



# **Geo-Prevention: Sistema de prevención de incendios forestales que utiliza una red neuronal para predecir la probabilidad de que ocurra un incendio**



**Proyecto para premio Don Bosco  
Edición 35**

**Autor**

Diego Blas Gastón

**Tutor**

Gorka Sanz Lopategui

Área: Innovación tecnológica

Categoría: Tecnologías de la Información y comunicación

Centro docente: Salesianos zaragoza, España



# Índice

<b>1 Introducción</b>	3
<b>2 Marco teórico</b>	4
<b>3 Metodología</b>	6
3.1.1 Recopilación y preprocesamiento de los datos	6
3.1.2 Procesamiento de datos meteorológicos	
3.1.3 INTERPOLACIÓN	8
3.1.4 Interpolación de datos climáticos para cada incendio	9
3.1.5 Datos sin incendio en puntos aleatorios	10
3.1.6 Dataset combinado de incendios y puntos aleatorios sin incendios para entrenamiento de red neuronal	12
3.2 Creación del modelo de predicción	13
3.3 Implementación de la API y el frontend	14
<b>4 Resultados</b>	16
<b>5 Discusión</b>	19
5.1 Posibles aplicaciones	19
5.2 Limitaciones	20
5.3 Oportunidades de mejora	21



# Capítulo 1

## INTRODUCCIÓN

La prevención de incendios es una problemática importante en todo el mundo, especialmente en zonas geográficas con clima seco y alto riesgo de incendios forestales. En España, los incendios forestales son una amenaza recurrente en muchas regiones del país, y pueden tener consecuencias devastadoras para el medio ambiente y la población local.

En este contexto, el objetivo de este proyecto es desarrollar un modelo de predicción de incendios basado en técnicas de aprendizaje automático. Para ello, se recopilaban datos sobre incendios forestales en España durante el año 2021 y se utilizó una red neuronal para predecir la ocurrencia de incendios en el futuro.

La idea principal de este proyecto es proporcionar información útil para la prevención y la gestión de incendios forestales. El modelo de predicción puede ayudar a las autoridades y a la población local a tomar medidas preventivas antes de que se produzcan los incendios, lo que puede minimizar los daños y las consecuencias negativas asociadas con este fenómeno.

En la presente memoria se detalla el proceso de creación del modelo, desde la recopilación de los datos hasta la implementación de la red neuronal y la evaluación de los resultados obtenidos. También se discuten las limitaciones del modelo y las oportunidades de mejora que podrían ser implementadas en futuros proyectos relacionados con la prevención de incendios.

En resumen, este proyecto tiene como objetivo contribuir a la prevención de incendios forestales en España y proporcionar una herramienta útil para la toma de decisiones de las autoridades y la población local en este ámbito. El código con la aplicación se encuentra en el siguiente enlace de GitHub: <https://github.com/DiegoBlasg/Geo-Prevention>



## Capítulo 2

# MARCO TEÓRICO

A continuación, se presentarán las definiciones de algunos términos utilizados en este proyecto con el fin de clarificar su significado y comprender su relevancia en el contexto de la predicción de incendios forestales.

Red neuronal: una red neuronal es un modelo matemático que se inspira en el funcionamiento del cerebro humano para procesar información y realizar tareas complejas de manera eficiente. Consiste en una serie de nodos interconectados que reciben y procesan señales y emiten una respuesta como resultado.

Red neuronal secuencial: es un tipo de red neuronal en el que los nodos están organizados en capas secuenciales, donde cada capa procesa la salida de la capa anterior. Este tipo de red se utiliza comúnmente en tareas de clasificación y predicción.

Capa: una colección de perceptrones que reciben la misma entrada y producen una salida conjunta. Las redes neuronales suelen estar compuestas por múltiples capas, y las capas se organizan en secuencia para formar una red neuronal secuencial.

React: es una biblioteca de JavaScript utilizada para construir interfaces de usuario interactivas. Se utiliza en aplicaciones web para crear componentes reutilizables y optimizar el rendimiento del navegador.

API: una interfaz de programación de aplicaciones (API) es un conjunto de reglas y protocolos que permite a diferentes aplicaciones comunicarse entre sí. Las API se utilizan comúnmente en el desarrollo de aplicaciones web y móviles para permitir que las aplicaciones compartan datos y funcionalidades.



Keras: es una biblioteca de redes neuronales de código abierto escrita en Python. Se trata de una API de redes neuronales de alto nivel, diseñada para permitir a los usuarios construir rápidamente prototipos de modelos de aprendizaje profundo con facilidad.



# Capítulo 3

## Metodología

La metodología utilizada en este proyecto de predicción de incendios forestales en España se divide en tres fases principales: la recopilación y preprocesamiento de los datos, la creación del modelo de predicción y la implementación de la API y el frontend para la visualización de las predicciones.

### 3.1.1 Recopilación y preprocesamiento de los datos:

Para llevar a cabo el preprocesamiento de los datos, se tuvieron en cuenta dos fuentes principales de información:

los datos climáticos de España en 2021 de AEMET

<https://datosclima.es/Aemet2013/DescargaDatos.html>

Los datos de los incendios forestales en España en 2021 de la NASA

<https://firms.modaps.eosdis.nasa.gov/country/>

Los datos climáticos se encontraban en 365 archivos XLS. Cada archivo contenía información sobre las condiciones climáticas de un día específico en cada una de las estaciones meteorológicas de AEMET. Para facilitar su procesamiento, se utilizó un script que convirtió estos archivos XLS en archivos CSV, y que además incluyó las coordenadas de cada estación meteorológica en el archivo correspondiente.

Una vez se contó con la información climática en formato CSV y la ubicación de cada estación meteorológica, se procedió a unir los datos de incendios forestales con los datos climáticos. Para ello, se utilizó un archivo CSV que contenía información sobre la fecha, ubicación y tamaño de cada incendio forestal. A través de la interpolación de los datos climáticos, se asignaron las condiciones climáticas correspondientes a cada ubicación y fecha de los incendios forestales registrados.

Para aumentar el tamaño del conjunto de datos y mejorar la precisión del modelo, se generaron además datos de incendios forestales ficticios en ubicaciones aleatorias. A estas



ubicaciones aleatorias se les asignaron también datos climáticos mediante el mismo proceso de interpolación utilizado con los datos reales.

En resumen, se utilizó un enfoque de preprocesamiento de datos que permitió unir información climática y de incendios forestales, y que generó datos adicionales para mejorar el tamaño y la calidad del conjunto de datos utilizado para la creación del modelo de predicción.

A continuación se procederá a explicar con detalle cómo se realizó cada paso en Python, este código se puede consultar en el siguiente archivo:

[https://github.com/DiegoBlasg/Geo-Prevention/blob/main/DATOS/data\\_procesing.ipynb](https://github.com/DiegoBlasg/Geo-Prevention/blob/main/DATOS/data_procesing.ipynb)

### **3.1.2 Procesamiento de datos meteorológicos**

El código proporcionado es un programa de Python que procesa datos meteorológicos de la Agencia Estatal de Meteorología (AEMET) de España para el año 2021. El programa genera una lista de fechas para todo el año y para cada fecha, carga los datos meteorológicos correspondientes de un archivo Excel, agrega información de ubicación y limpieza de datos, y luego guarda los datos procesados en un archivo CSV en una carpeta llamada "ClimaData".

El programa utiliza las bibliotecas multiprocessing, pandas, numpy y datetime. La biblioteca multiprocessing se utiliza para crear procesos múltiples que pueden ejecutarse simultáneamente, lo que aumenta la eficiencia del procesamiento de datos. La biblioteca pandas se utiliza para cargar y manipular los datos meteorológicos y la biblioteca numpy se utiliza para crear matrices de ceros que se utilizan para agregar información de ubicación a los datos. La biblioteca datetime se utiliza para generar la lista de fechas para todo el año.

El programa comienza definiendo una función llamada getDays() que genera una lista de fechas desde el 1 de enero de 2021 hasta el 31 de diciembre de 2021. La función utiliza la biblioteca datetime para definir el rango de fechas y luego itera sobre ese rango para generar una lista de fechas en formato "YYYY-MM-DD".

A continuación, se define una función llamada process\_date(date) que se utiliza para procesar los datos meteorológicos para una fecha dada. La función toma como entrada una



cadena que representa la fecha en formato "YYYY-MM-DD". La función primero imprime un mensaje que indica la fecha que se está procesando y luego construye el nombre de archivo de los datos meteorológicos correspondientes. Los archivos de datos se encuentran en una carpeta llamada "ClimaAemet2021" y tienen un nombre que incluye el mes y día de la fecha en formato "Aemet2021-MM-DD.xls". El nombre de archivo CSV para los datos procesados también se crea a partir de la fecha y se guarda en la carpeta "ClimaData".

La función carga los datos meteorológicos de la fecha correspondiente y selecciona las filas que contienen los datos reales. Los datos se almacenan en un objeto DataFrame de pandas llamado `dataset_c`. Luego, la función agrega información de ubicación a los datos al crear dos nuevas columnas en el objeto `dataset_c` para la latitud y la longitud. La función carga un archivo que contiene información sobre las estaciones de medición y busca la estación correspondiente en el archivo de datos meteorológicos. Una vez que se encuentra la estación correspondiente, la función calcula las coordenadas de latitud y longitud de la estación utilizando la información del archivo de estaciones y las agrega al objeto `dataset_c`.

La función elimina cualquier fila que contenga datos faltantes y luego renombra las columnas para que tengan nombres más claros. Finalmente, los datos procesados se guardan en un archivo CSV en la carpeta "ClimaData".

Por último, el programa principal utiliza un bloque `if __name__ == '__main__':` para ejecutar el procesamiento de datos para cada fecha en la lista generada por `getDays()`. El programa itera sobre la lista de fechas y para cada fecha, ejecuta la función `process_date(date)` utilizando `multiprocessing` para una mayor optimización de tiempo.

### 3.1.3 INTERPOLACIÓN

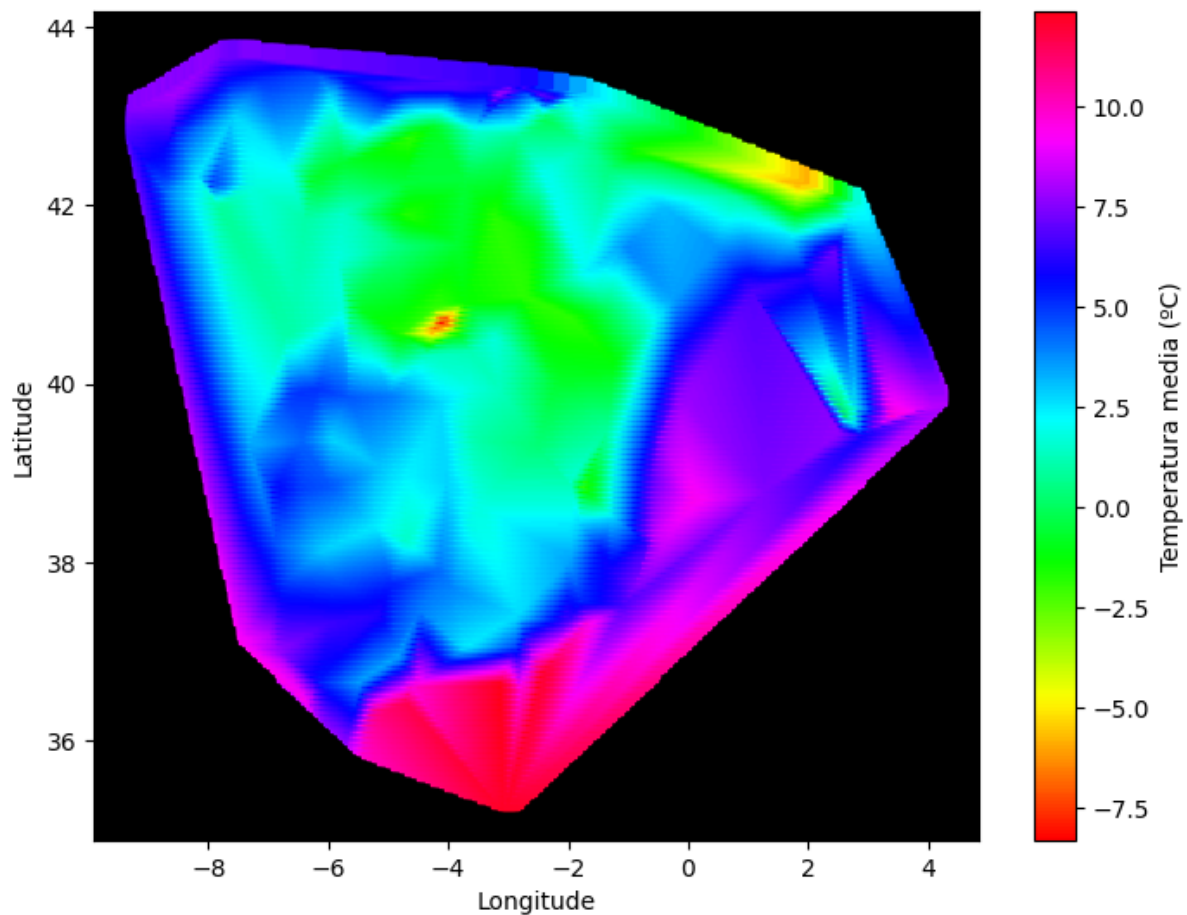
El objetivo de este código es crear un mapa de calor de España de los datos climáticos correspondientes a una fecha determinada. El código lee un archivo CSV que contiene datos de latitud, longitud y la columna seleccionada, filtra los datos para mostrar solo los ubicados dentro de España y define los límites del mapa de calor. Luego, el código crea una cuadrícula de puntos de latitud y longitud, extrae los puntos y valores correspondientes de la columna elegida y los interpola en los puntos de la cuadrícula utilizando el método de interpolación lineal. Finalmente, el código gráfica un mapa de calor utilizando los puntos y valores interpolados y ajusta la posición y los detalles de la visualización. La función





"view\_heatmap" permite cambiar la columna seleccionada y la fecha de los datos a visualizar.

La siguiente imagen es un ejemplo de un gráfico obtenido al ejecutar esta función especificando que queremos los datos de Temperatura media de la fecha 2021-01-02.



### 3.1.4 Interpolación de datos climáticos para cada incendio

Este código carga un conjunto de datos de incendios forestales junto con datos climáticos, y realiza una interpolación de los datos climáticos para cada punto de incendio en el conjunto de datos.

En primer lugar, se cargan las bibliotecas necesarias, incluyendo pandas para cargar el conjunto de datos y numpy y matplotlib para la manipulación de datos y la visualización,



respectivamente. Además, se importa la función `griddata` de `scipy.interpolate` para la interpolación de los datos climáticos.

A continuación, se carga el conjunto de datos de incendios forestales desde un archivo CSV, y se filtran los datos para excluir los puntos con una latitud inferior a 30.

Luego se crea una lista de nombres de columna que contendrán los datos climáticos interpolados, y se agrega cada columna a los datos del incendio con valores iniciales de cero.

Después se define una función llamada `interpolate_climate_data` que toma el índice y la columna de la lista de columnas de clima y realiza la interpolación de los datos climáticos correspondientes a ese punto de incendio en particular.

En primer lugar, se carga el archivo CSV que contiene los datos climáticos correspondientes a la fecha del punto de datos del incendio actual. A continuación, se filtran los datos climáticos por límites de latitud y longitud.

Luego, se convierten los valores de datos climáticos a flotantes y se extraen las latitudes, longitudes y valores de datos climáticos correspondientes. A continuación, se utiliza la función `griddata` para realizar una interpolación lineal de los datos climáticos para el punto de incendio actual.

Finalmente, se guarda el valor interpolado en la columna correspondiente del conjunto de datos del incendio. La función `interpolate_climate_data` se llama para cada columna de la lista de columnas climáticas para cada punto de datos del incendio.

Por último, se guardan los datos de incendio con datos climáticos interpolados en un archivo CSV

### **3.1.5 Datos sin incendio en puntos aleatorios**

Este código en Python es un ejemplo de cómo generar un conjunto de datos de puntos aleatorios con información meteorológica interpolada para un rango de fechas determinado.

Para ello, se utilizan las librerías `Pandas`, `NumPy`, `Random`, `SciPy` y `Datetime`.



Primero, se define la función `getDays()` que genera una lista de fechas en el rango de tiempo deseado. Luego, se crea un `DataFrame` vacío llamado `datasetRandomPoints` con columnas que contienen información sobre latitud, longitud, fecha de adquisición y varias variables meteorológicas.

A continuación, se utiliza un bucle `for` para generar 10 puntos aleatorios para cada fecha en la lista de fechas generada anteriormente y se añaden al `DataFrame` `datasetRandomPoints`. La información meteorológica de cada punto se completará posteriormente mediante interpolación.

A continuación, se define la función `interpolate_climate_data()`, que recibe un índice que corresponde a un punto en el `DataFrame` `datasetRandomPoints`. La función lee un archivo CSV que contiene información meteorológica para la fecha correspondiente a ese punto y luego filtra los datos para seleccionar solo los que se encuentran dentro de un rango de coordenadas geográficas específico.

Si la columna contiene texto, se extrae el valor numérico. Luego, se divide el `DataFrame` en dos partes: uno con las columnas de latitud y longitud, y otro con la variable meteorológica de interés.

A continuación, se utilizan las librerías `NumPy` y `SciPy` para interpolar los valores de la variable meteorológica en el punto aleatorio del `DataFrame` de `datasetRandomPoints` utilizando `griddata`.

Finalmente, se guarda el `DataFrame` `datasetRandomPoints` como un archivo CSV sin el índice.

En resumen, este código utiliza Python y varias librerías para generar un conjunto de datos de puntos aleatorios con información meteorológica interpolada para un rango de fechas determinadas.



### **3.1.6 Dataset combinado de incendios y puntos aleatorios sin incendios para entrenamiento de red neuronal**

Este código de Python utiliza la biblioteca Pandas para procesar datos y generar un nuevo conjunto de datos a partir de dos conjuntos de datos existentes. El código se divide en varias secciones:

Lectura de los datasets:

El código lee dos archivos CSV y los guarda como dos DataFrames de Pandas. El primer archivo CSV se llama datasetRandomPoints.csv y el segundo se llama incendios.csv.

Selección de columnas relevantes del dataset de incendios:

El código selecciona las columnas 0, 1, 5 y 15-24 del conjunto de datos de incendios y descarta el resto.

Extracción de mes, día y año de las fechas en ambos datasets:

El código extrae el mes, día y año de las fechas en ambos conjuntos de datos utilizando expresiones regulares.

Eliminación de columna acq\_date en ambos datasets:

El código elimina la columna 'acq\_date' de ambos conjuntos de datos.

Agregar una columna fire a cada dataset para diferenciar incendios de no-incendios:

El código agrega una columna 'fire' a cada conjunto de datos para diferenciar entre los incendios y los puntos aleatorios. La columna 'fire' en el conjunto de datos de incendios se establece en 1 y la columna 'fire' en el conjunto de datos de puntos aleatorios se establece en 0.

Concatenación de ambos datasets en uno solo y guardar como archivo csv:

El código combina ambos conjuntos de datos en un solo conjunto de datos y lo guarda como un archivo CSV llamado CompleteDataset.csv sin el índice de Pandas.

En general, este código es un ejemplo de cómo se pueden combinar y procesar conjuntos de datos utilizando la biblioteca Pandas de Python.



### 3.2 Creación del modelo de predicción:

En esta fase, se creó una red neuronal para predecir la ocurrencia de incendios forestales en el futuro.. La red se entrena para predecir la probabilidad de incendio en función de las variables de entrada, y se puede utilizar para hacer predicciones sobre nuevas ubicaciones geográficas.

En este proyecto se ha utilizado la biblioteca Keras de TensorFlow para desarrollar una red neuronal que permita realizar predicciones precisas y confiables.

La red neuronal se compuso de dos capas, la capa de entrada con 16 nodos y la capa de salida con un único nodo. Se optó por utilizar la función de activación sigmoide para ambas capas debido a su capacidad para manejar valores entre 0 y 1. Además, se eligió el algoritmo de optimización Adam y la función de pérdida de entropía cruzada binaria para entrenar la red.

Es importante destacar que se dividió el conjunto de datos en dos conjuntos: uno para el entrenamiento y otro para la validación. Esto permitió evaluar la precisión de la red neuronal utilizando la métrica de precisión (accuracy), lo que dio como resultado una fiabilidad del 97.38%.

La alta fiabilidad obtenida en la predicción de incendios forestales mediante el modelo de red neuronal desarrollado, representa una herramienta valiosa para los expertos en la materia y para los organismos encargados de la gestión de emergencias. Con esta herramienta, se podrán tomar decisiones más acertadas y rápidas, lo que contribuirá a minimizar los daños causados por los incendios forestales.

En caso de querer profundizar en el código utilizado para la creación del modelo, se puede acceder al enlace proporcionado, en el cual se encuentra el archivo correspondiente en el repositorio de GitHub del proyecto.

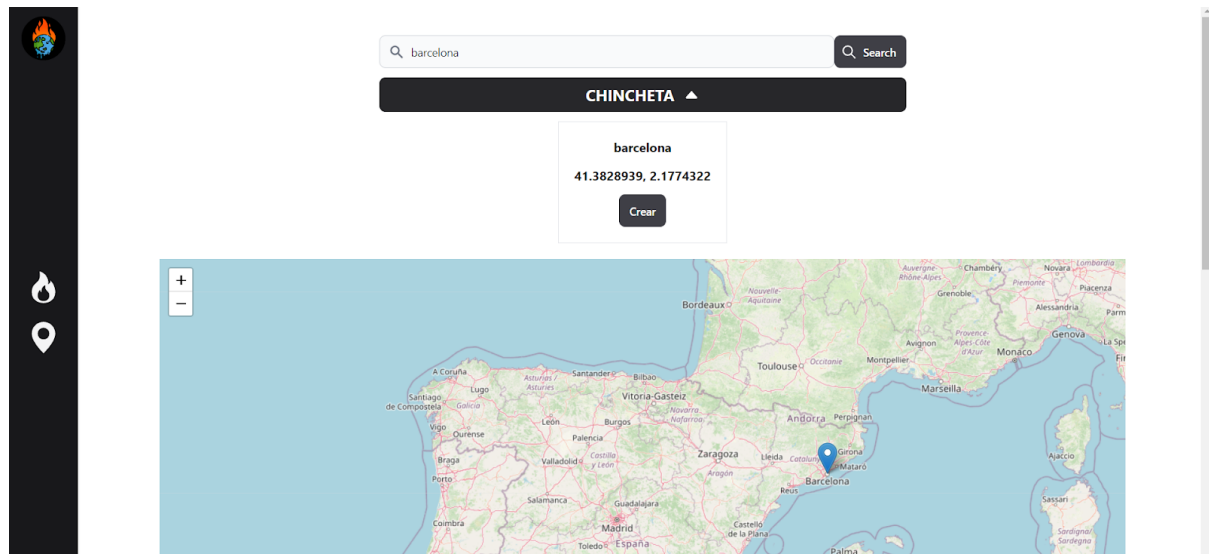
<https://github.com/DiegoBlasg/Geo-Prevention/blob/main/DATOS/RedNeuronal.ipynb>



### 3.3 Implementación de la API y el frontend:

En la fase final, se implementó una API y un frontend para la visualización de las predicciones del modelo. La API se desarrolló utilizando el framework Flask de Python, que permite crear servicios web de forma sencilla. La API permitió a los usuarios enviar una ubicación y una fecha específicas y obtener una predicción de la probabilidad de que se produzca un incendio en esa ubicación en esa fecha.

El frontend se desarrolló utilizando la React, biblioteca de Javascript anteriormente definida, esto permite a los usuarios visualizar las predicciones de forma interactiva en un mapa. Los usuarios pueden introducir una ubicación y ver la probabilidad de incendio en esa ubicación en un mapa



En esta figura se muestra la interfaz de usuario en la sección de ubicación, en esta página el usuario puede introducir el nombre o las coordenadas de la ubicación en la que se desea hacer la predicción. La interfaz de usuario proporciona un mapa que permite al usuario visualizar la ubicación de interés y un botón para guardar la chincheta.

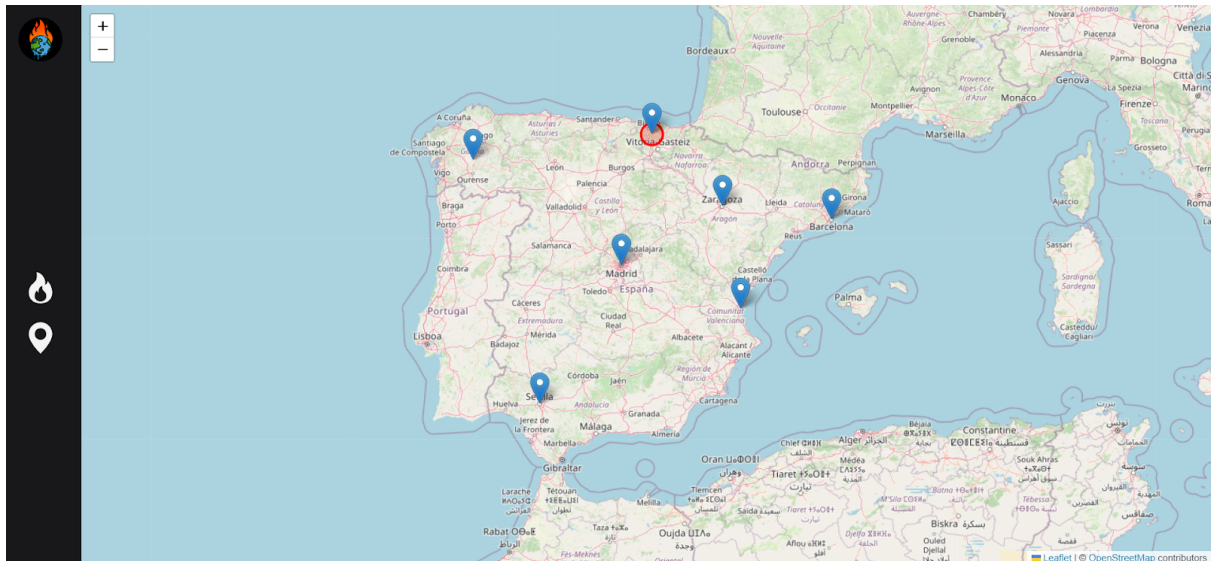
Una vez seleccionada la ubicación, el sistema guardará la chincheta y posteriormente como se puede ver en la siguiente figura se realizarán las predicciones de todas las chinchetas introducidas por el usuario. Para realizar la predicción del riesgo de incendio en la zona seleccionada. La información geográfica incluye las variables necesarias iguales a las



variables con las que se entrenó la red neuronal pero actuales a través de la API ofrecida por openwater maps.

<https://openweathermap.org/>

Si la red neuronal predice que hay más de un 50% de probabilidad de incendio, marcará la chinchera con un círculo rojo.

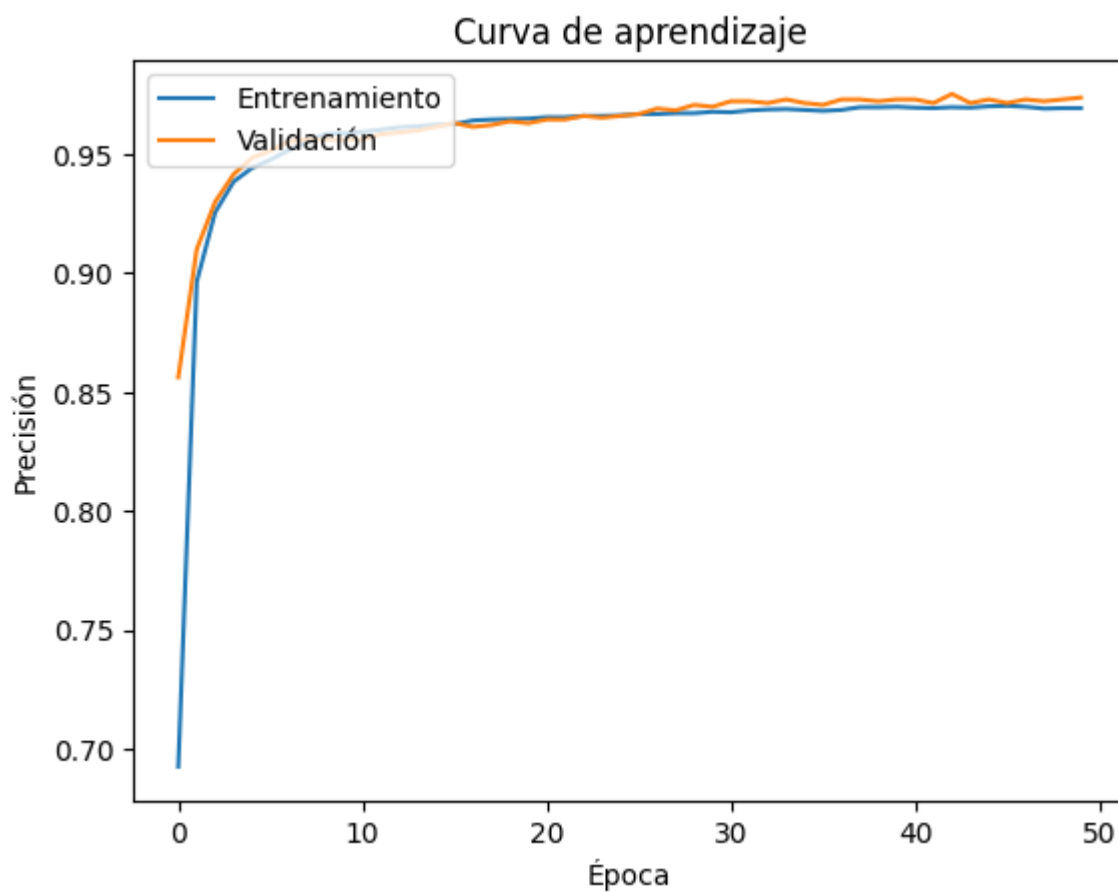




# Capítulo 4

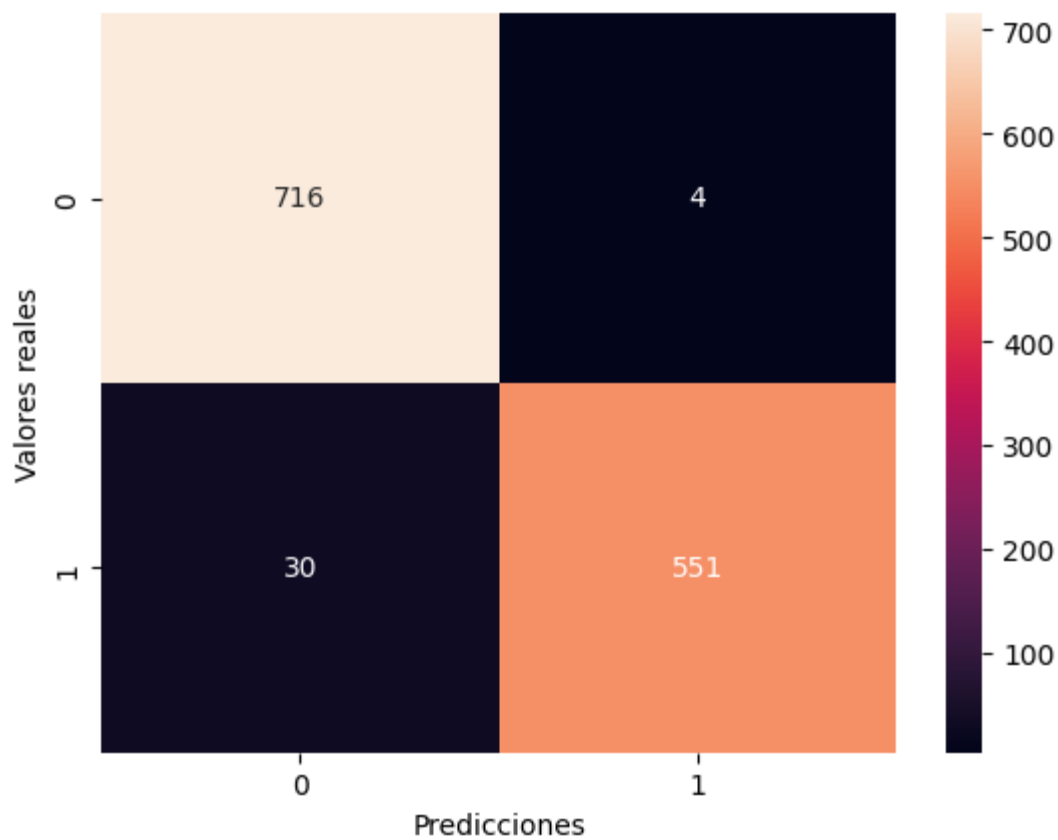
## Resultados

En esta sección se presentan los resultados obtenidos del modelo de predicción. Se muestran las diferentes gráficas que ilustran el funcionamiento de la red neuronal, la evaluación del modelo y el análisis de las variables más importantes.

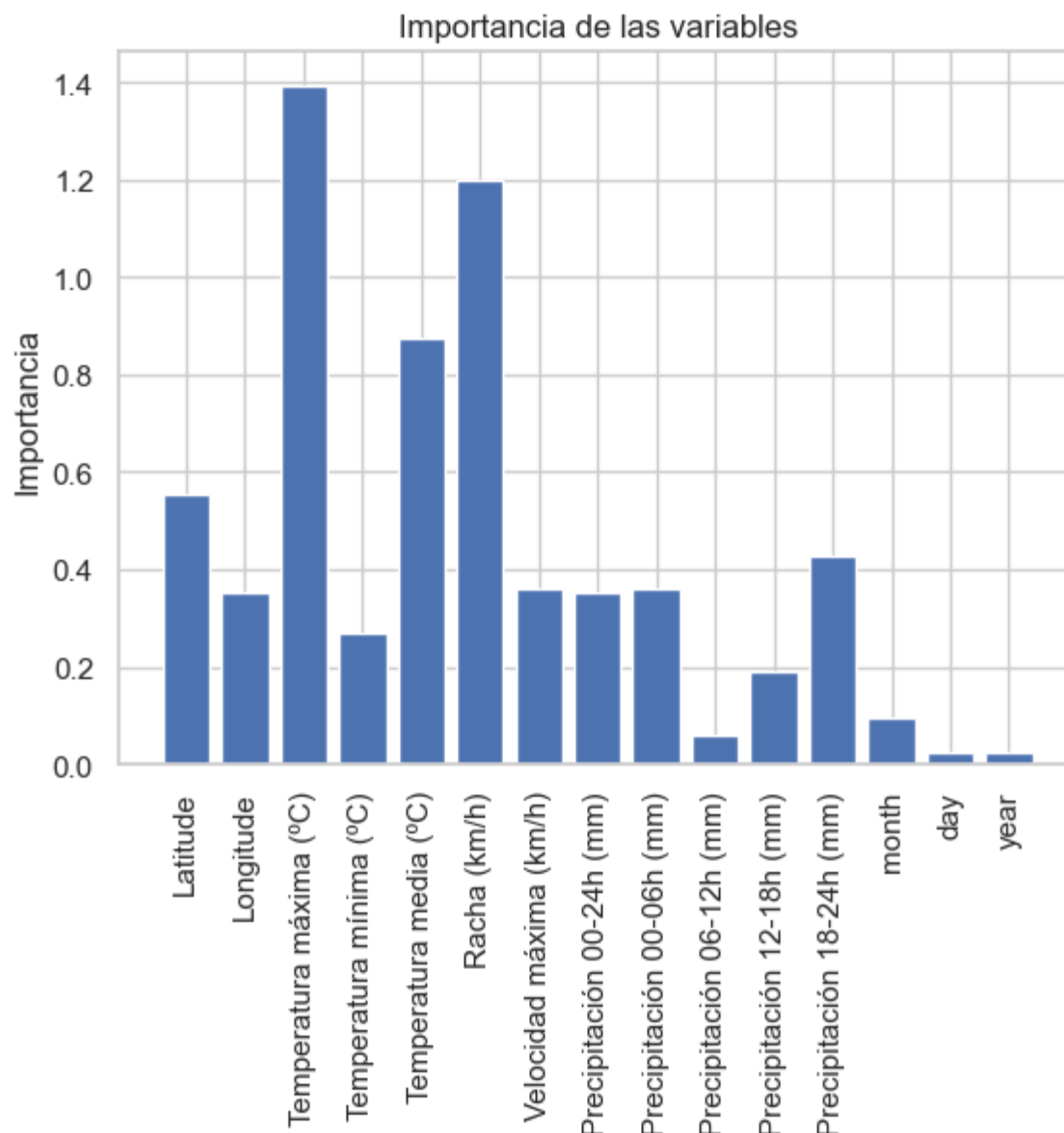


Esta primera gráfica muestra la precisión del modelo durante el entrenamiento y la validación. Como se puede observar, la precisión de la red neuronal aumenta gradualmente durante las épocas de entrenamiento y la curva de validación sigue la misma tendencia, lo que indica que el modelo no está sobre ajustando los datos de entrenamiento.





La segunda gráfica muestra la matriz de confusión del modelo, que evalúa la precisión de las predicciones con los datos de test en términos de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Los resultados muestran que el modelo tiene una alta precisión en la predicción de incendios, con un bajo número de falsos positivos y aún más bajos falsos negativos con estos datos obtenemos que la red neuronal tiene un 97.3% de fiabilidad.



Finalmente se presenta una gráfica que muestra la importancia relativa de cada variable de entrada en la predicción de incendios. La variable más importante es la temperatura media, seguida de la racha y la temperatura media. Las variables de precipitaciones y las fechas tienen una importancia mucho menor en la predicción de incendios.

En general, los resultados obtenidos muestran que la red neuronal tiene una alta precisión en la predicción de incendios forestales y que las variables más importantes para la predicción son la temperatura media, la latitud y la velocidad máxima del viento. Sin embargo, hay oportunidades de mejora, como la inclusión de más variables y el uso de técnicas de aprendizaje automático más avanzadas.



# Capítulo 5

## Discusión

La discusión de los resultados obtenidos en este proyecto es esencial para analizar la eficacia y las limitaciones del modelo. En esta sección se explorarán las posibles aplicaciones del modelo, así como las oportunidades de mejora y las limitaciones identificadas durante la implementación del modelo. La discusión también se centrará en la relevancia y las implicaciones prácticas de los resultados para los equipos de emergencia, las autoridades y el público en general.

### 5.1 posibles aplicaciones

Las posibles aplicaciones del modelo desarrollado son diversas y pueden ser de gran utilidad para la prevención y gestión de incendios forestales en España. Algunas de estas aplicaciones incluyen:

**Predecir futuros incendios en una zona determinada y tomar medidas preventivas:** Con la ayuda del modelo, se pueden identificar zonas que tienen mayor riesgo de sufrir incendios y así tomar medidas preventivas, como la limpieza de la vegetación o la creación de cortafuegos.

**Ayudar a los equipos de emergencia a priorizar sus esfuerzos de extinción de incendios:** Una vez que se han producido los incendios, el modelo puede proporcionar información sobre la probabilidad de que el fuego se propague a otras áreas, lo que puede ayudar a los equipos de emergencia a priorizar sus esfuerzos de extinción y evitar que el fuego se propague.

**Proporcionar información a las autoridades y al público en general sobre las zonas con mayor riesgo de incendios y las épocas del año en que el riesgo es mayor:** La información proporcionada por el modelo puede ser de gran utilidad para las autoridades encargadas de



la prevención y gestión de incendios, así como para el público en general. Con esta información, se pueden tomar medidas preventivas y evitar situaciones de riesgo.

En resumen, el modelo desarrollado puede tener un gran impacto en la prevención y gestión de incendios forestales en España, y sus aplicaciones pueden ser de gran utilidad tanto para los equipos de emergencia como para las autoridades y el público en general.

## **5.2 limitaciones**

La precisión de las predicciones puede verse afectada por la calidad de los datos utilizados: La calidad de los datos es un factor clave en cualquier modelo predictivo. Si los datos utilizados para entrenar el modelo son incompletos, inexactos o sesgados, la precisión de las predicciones puede verse afectada.

El modelo solo considera las variables que se incluyen en los datos de entrenamiento y no tiene en cuenta factores que podrían influir en los incendios, como la velocidad y la dirección del viento: El modelo se basa en los datos de entrenamiento para hacer predicciones. Si no se incluyen todas las variables relevantes en los datos de entrenamiento, es posible que el modelo no pueda tener en cuenta factores importantes que influyen en la ocurrencia de incendios.

El modelo no puede prever los incendios causados por actividades humanas ilegales o intencionales: El modelo se basa en datos históricos y patrones observados para hacer predicciones sobre futuros incendios. Sin embargo, no puede prever los incendios causados por actividades humanas ilegales o intencionales, ya que estos eventos son impredecibles por naturaleza.

En conclusión, aunque el modelo desarrollado puede ser una herramienta útil para la prevención y gestión de incendios forestales, es importante tener en cuenta sus limitaciones y utilizarlo junto con otras fuentes de información y experiencia para tomar decisiones informadas.



### 5.3 Oportunidades de mejora

En el futuro, existen varias oportunidades para mejorar y ampliar la aplicabilidad del modelo de predicción de incendios forestales desarrollado en este proyecto. Algunas de estas oportunidades son:

Incluir más variables en el modelo: Actualmente, el modelo utiliza variables como la temperatura, la humedad y la precipitación para predecir los incendios forestales. Sin embargo, sería interesante incorporar otras variables que puedan influir en la propagación del fuego, como la velocidad y la dirección del viento, la topografía del terreno y la vegetación circundante.

Utilizar técnicas de aprendizaje automático más avanzadas: Aunque el modelo de red neuronal utilizado en este proyecto ha demostrado ser eficaz en la predicción de incendios forestales, existen técnicas de aprendizaje automático más avanzadas, como el aprendizaje profundo y las redes neuronales convolucionales, que podrían mejorar aún más la precisión de las predicciones.

Sería beneficioso integrar el modelo en una plataforma de seguimiento en tiempo real de los incendios forestales, lo que permitiría a las autoridades y equipos de emergencia tomar decisiones informadas en tiempo real.