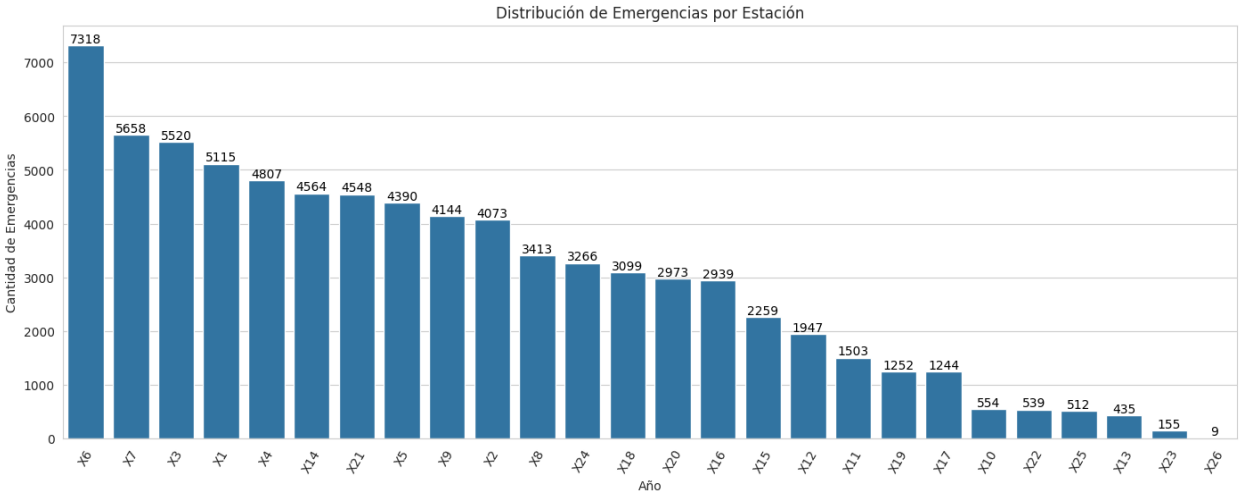
```
plt.show()
```




```
# Distribución de emergencias por especialidad
plt.figure(figsize=(10, 6))
ax = sns.countplot(data=df, y="especialidad",
order=df["especialidad"].value_counts().index)

# Agregar etiquetas con los valores en cada barra
for p in ax.patches:
    ax.annotate(f'{int(p.get_width())}',  # Texto con el valor de la barra
                (p.get_width(), p.get_y() + p.get_height() / 2),  #
Posición del texto
                ha='left', va='center', fontsize=12, color='black',
fontweight='normal', xytext=(5, 0), textcoords='offset points')

plt.title("Distribución de Emergencias por Especialidad")
plt.xlabel("Cantidad de Emergencias")
plt.ylabel("Especialidad")
plt.show()
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151:
UserWarning: Glyph 147 (\x93) missing from font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```




```
# Emergencias por estación
plt.figure(figsize=(17, 6))
ax = sns.countplot(data=df, x="estacion",
order=df["estacion"].value_counts().index)
for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}',  # Texto con el valor de la
barra
                (p.get_x() + p.get_width() / 2, p.get_height()),  #
Posición del texto
                ha='center', va='bottom', fontsize=10, color='black',
fontweight='normal')

plt.title("Distribución de Emergencias por Estación")
plt.xlabel("Año")
plt.ylabel("Cantidad de Emergencias")
plt.xticks(rotation=60)
plt.show()
```




```
# Filtrado de Datos
# Lista de estaciones a excluir
estaciones_excluir = ['X26', 'X10', 'X22', 'X25', 'X13', 'X23']

# Eliminar estaciones
df = df[~df['estacion'].isin(estaciones_excluir)]

# Verificar que las estaciones fueron eliminadas
print(df['estacion'].unique())  # Muestra las estaciones restantes
```

```
['X5' 'X6' 'X3' 'X15' 'X21' 'X7' 'X1' 'X16' 'X20' 'X19' 'X24' 'X17' 'X9'
 'X4' 'X18' 'X2' 'X14' 'X11' 'X8' 'X12']
```

```
# Eliminación de columnas sin relevancia para el modelo predictivo

columnas_a_eliminar = [
    'numero_parte',
    'codigo_especialidad',
    'codigo_subespecilidad',
    'subespecilidad',
    'forma_aviso',
    'numero_vehiculos',
    'numero_personal_operativo',
    'canton'
]

for col in columnas_a_eliminar:
    if col in df.columns:
        df.drop(col, axis=1, inplace=True)

# Mostrar primeras 5 registros
```

```
print(df.head())
```

```
   anio  mes  diadelmes                          especialidad estacion
0  2024   12         31                        PREHOSPITALARIA       X5
1  2024   12         31  EVENTOS HIDRICOS Y METEREOLÃ□GICOS       X6
2  2024   12         31                        PREHOSPITALARIA       X3
4  2024   12         31                        PREHOSPITALARIA      X15
5  2024   12         31                        PREHOSPITALARIA      X21
```

```
print("Número de variables y registros antes de la agrupación");
print(df.shape)
# Agrupamos por columas
# Aseguramos que las columnas clave existan
columnas_grupo = ['anio', 'mes', 'diadelmes', 'especialidad', 'estacion']

df_agrupado =
df.groupby(columnas_grupo).size().reset_index(name='emergencias_atendidas'
)
```

```
Número de variables y registros antes de la agrupación
(74032, 5)
```

```
# Agregar columna dia_de_la_semana
# Construimos una columna temporal de fecha a partir de anio, mes,
diadelmes
df_agrupado['fecha'] = pd.to_datetime({
    'year': df_agrupado['anio'],
    'month': df_agrupado['mes'],
    'day': df_agrupado['diadelmes']
}, errors='coerce')

# De la columna fecha extraemos el día de la semana en número (lunes=0 ...
domingo=6)
df_agrupado['dia_de_la_semana'] = df_agrupado['fecha'].dt.dayofweek
# Eliminar variable temporal fecha
df_agrupado.drop('fecha', axis=1, inplace=True)

print (df_agrupado.head())
```

```
   anio  mes  diadelmes                            especialidad estacion  \
0  2022    1          1  ASISTENCIA ESPECIALIZADA SINIESTROS       X5
1  2022    1          1                            INCENDIOS       X1
2  2022    1          1                            INCENDIOS      X14
3  2022    1          1                            INCENDIOS       X2
4  2022    1          1                            INCENDIOS       X3

   emergencias_atendidas  dia_de_la_semana
0                      1                 5
```

```
1                       1                  5
2                       2                  5
3                       1                  5
4                       6                  5
```

```
print('Dataset agrupado')
print(df_agrupado.head())
print('Número de variables y registros dataset agrupado')
print(df_agrupado.shape)
print(df_agrupado.info())
print(df_agrupado.describe(include="all"))
print(df_agrupado.head())
```

```
Dataset agrupado
   anio  mes  diadelmes                          especialidad estacion  \
0  2022    1          1  ASISTENCIA ESPECIALIZADA SINIESTROS       X5
1  2022    1          1                           INCENDIOS       X1
2  2022    1          1                           INCENDIOS      X14
3  2022    1          1                           INCENDIOS       X2
4  2022    1          1                           INCENDIOS       X3

   emergencias_atendidas  dia_de_la_semana
0                      1                 5
1                      1                 5
2                      2                 5
3                      1                 5
4                      6                 5
Número de variables y registros dataset agrupado
(34799, 7)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34799 entries, 0 to 34798
Data columns (total 7 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   anio                   34799 non-null  int64
 1   mes                    34799 non-null  int64
 2   diadelmes              34799 non-null  int64
 3   especialidad           34799 non-null  object
 4   estacion               34799 non-null  object
 5   emergencias_atendidas  34799 non-null  int64
 6   dia_de_la_semana       34799 non-null  int32
dtypes: int32(1), int64(4), object(2)
memory usage: 1.7+ MB
None
                anio           mes     diadelmes     especialidad estacion
\
count   34799.000000  34799.000000  34799.000000            34799    34799
unique           NaN           NaN           NaN                7       20
top              NaN           NaN           NaN  PREHOSPITALARIA       X6
freq             NaN           NaN           NaN            16817     2676
mean     2023.009253      6.604356     15.755022              NaN      NaN
std         0.820330      3.399163      8.805131              NaN      NaN
min      2022.000000      1.000000      1.000000              NaN      NaN
25%      2022.000000      4.000000      8.000000              NaN      NaN
```

```
50%      2023.000000      7.000000     16.000000              NaN      NaN
75%      2024.000000      9.000000     23.000000              NaN      NaN
max      2024.000000     12.000000     31.000000              NaN      NaN

        emergencias_atendidas  dia_de_la_semana
count             34799.000000      34799.000000
unique                     NaN               NaN
top                        NaN               NaN
freq                       NaN               NaN
mean                  2.127417          3.072876
std                   1.528143          2.015923
min                   1.000000          0.000000
25%                   1.000000          1.000000
50%                   1.000000          3.000000
75%                   3.000000          5.000000
max                  13.000000          6.000000
   anio  mes  diadelmes                          especialidad estacion  \
0  2022    1          1  ASISTENCIA ESPECIALIZADA SINIESTROS       X5
1  2022    1          1                            INCENDIOS       X1
2  2022    1          1                            INCENDIOS      X14
3  2022    1          1                            INCENDIOS       X2
4  2022    1          1                            INCENDIOS       X3

   emergencias_atendidas  dia_de_la_semana
0                      1                 5
1                      1                 5
2                      2                 5
3                      1                 5
4                      6                 5
```

```python
# Cargar dataset de feriados

df_feriados = pd.read_excel("feriados2022_2023_2024.xlsx",
sheet_name="Hoja1")

# Renombrar la columna 'día' a 'diadelmes' para que coincida con el
dataset principal
df_feriados.rename(columns={'día': 'diadelmes'}, inplace=True)

# Unir el dataset de emergencias con el de feriados
df_final = df_agrupado.merge(df_feriados, on=['anio', 'mes', 'diadelmes'],
how='left')

# Llenar valores NaN en 'tipo_dia_festivo' con 'Día regular' para
diferenciar los días normales de los feriados
df_final['tipo_dia_festivo'].fillna('Día regular', inplace=True)

# Mostrar el dataset final
print(df_final.head())
```

```
  anio  mes  diadelmes                          especialidad estacion  \
```

```
0 2022   1      1 ASISTENCIA ESPECIALIZADA SINIESTROS     X5
1 2022   1      1              INCENDIOS     X1
2 2022   1      1              INCENDIOS    X14
3 2022   1      1              INCENDIOS     X2
4 2022   1      1              INCENDIOS     X3

  emergencias_atendidas  dia_de_la_semana tipo_dia_festivo
0                     1                 5    Día regular
1                     1                 5    Día regular
2                     2                 5    Día regular
3                     1                 5    Día regular
4                     6                 5    Día regular
```

```
# Transformación de variables Categóricas  - estacion se elimina el
caracter X y se mantiene el número que identifica a la estación

df_final['estacion'] = df_final['estacion'].str.replace('X', '',
regex=False)
df_final['estacion'] = pd.to_numeric(df_final['estacion'],
errors='coerce')
# Convertimos variable 'especialidad' a categórico
```

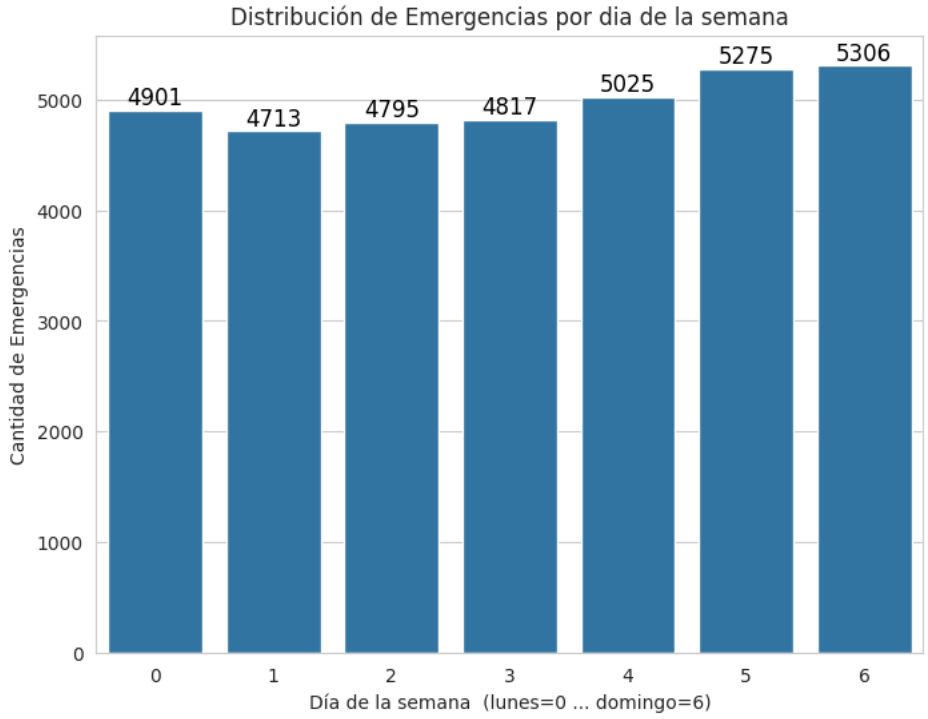
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34832 entries, 0 to 34831
Data columns (total 8 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   anio                   34832 non-null  int64
 1   mes                    34832 non-null  int64
 2   diadelmes              34832 non-null  int64
 3   especialidad           34832 non-null  object
 4   estacion               34832 non-null  int64
 5   emergencias_atendidas  34832 non-null  int64
 6   dia_de_la_semana       34832 non-null  int32
 7   tipo_dia_festivo       34832 non-null  object
dtypes: int32(1), int64(5), object(2)
memory usage: 2.0+ MB
None
```

```
# aplicar  One-hotEncoding en la variable especialidad
df_final = pd.get_dummies(df_final, columns=['especialidad'],
prefix='esp')
df_final = pd.get_dummies(df_final, columns=['tipo_dia_festivo'],
prefix='festivo')
print(df_final.head())
```

```
   anio  mes  diadelmes  estacion  emergencias_atendidas  dia_de_la_semana
\
0  2022    1          1         5                      1                 5
1  2022    1          1         1                      1                 5
2  2022    1          1        14                      2                 5
3  2022    1          1         2                      1                 5
4  2022    1          1         3                      6                 5
```

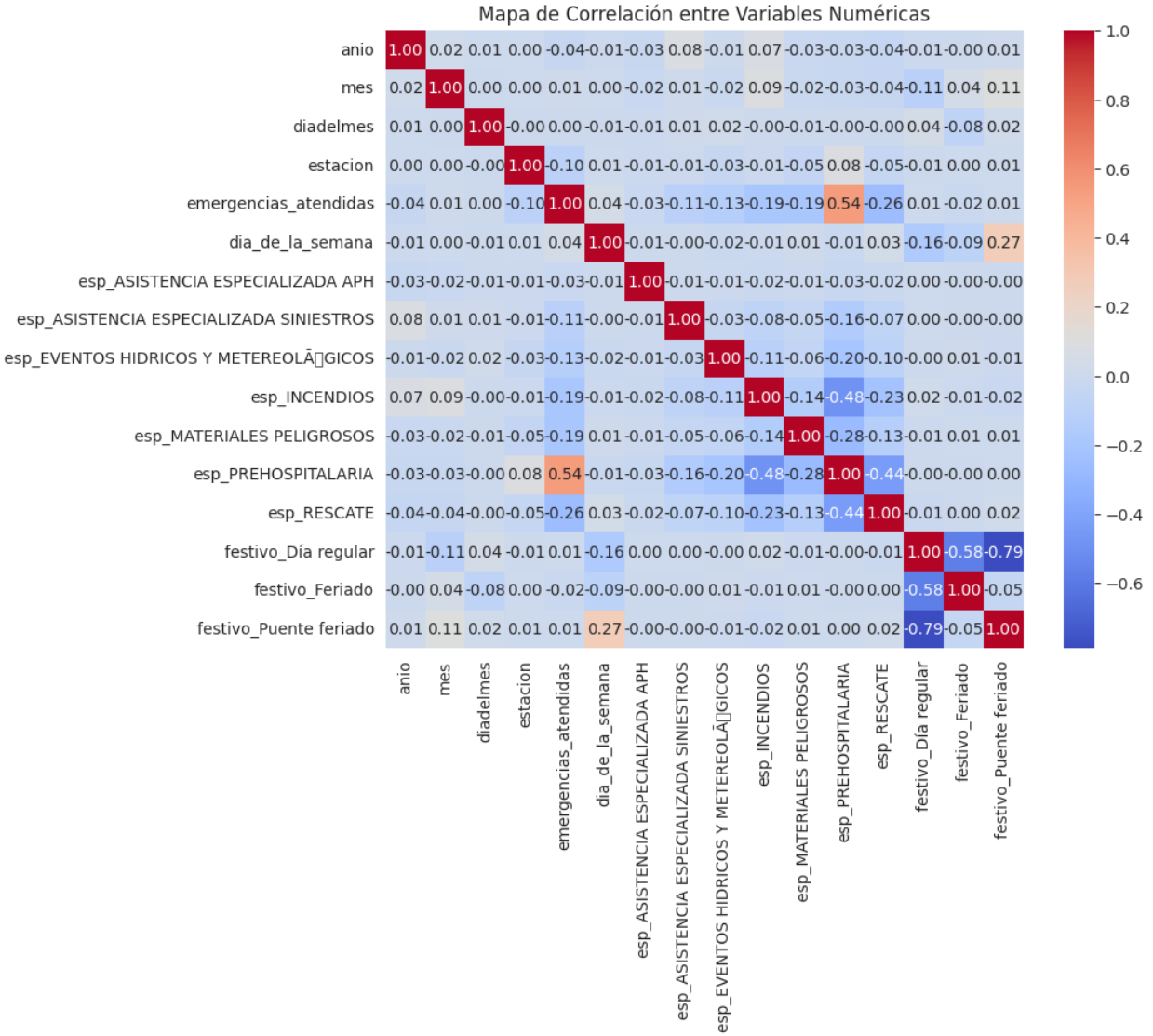
```
   esp_ASISTENCIA ESPECIALIZADA APH  esp_ASISTENCIA ESPECIALIZADA
SINIESTROS  \
0                              False
True
1                              False
False
2                              False
False
3                              False
False
4                              False
False

   esp_EVENTOS HIDRICOS Y METEREOLÃ□GICOS  esp_INCENDIOS  \
0                                    False          False
1                                    False           True
2                                    False           True
3                                    False           True
4                                    False           True

   esp_MATERIALES PELIGROSOS  esp_PREHOSPITALARIA  esp_RESCATE  \
0                      False                False        False
1                      False                False        False
2                      False                False        False
3                      False                False        False
4                      False                False        False

   festivo_Día regular  festivo_Feriado  festivo_Puente feriado
0                 True            False                   False
1                 True            False                   False
2                 True            False                   False
3                 True            False                   False
4                 True            False                   False
```

```
# Visualización de Distribución de Emergencias por dia de la semana
# Emergencias por dia de la semana
plt.figure(figsize=(8, 6))
ax = sns.countplot(data=df_final, x="dia_de_la_semana")
for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}',  # Texto con el valor de la
barra
                (p.get_x() + p.get_width() / 2, p.get_height()),  #
Posición del texto
                ha='center', va='bottom', fontsize=12, color='black',
fontweight='normal')

plt.title("Distribución de Emergencias por dia de la semana")
plt.xlabel("Día de la semana  (lunes=0 ... domingo=6)")
plt.ylabel("Cantidad de Emergencias")
plt.show()
```




```
# Análisis de correlación entre variables numéricas
plt.figure(figsize=(9, 7))
sns.heatmap(df_final.corr(numeric_only=True), annot=True, cmap="coolwarm",
fmt=".2f")
plt.title("Mapa de Correlación entre Variables Numéricas")
plt.show()
```

```
/usr/local/lib/python3.11/dist-packages/seaborn/utils.py:61: UserWarning:
Glyph 147 (\x93) missing from font(s) DejaVu Sans.
  fig.canvas.draw()
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151:
UserWarning: Glyph 147 (\x93) missing from font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```




```python
# Entrenammiento del modelo
# se seleccionas las variables dependiente e independientes
X = df_final.drop('emergencias_atendidas', axis=1)
y = df_final['emergencias_atendidas']

# Dividir en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=42
)
# Generar algoritmo de regresión lineal como línea base
# para comparar resultados con modelo random forest
lr = LinearRegression()

# Entrenar modelo
lr.fit(X_train, y_train)
```

```
# Hacer predicciones
y_pred_lr = lr.predict(X_test)

# Evaluar el modelo
mae_lr = mean_absolute_error(y_test, y_pred_lr)
mse_lr = mean_squared_error(y_test, y_pred_lr)
rmse_lr = np.sqrt(mse_lr)
r2_lr = r2_score(y_test, y_pred_lr)

# Mostrar resultados
print("Resultados de Regresión Lineal:")
print(f"MAE: {mae_lr:.2f}")
print(f"MSE: {mse_lr:.2f}")
print(f"RMSE: {rmse_lr:.2f}")
print(f"R²: {r2_lr:.2f}")
```

```
Resultados de Regresión Lineal:
MAE: 0.89
MSE: 1.56
RMSE: 1.25
R²: 0.33
```

```
# Generar redes neuronales como línea base para comparar
# resultados con modelo random forest
# Escalar datos
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Definir el modelo de red neuronal
nn = MLPRegressor(hidden_layer_sizes=(50, 30), activation='relu',
                  solver='adam', max_iter=500, random_state=42)

# Entrenar modelo
nn.fit(X_train_scaled, y_train)

# Hacer predicciones
y_pred_nn = nn.predict(X_test_scaled)

# Evaluar el modelo
mae_nn = mean_absolute_error(y_test, y_pred_nn)
mse_nn = mean_squared_error(y_test, y_pred_nn)
rmse_nn = np.sqrt(mse_nn)
r2_nn = r2_score(y_test, y_pred_nn)
```

```
# Mostrar resultados
print("Resultados de Redes Neuronales:")
print(f"MAE: {mae_nn:.2f}")
print(f"MSE: {mse_nn:.2f}")
print(f"RMSE: {rmse_nn:.2f}")
print(f"R²: {r2_nn:.2f}")
```

```
Resultados de Redes Neuronales:
MAE: 0.80
MSE: 1.30
RMSE: 1.14
R²: 0.44
```

```
# Definición de modelo Random Forest
rf = RandomForestRegressor(random_state=42)

param_grid = {
    'n_estimators': [50, 100, 150],
    'max_depth': [10, 15, 20],
    'min_samples_split': [2, 4, 6]
}

grid_search = GridSearchCV(estimator=rf,
                           param_grid=param_grid,
                           scoring='neg_mean_squared_error',
                           cv=3,
                           n_jobs=-1)

grid_search.fit(X_train, y_train)

print("Mejores Hiperparámetros:", grid_search.best_params_)

best_params = grid_search.best_params_
rf_final = RandomForestRegressor(
    n_estimators=best_params['n_estimators'],
    max_depth=best_params['max_depth'],
    min_samples_split=best_params['min_samples_split'],
    random_state=42
)

rf_final.fit(X_train, y_train)
```

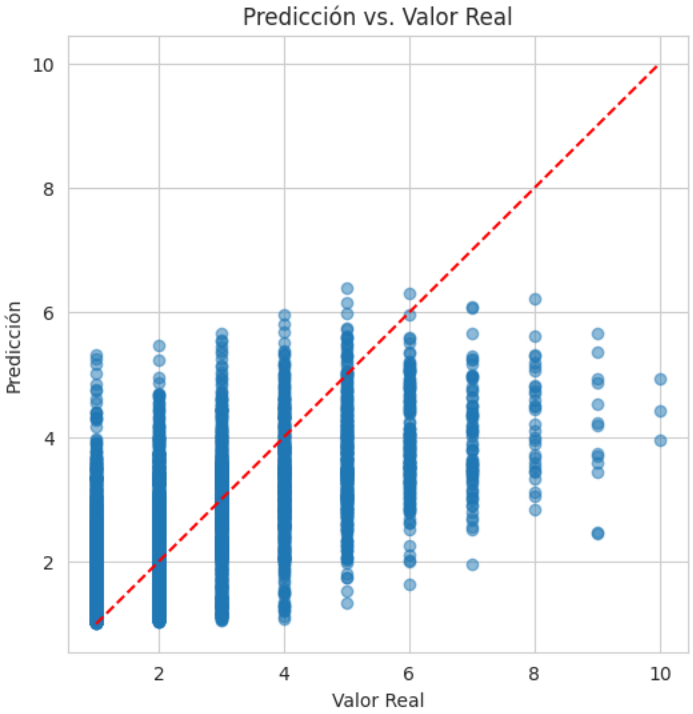
```
Mejores Hiperparámetros: {'max_depth': 10, 'min_samples_split': 6,
'n_estimators': 100}
RandomForestRegressor
```

```
?i
RandomForestRegressor(max_depth=10, min_samples_split=6, random_state=42)
```

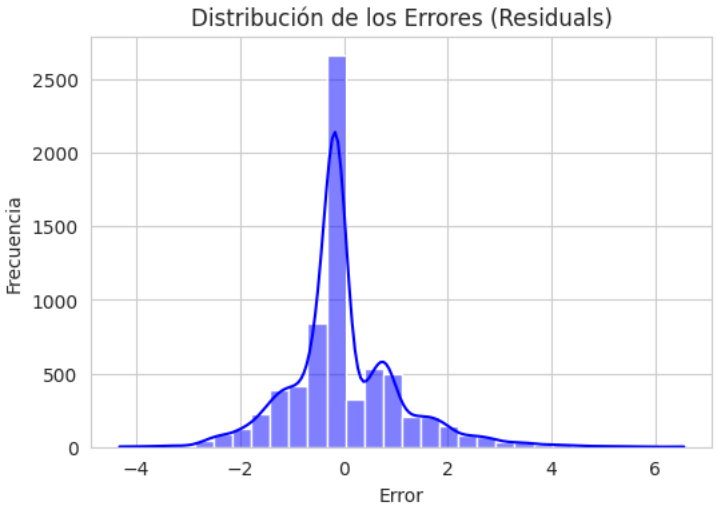
```
# Validación y evaluación del modelo
y_pred = rf_final.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print("Resultados de la Evaluación del Random Forest:")
print(f"MAE:      {mae:.2f}")
print(f"MSE:       {mse:.2f}")
print(f"RMSE:  {rmse:.2f}")
print(f"R²: {r2:.2f}")
```

```
Resultados de la Evaluación del Modelo:
MAE  (Mean Absolute Error):      0.74
MSE  (Mean Squared Error):       1.16
RMSE (Root Mean Squared Error):  1.08
R^2  (Coeficiente de Determinación): 0.50
```

```
# Interpretación de los resultados

# Gráfico de Predicción vs. Valor Real
plt.figure(figsize=(6,6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--
')
plt.xlabel('Valor Real')
plt.ylabel('Predicción')
plt.title('Predicción vs. Valor Real')
plt.show()
```




```
# Distribución de Errores (Residuals)
residuals = y_test - y_pred  # error = real - pred
plt.figure(figsize=(6,4))
sns.histplot(residuals, kde=True, color='blue', bins=30)
plt.title('Distribución de los Errores (Residuals)')
plt.xlabel('Error')
plt.ylabel('Frecuencia')
plt.show()
```




```
# Análisis de Errores vs. Valores Reales
plt.figure(figsize=(6,4))
```

```
plt.scatter(y_test, residuals, alpha=0.5)
plt.hlines(y=0, xmin=y_test.min(), xmax=y_test.max(), colors='r',
linestyles='--')
plt.xlabel('Valor Real')
plt.ylabel('Residual (Real - Predicción)')
plt.title('Residuales vs. Valor Real')
plt.show()
```