



Máster de Formación Permanente en Inteligencia  
Artificial

**MODELADO DEL  
COMPORTAMIENTO DE  
INCENDIOS FORESTALES CON  
INTELIGENCIA ARTIFICIAL**

---

**AUTHOR:** Eduardo Palomar Trullen

---

**DIRECTOR DEL MASTER:** Emilio Soria Olivas

---

**TUTOR:** Valero Laparra Perez-Muelas

---

Junio. 2025





## Abstract

Los incendios forestales representan una amenaza creciente para los ecosistemas, la biodiversidad y la seguridad humana. La predicción de su propagación y comportamiento es una tarea compleja, tradicionalmente abordada mediante modelos físicos y algoritmos celulares. En el contexto del Máster de Formación Permanente en Inteligencia Artificial, el presente trabajo explora la viabilidad de utilizar redes neuronales profundas (deep learning) para predecir la extensión e intensidad de incendios forestales. Tras una revisión de las herramientas y enfoques existentes, se proponen varios modelos de aprendizaje profundo que incorporan variables dinámicas como el estado actual del fuego y la velocidad del viento, así como variables estáticas como la vegetación y la altitud del terreno. El objetivo es evaluar la capacidad de las redes neuronales para aprender patrones de propagación del fuego a partir de datos espaciales-temporales y compararlas con los enfoques tradicionales. Los resultados obtenidos permiten discutir las ventajas y limitaciones del uso de inteligencia artificial en este campo, abriendo posibles líneas de investigación y mejora en la predicción temprana de desastres naturales.

Wildfires pose a growing threat to ecosystems, biodiversity, and human safety. Predicting their spread and behavior is a complex task, traditionally addressed using physical models and cellular automata algorithms. In the context of the Master's Degree in Artificial Intelligence, this work explores the feasibility of using deep neural networks (deep learning) to predict the extent and intensity of wildfires.

After reviewing existing tools and approaches, several deep learning models are proposed that incorporate dynamic variables such as the current fire status and wind speed, as well as static variables like vegetation and terrain altitude. The objective is to evaluate the ability of neural networks to learn fire propagation patterns from spatiotemporal data and compare them with traditional approaches. The results obtained allow for a discussion of the advantages and limitations of using artificial intelligence in this field, opening potential lines of research and improvement in the early prediction of natural disasters.

# 1 Contenido

2	Introducción .....	11
2.1	Contextualización del problema de los incendios forestales. ....	11
2.2	Motivación del estudio: La Urgencia de la IA en la Predicción de Incendios Forestales. ....	13
2.3	Objetivos generales y específicos del trabajo.....	13
2.4	Breve descripción de la metodología utilizada.....	14
2.4.1	Búsqueda y Análisis de Variables Relevantes y Datos Disponibles.....	14
2.4.2	Revisión de Modelos y Programas de Predicción Existentes.....	15
2.4.3	Generación de Datos Sintéticos Mediante un Algoritmo Celular Personalizado .....	15
2.4.4	Desarrollo y Optimización del Modelo de Deep Learning .....	15
2.4.5	Creación de una Aplicación de Uso Rápido.....	16
2.5	Estado del Arte .....	16
2.5.1	Modelos Físicos y Empíricos .....	16
2.5.2	Aplicaciones Actuales .....	16
3	Datos Utilizados .....	18
3.1	Variables Involucradas.....	18
3.1.1	Altitud (Estático).....	18
3.1.2	Vegetación (Estático).....	20
3.1.3	Velocidad y Dirección del Viento (Dinámico).....	24
3.1.4	Estado del Fuego (Dinámico).....	29
3.2	Procesamiento y normalización de datos. ....	33
3.2.1	Formato de los datos.....	33
3.2.2	Pre-procesamiento .....	33
3.2.3	Normalización .....	33
3.2.4	División del conjunto de datos .....	34
4	Metodología .....	36
4.1	Modelo I – CNN estática (Conv2D).....	37
4.1.1	Arquitectura del modelo .....	37
4.1.2	Características .....	38

4.1.3	Resultados obtenidos .....	39
4.2	Modelo II – Trained_model_II_Improved_Sequential_Conv2D .....	41
4.2.1	Arquitectura del modelo .....	41
4.2.2	Características principales.....	42
4.2.3	Resultados obtenidos: .....	44
4.3	Modelo III-Trained_model_III_ClassBased .....	45
4.3.1	Arquitectura del modelo .....	45
4.3.2	Características principales.....	47
4.3.3	Resultados obtenidos .....	48
4.4	Modelo IV — Trained_model_IV_AttnFusionDilated.....	50
4.4.1	Arquitectura del modelo .....	50
4.4.2	Características principales.....	52
4.4.3	Resultados obtenidos .....	53
4.5	Modelo V —Trained_model_V_GlobalFlattenDense .....	55
4.5.1	Arquitectura del modelo .....	55
4.5.2	Características principales.....	56
4.5.3	Resultados obtenidos .....	58
4.6	Modelo VI — Trained_model_VI_LocalGlobalFusion.....	60
4.6.1	Arquitectura del modelo .....	60
4.6.2	Características principales.....	61
4.6.3	Resultados obtenidos .....	63
4.7	Modelo VIII — Trained_model_VIII_UNet_Segmentation_Improved .....	65
4.7.1	Arquitectura del modelo .....	65
4.7.2	Características principales.....	69
4.7.3	Resultados obtenidos .....	70
4.8	Modelo IX — Trained_model_IX_CNN_Transformer2D .....	72
4.8.1	Arquitectura del modelo .....	72
4.8.2	Características principales.....	75
4.8.3	Resultados obtenidos .....	77
4.9	Modelo X — Trained_model_X_Attention_UNet.....	79
4.9.1	Arquitectura del modelo .....	79
4.9.2	Características principales.....	83
4.9.3	Resultados obtenidos .....	85



5	Resultados .....	87
5.1	Modelo I – CNN estática (Conv2D).....	<b>¡Error! Marcador no definido.</b>
5.2	Modelo II – Trained_model_II_Improved_Sequential_Conv2D .....	88
5.3	Modelo III-Trained_model_III_ClassBased .....	89
5.4	Modelo IV — Trained_model_IV_AttnFusionDilated.....	90
5.5	Modelo V —Trained_model_V_GlobalFlattenDense .....	91
5.6	Modelo VI — Trained_model_VI_LocalGlobalFusion.....	91
5.7	Modelo VIII — Trained_model_VIII_UNet_Segmentation_Improved .....	93
5.8	Modelo IX — Trained_model_IX_CNN_Transformer2D .....	94
5.9	Modelo X — Trained_model_X_Attention_UNet.....	94
6	Discusión.....	96
6.1	Interpretación de los resultados .....	96
6.2	Limitaciones del modelo y de los datos .....	96
6.3	Posibles mejoras en la arquitectura y los datos .....	97
6.4	Aplicabilidad real en contextos de predicción operativa .....	97
7	Conclusiones .....	98
7.1	Resumen de los logros del trabajo. ....	98
7.2	Reflexión sobre la viabilidad del uso de IA en este campo.....	98
7.3	Futuras líneas de investigación o mejoras.....	99
8	Bibliografía .....	99

## Lista de Ilustraciones

Ilustración 1__ Evolución del número de siniestros y superficies afectadas 1968 – 2015 .....	12
Ilustración 2_Imagen del simulador FlamMap .....	17
Ilustración 3_'Puntacarnero': (36.075818, -5.449429).....	19
Ilustración 4_Mapa Elevación_'Puntacarnero': (36.075818, -5.449429) _n=200 .....	20
Ilustración5_'Puntacarnero': (36.075818, -5.449429).....	21
Ilustración 6_Tipos bosques_zona_'Puntacarnero': (36.075818, -5.449429) _n=200.....	22
Ilustración 7_Tipos bosques_Ajustada_'Puntacarnero': (36.075818, -5.449429) _n=200.....	23
Ilustración 8_Tipos bosque_Matriz numérica_'Puntacarnero': (36.075818, -5.449429) _n=200 .....	24
Ilustración 9_Condiciones Contorno e Iniciales Campo viento Punta Carnero Angulo 79° vel 5m/s _n=200 .....	27
Ilustración 10_Post-Procesamiento Campo viento Punta Carnero Angulo 79° vel 5m/s _n=200.....	28
Ilustración 11_Inicialización Fuego Campo viento Punta Carnero Angulo 79° vel 5m/s _n=200.....	30
Ilustración 12_Paso 100 Fuego Campo viento Punta Carnero Angulo 79° vel 5m/s _n=200.....	31
Ilustración 13_Paso 200 Fuego Campo viento Punta Carnero Angulo 79° vel 5m/s _n=200.....	31
Ilustración 14_Paso 201 Fuego Campo viento Punta Carnero Angulo 347° vel 11.8m/s _n=200.....	32
Ilustración 15_Paso 300 Fuego Campo viento Punta Carnero Angulo 347° vel 11.8m/s _n=200.....	32
Ilustración 16_Evolución entrenamiento Modelo I.....	39
Ilustración 17_Comparación Visual_Model I.....	40
Ilustración 18_Evolución Autoregresiva_Model I.....	40
Ilustración 19_Evolución entrenamiento Modelo II.....	43
Ilustración 20_Comparación Visual_Model II.....	44
Ilustración 21_Evolución Autoregresiva_Model II.....	44
Ilustración 22_Evolución entrenamiento Modelo III .....	47
Ilustración 23_Comparación Visual_Model III .....	48
Ilustración 24_Comparación Visual_Model III .....	48
Ilustración 25_Evolución entrenamiento Modelo IV .....	52
Ilustración 26_Comparación Visual_Model IV .....	54
Ilustración 27_Evolución Autoregresiva_Model IV .....	54
Ilustración 28_Evolución entrenamiento Modelo V .....	57
Ilustración 29_Comparación Visual_Model V .....	58
Ilustración 30_Evolución Autoregresiva_Model V.....	59
Ilustración 31_Evolución entrenamiento Modelo VI .....	62
Ilustración 32_Comparación Visual_Model VI .....	63
Ilustración 33_Evolución Autoregresiva_Model VI .....	64
Ilustración 34_Evolución entrenamiento Modelo VIII.....	69
Ilustración 35_Comparación Visual_Model VIII.....	71
Ilustración 36_Evolución Autoregresiva_Model VIII.....	71

Ilustración 37_Evolución entrenamiento Modelo IX .....	76
Ilustración 38_Comparación Visual_Model IX .....	78
Ilustración 39_Evolució Autoregresiva_Model IX .....	78
Ilustración 40_Evolución entrenamiento Modelo X .....	84
Ilustración 41_Comparación Visual_Model X .....	86
Ilustración 42_Evolució Autoregresiva_Model X.....	86
Ilustración 43_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model I .....	87
Ilustración 44_Paso 25 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model II .....	88
Ilustración 45_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model II .....	88
Ilustración 46_Paso 25 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model III.....	89
Ilustración 47_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model III.....	89
Ilustración 48_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model IV.....	90
Ilustración 49_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model V.....	91
Ilustración 50_Paso 25 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model VI.....	91
Ilustración 51_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model VI.....	92
Ilustración 52_Paso 25 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model VIII.....	93
Ilustración 53_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model VIII.....	93
Ilustración 54_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model IX.....	94
Ilustración 55_Paso 25 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model X.....	94
Ilustración 56_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s_n=50_Model X.....	95

## Lista de Tablas

Tabla 1_Descripción Arquitectura Modelo I .....	37
Tabla2_ Resultados Modelo I .....	40
Tabla3_Descripción ArquitecturaModelo II .....	41
Tabla4_ Resultados Modelo II.....	44
Tabla5_Descripción ArquitecturaModelo III.....	46
Tabla6_ Resultados Modelo III.....	48
Tabla7_Descripción ArquitecturaModelo IV .....	50
Tabla8_ Resultados Modelo IV .....	53
Tabla 9_Descripción Arquitectura Modelo V.....	55
Tabla 10_ Resultados Modelo V .....	58
Tabla 11_Descripción Arquitectura Modelo VI.....	60
Tabla 12_ Resultados Modelo VI.....	63
Tabla 13_Descripción Arquitectura Modelo VIII .....	66
Tabla 14_ Resultados Modelo VIII .....	70
Tabla 15_Descripción Arquitectura Modelo IX.....	73
Tabla 16_ Resultados Modelo IX.....	77
Tabla 17_Descripción Arquitectura Modelo X.....	80
Tabla 18_ Resultados Modelo X.....	85

## 2 Introducción

Los incendios forestales constituyen actualmente uno de los factores de degradación y pérdida más graves del patrimonio forestal y una amenaza para bienes materiales e incluso vidas humanas. A pesar de su presencia histórica en nuestro territorio, en los últimos años la repercusión mediática de los mismos los ha convertido en una realidad mucho más presente en la sociedad, que también está más sensibilizada.

Para evitar los incendios o controlarlos cuando se producen, son muchas las personas que año tras año trabajan con esfuerzo. De toda la cadena humana que hay detrás de la defensa contra incendios, la mayor visibilidad siempre recae en el personal que atiende las emergencias en primera línea o en los medios aéreos, porque se enfrentan directamente a las llamas en maniobras siempre arriesgadas. Pero un fenómeno tan complejo como el fuego forestal requiere de una gestión planificada que vele porque todo funcione coordinadamente, con eficiencia, con seguridad y con ideas innovadoras. Y para ello la recopilación y análisis de la información, planificación de diversas iniciativas de prevención, formación y coordinación o abordar mejoras tecnológicas, entre otros aspectos, pasa a ser fundamental en la lucha contra los incendios forestales.

Los incendios forestales son un fenómeno complejo, cíclico y recurrente; y muchas veces por causas naturales, no obstante, los escenarios confirmados de cambio climático, junto con el proceso continuo de transformación social, económica y ecológica del medio forestal, apuntan a situaciones de emergencia de mayor complejidad. Por eso es preciso un trabajo continuo de esfuerzo y mejora.

### 2.1 Contextualización del problema de los incendios forestales.

El análisis presentado por el Ministerio de Agricultura, Pesca y Alimentación muestra únicamente hasta 2015, esto es debido a que se hace un informe por decenio. El próximo previsiblemente se presentará a lo largo de 2026. Servirá como una buena presentación de lo acontecido en nuestro País:

Analizando los datos contenidos en la EGIF desde 1968 y en la Figura 1 se puede observar la evolución del número de siniestros, las superficies arboladas, desarboladas y superficies totales forestales afectadas por incendios. Los datos contenidos en EGIF están basados en la información contenida en los Partes de Incendio y de Monte. La tendencia del número de siniestros en el periodo comprendido entre 1968 y 2015 es variable. En buena parte de la serie los valores reflejan una evolución creciente, con cierta estabilización a partir de los años noventa y evidentes signos de decrecimiento en la última década. Los valores mínimos de ocurrencia se producen en 1971 (1.665 siniestros), produciéndose un incremento progresivo hasta alcanzarse el máximo en 1996, con 25.557 siniestros, cifra similar a la alcanzada en 2005, con 25.492 siniestros. A partir de este año se aprecia un descenso acusado, con una reducción en los valores máximos, hasta situarse en 11.810 siniestros en el último año de la serie; este valor no se alcanzaba desde inicios de los años 80. En la evolución del número de siniestros el clima es un factor determinante, pero especialmente las circunstancias políticas, sociales, económicas o

culturales que favorecen o reducen la intencionalidad, accidentalidad o negligencia, causas principales en la generación de incendios; entre estas circunstancias se encuentran los condicionantes legales, la gestión del medio forestal, los resultados en las acciones de prevención o la conciliación de intereses, entre otras.

El análisis de las superficies afectadas revela una evolución similar, aunque con un aumento más marcado, mayor variabilidad interanual y un descenso más temprano (Ilustración 1). La tendencia es creciente hasta mediados de los años 80, con una superficie máxima alcanzada en la serie de 484.475,20 ha en 1985, cifra similar a la de 1978, 1989 y 1994, años con más de 400.000 hectáreas quemadas (1994 destacó también por ser el de mayor superficie arbolada afectada, con 250.422,80 ha). A partir de 1994 se produce un notable descenso, hasta situarse en cifras ligeramente superiores a las 100.000 ha en la última década, habiéndose alcanzado en 2014 el segundo valor más reducido de toda la serie con 48.717,83 ha. Este descenso marcado a partir de 1994 se ha venido explicando debido, en gran parte, a la implantación, desarrollo y mejora de la eficacia de los dispositivos de extinción de incendios forestales autonómicos, tras el traspaso de competencias desde el Estado, incluyendo la consecución de resultados de las diversas actuaciones preventivas iniciadas en años precedentes. Sin embargo, la variabilidad climática cíclica estaría detrás de la variación interanual observada en las superficies afectadas por el fuego; de hecho, de acuerdo con datos de la Agencia Estatal de Meteorología (AEMET), en 1991-1995 y 2004-2007 la Península registró sendos periodos de sequía prolongados, mientras que las principales olas de calor desde entonces, por extensión y duración, afectaron en los años 2006, 2012 y 2015<sup>1</sup>.

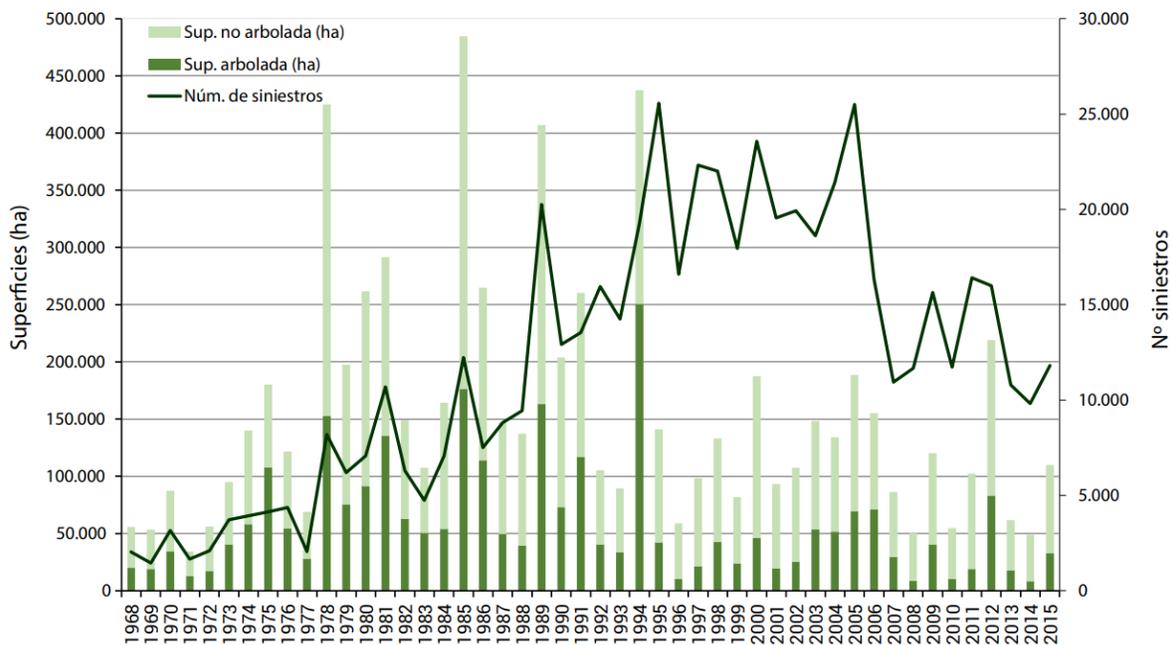


Ilustración 1 \_\_ Evolución del número de siniestros y superficies afectadas 1968 – 2015

<sup>1</sup>Los incendios forestales en España. Decenio 2006-2015.

## 2.2 Motivación del estudio: La Urgencia de la IA en la Predicción de Incendios Forestales.

La capacidad de predecir la propagación y el comportamiento de un incendio se ha convertido en una necesidad crítica para la gestión de riesgos y la mitigación de daños.

Tradicionalmente, la predicción de incendios forestales ha dependido en gran medida de modelos físicos y algoritmos celulares<sup>2</sup>. Si bien estos enfoques han demostrado ser útiles, a menudo se encuentran con limitaciones significativas. Su principal debilidad reside en la complejidad inherente de los sistemas de incendios forestales, que son dinámicos, caóticos y están influenciados por una miríada de variables interconectadas: desde la topografía del terreno y el tipo de vegetación hasta las condiciones meteorológicas en tiempo real (velocidad y dirección del viento, humedad, temperatura). Los modelos tradicionales pueden tener dificultades para capturar estas interacciones no lineales de manera precisa y en tiempo real, lo que a menudo lleva a predicciones con un margen de error considerable. Además, su adaptabilidad a nuevos escenarios o datos imprevistos puede ser limitada, requiriendo reajustes manuales y consumo de tiempo elevado.

## 2.3 Objetivos generales y específicos del trabajo.

Dentro del presente trabajo se van a analizar y comprobar varios puntos relacionados con las redes neuronales y sus capacidades:

1. Capacidad de simular fenómenos físicos con redes neuronales:

Uno de los pilares fundamentales de este proyecto es explorar la **capacidad de las redes neuronales para simular fenómenos físicos complejos**, específicamente la propagación del fuego forestal. Tradicionalmente, la simulación de estos fenómenos se ha logrado mediante modelos basados en principios físicos y ecuaciones diferenciales que describen la transferencia de calor, la combustión de materiales y la interacción con el entorno (viento, topografía). Sin embargo, la inherente complejidad y no linealidad de estos procesos a menudo dificultan la creación de modelos físicos que sean a la vez precisos, eficientes y adaptables a una amplia gama de escenarios.

2. Capacidad de predecir la evolución de problemas multivariable:

El segundo punto crucial de este trabajo es evaluar la capacidad de las redes neuronales para predecir la evolución de los incendios basándose en una multitud de variables interconectadas. La propagación de un incendio forestal es un fenómeno multifactorial, y

---

<sup>2</sup>Juan Li et al 2020 J. Phys.: Conf. Ser. 1575 012058

su complejidad radica en la interacción dinámica y a menudo impredecible de diversos elementos. Para lograr una predicción precisa, es fundamental que el modelo pueda integrar y procesar simultáneamente información de diversas fuentes.

En este proyecto, se considerarán tanto variables estáticas como variables dinámicas:

- Variables Estáticas: Incluyen características del terreno que no cambian rápidamente, como el tipo de vegetación (pastos, matorrales, árboles, etc.), la densidad de combustible, la altitud y la pendiente del terreno. Estas variables definen el "escenario" en el que se desarrolla el incendio.
  - Variables Dinámicas: Son aquellas que cambian en tiempo real o a corto plazo y que tienen un impacto directo y a menudo inmediato en la propagación del fuego. Ejemplos clave son el estado actual del fuego (su tamaño, forma, y frentes activos), la velocidad y dirección del viento, la humedad relativa del aire, la temperatura ambiente y el contenido de humedad de la vegetación (combustible).
3. Comparación entre diferentes modelos de redes neuronales y su adaptación al problema específico

El campo del Deep Learning ofrece una amplia gama de arquitecturas de redes neuronales, cada una con fortalezas y debilidades particulares. Dada la naturaleza espacial-temporal de los datos de incendios forestales, la elección del modelo adecuado es crucial para el éxito de la predicción. En este apartado del trabajo, se realizará una comparación exhaustiva entre diferentes modelos de redes neuronales para determinar cuál se adapta mejor al problema específico de la predicción de la extensión e intensidad de incendios forestales.

## 2.4 Breve descripción de la metodología utilizada.

La metodología de este trabajo se estructuró en varias fases interconectadas, diseñadas para superar las limitaciones de los datos existentes y construir un modelo de predicción de incendios forestales basado en Deep Learning.

### 2.4.1 Búsqueda y Análisis de Variables Relevantes y Datos Disponibles

Inicialmente, se llevó a cabo una exhaustiva búsqueda para identificar las variables clave que influyen en la propagación y el comportamiento de un incendio forestal. Esto incluyó factores como el tipo y la humedad de la vegetación, la topografía (pendiente, altitud), y las condiciones meteorológicas (velocidad y dirección del viento, temperatura, humedad relativa). Paralelamente, se investigó la disponibilidad de estos datos en la red.

La exploración reveló una escasez crítica de conjuntos de datos de incendios forestales que fueran lo suficientemente detallados y precisos para el entrenamiento de una red neuronal profunda. Si bien se encontraron datos globales como la causa del incendio o el área total quemada, la información sobre la evolución espacial y temporal del fuego, incluyendo su frente de avance, intensidad en diferentes puntos, y la correlación con variables dinámicas como el viento en el momento preciso de la propagación, es muy limitada. De manera similar, la disponibilidad de imágenes satelitales o aéreas que capturaran la progresión del fuego en tiempo real y con la resolución necesaria para el entrenamiento de un modelo de Deep Learning era insuficiente o carecía de la información contextual (ej: viento, humedad) indispensable.

#### **2.4.2 Revisión de Modelos y Programas de Predicción Existentes**

Para sentar una base de conocimiento y comprender los enfoques actuales, se realizó una investigación intensiva sobre los modelos y programas existentes para la predicción de la evolución de incendios forestales. Se observó que la mayoría de las herramientas y aplicaciones disponibles se basaban predominantemente en algoritmos celulares o modelos físicos simplificados. Si bien estos modelos son útiles, a menudo presentan limitaciones en su capacidad para capturar la complejidad no lineal de la propagación del fuego y su adaptabilidad a diversas condiciones.

#### **2.4.3 Generación de Datos Sintéticos Mediante un Algoritmo Celular Personalizado**

Ante la limitación crítica de datos reales de alta resolución y con todas las variables necesarias, se ha optado por una estrategia alternativa: la generación de datos sintéticos.

Se buscó la disponibilidad de modelos de predicción, programas y ejecutables en un entorno local que pudieran servir como punto de partida o referencia para el entrenamiento de la red neuronal.

Finalmente se optó por el desarrollo de un programa personalizado basado en algoritmos celulares. Aunque la velocidad de ejecución de este simulador era intrínsecamente lenta, su principal ventaja era la capacidad de proporcionar datos controlados sobre la expansión del fuego bajo diversas condiciones predefinidas de vegetación, topografía y meteorología. Este simulador permitió crear un conjunto de datos rico y multidimensional, incluyendo la evolución paso a paso del área quemada, la intensidad simulada y las variables ambientales asociadas, lo que es crucial para el entrenamiento del modelo de Deep Learning requerido.

#### **2.4.4 Desarrollo y Optimización del Modelo de Deep Learning**

Una vez que se dispuso de un conjunto de datos sintéticos adecuado, se procedió al diseño e implementación del modelo de Deep Learning. Se comenzó con una red neuronal básica, estableciendo una arquitectura inicial y un proceso de entrenamiento fundamental. A partir de esta base, el modelo se fue aumentando en funcionalidades y optimizando progresivamente. Esto implicó la experimentación con diferentes arquitecturas de redes neuronales, la calibración de

hiperparámetros, la implementación de técnicas de regularización y la evaluación de diversas funciones de pérdida y métricas de rendimiento.

### 2.4.5 Creación de una Aplicación de Uso Rápido

Finalmente, para demostrar la viabilidad y la aplicabilidad práctica del modelo desarrollado, se generó una "aplicación" o interfaz de usuario básica. Esta herramienta permite la ejecución rápida y sencilla del programa de predicción, facilitando la interacción con el modelo y la visualización de sus resultados. Aunque rudimentaria, esta aplicación sirve como una prueba de concepto para futuras implementaciones y subraya el potencial de integrar estos modelos de IA en sistemas de alerta temprana y gestión de desastres.

## 2.5 Estado del Arte

A lo largo del tiempo, se han desarrollado diversas metodologías, evolucionando desde enfoques empíricos hasta modelos complejos, y más recientemente, integrando las capacidades de la Inteligencia Artificial.

### 2.5.1 Modelos Físicos y Empíricos

- Simuladores Basados en Rejilla (Grid-based Simulators)
- Algoritmos Celulares (Cellular Automata - CA):

### 2.5.2 Aplicaciones Actuales

A partir de estos métodos tradicionales, se han desarrollado numerosas herramientas y software que son ampliamente utilizados por agencias de gestión de incendios en todo el mundo:

- **FARSITE (Fire Area Simulator)**
- **FlamMap**
- **BehavePlus**

El advenimiento de la Inteligencia Artificial (IA) y, en particular, el Machine Learning (ML) y el Deep Learning (DL), ha abierto nuevas y prometedoras vías para abordar la complejidad de la predicción de incendios forestales. A diferencia de los modelos físicos que requieren un conocimiento explícito de las leyes que rigen el fenómeno, los enfoques de IA tienen la capacidad de aprender patrones y relaciones ocultas directamente a partir de grandes volúmenes de datos, incluso cuando esas relaciones son no lineales o difíciles de modelar explícitamente.

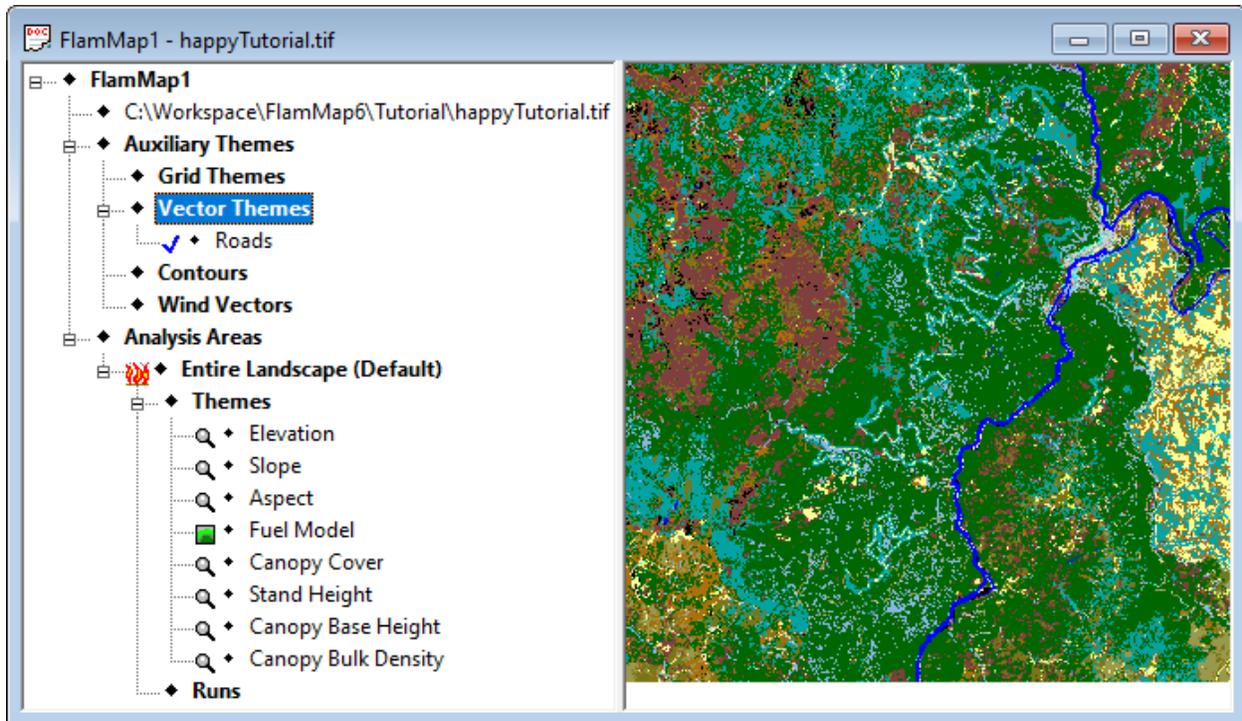


Ilustración 2\_Imagen del simulador FlamMap

## 3 Datos Utilizados

El éxito de cualquier modelo de Deep Learning depende directamente de la calidad y representatividad de los datos de entrenamiento. Dada la escasez de conjuntos de datos reales de incendios forestales con los detalles necesarios para el entrenamiento de redes neuronales comentado anteriormente, parte de la metodología de este estudio se centró en la generación de un conjunto de datos sintéticos que capturara las variables clave que influyen en la propagación del fuego. Este enfoque, como ya se comentó anteriormente, permitió un control preciso sobre las condiciones de simulación y la obtención de datos coherentes y completos.

La resolución espacial del conjunto de datos es de 30 metros por celda, una distancia dada entre la granularidad deseada para capturar detalles del fuego y la disponibilidad de datos base (especialmente topográficos). Esta resolución permite una representación adecuada de la heterogeneidad del terreno y la vegetación.

En cuanto a la resolución temporal, esta está determinada por los pasos de simulación del algoritmo celular. Cada "instantánea" del estado del fuego y las variables dinámicas (viento) se genera a intervalos regulares, permitiendo al modelo de Deep Learning aprender la evolución temporal del incendio.

La combinación de estas variables, tanto estáticas como dinámicas, y la generación de datos sintéticos controlados, proporcionan un entorno de entrenamiento robusto para investigar la capacidad de las redes neuronales para predecir la extensión y la intensidad de los incendios forestales.

### 3.1 Variables Involucradas

Las variables involucradas en el conjunto de datos son:

#### 3.1.1 Altitud (Estático)

La altitud de celda es un factor estático crucial en la propagación de incendios forestales, ya que a partir de ella podemos obtener la pendiente del terreno. La pendiente, a su vez, afecta directamente la velocidad de propagación del fuego: los incendios tienden a acelerar cuesta arriba y a ralentizarse cuesta abajo debido a la precalefacción del combustible y los efectos de la convección.

#### Obtención:

Los datos de altitud para el área de estudio se obtienen mediante la función *obtener\_matriz\_altura*, que interactúa con la plataforma OpenTopography<sup>3</sup> para descargar

---

<sup>3</sup><https://opentopography.org/>. OpenTopography es una organización distribuida operada en colaboración por el San Diego Supercomputer Center de la Universidad de California San Diego, el EarthScope Consortium y la School of Earth and Space Exploration de la Arizona State University. El soporte operativo principal para OpenTopography proviene de la Division of Earth Sciences de la National Science Foundation de EEUU.

Modelos de Elevación Digital (DEM) de alta resolución y luego los procesa para adaptarlos al formato de matriz requerido.

1. **Definición del Área de Interés (Bounding Box):** Al igual que con la variable de vegetación, se utiliza una función auxiliar (`calcular_bounding_box`) para establecer las coordenadas geográficas (latitud/longitud) de la región de estudio, basándose en el tamaño de la cuadrícula (`n`), la resolución deseada (`resolución`) y el punto central (`centro_lat`, `centro_lon`).

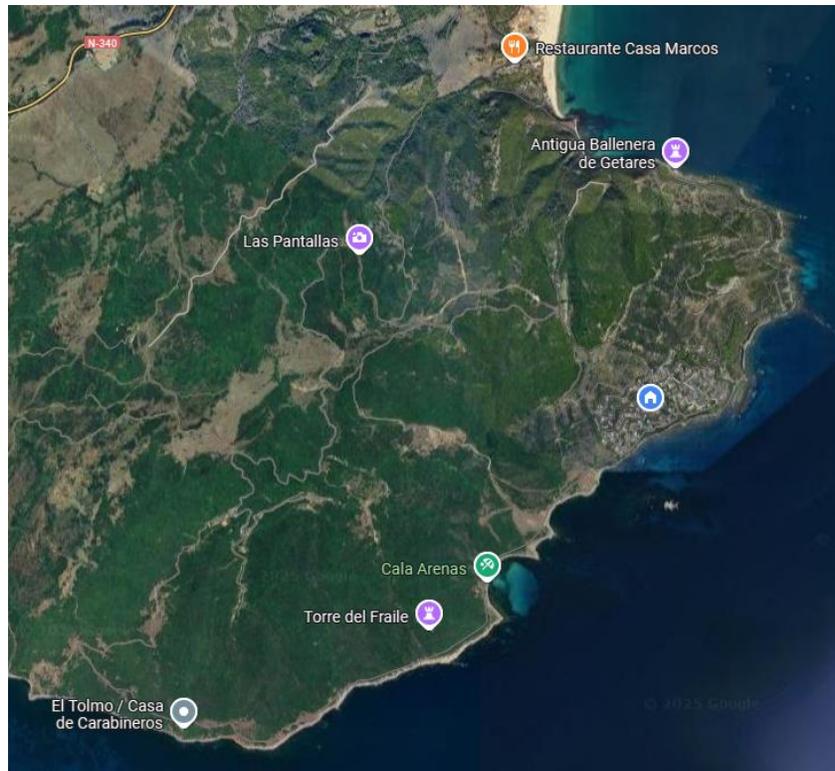


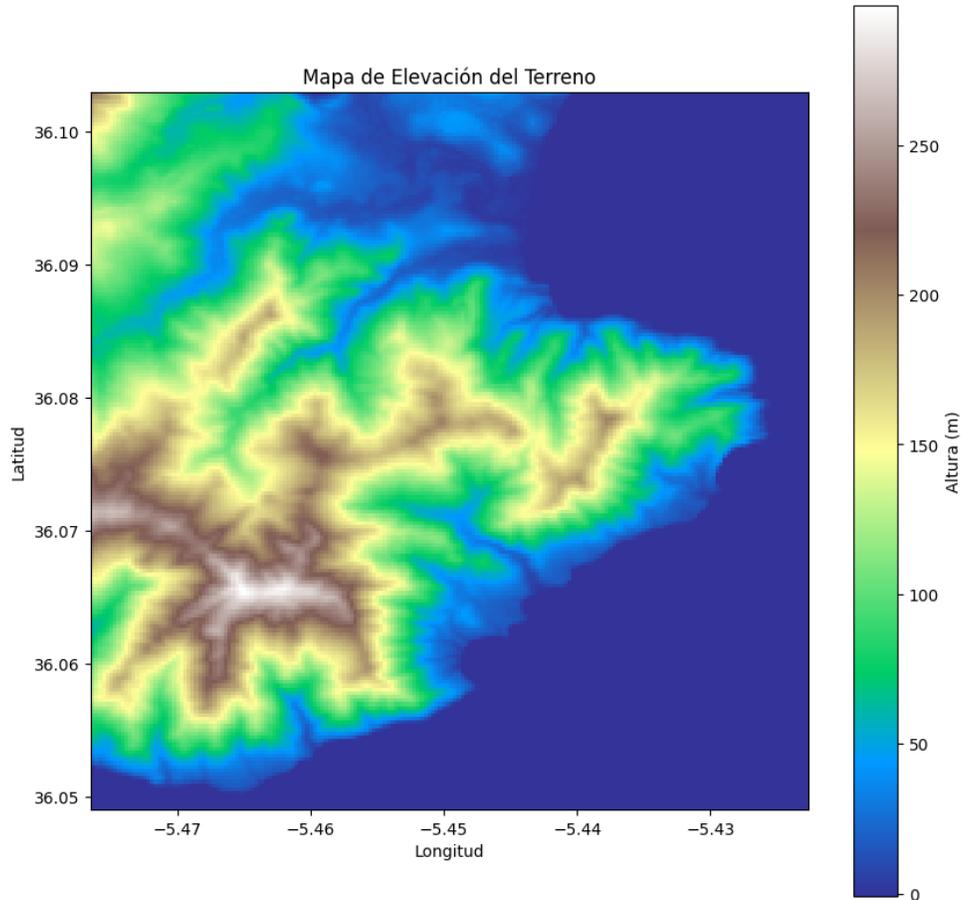
Ilustración 3\_'Puntacarnero': (36.075818, -5.449429)

2. **Solicitud y Descarga del DEM:** La función construye una URL de solicitud para la API de OpenTopography, especificando el tipo de DEM (COP30 para Copernicus 30m DEM), las coordenadas de la `bounding_box`, el formato de salida (GTiff para GeoTIFF) y una API Key personal. Se realiza una solicitud HTTP a esta URL, y si la respuesta es exitosa, el contenido del archivo GeoTIFF se guarda localmente.

OpenTopography fue seleccionada por su versatilidad y la disponibilidad de DEMs globales de 30 metros, resolución espacial conveniente para el estudio.

3. **Lectura y Procesamiento del DEM:** El archivo GeoTIFF descargado se abre y se lee utilizando la biblioteca rasterio. La primera banda del DEM (`dem.read(1)`) se extrae para obtener los valores de elevación (`elevacion`). Durante este paso, se imprime información relevante sobre el DEM, como su resolución y dimensiones.

4. **Redimensionamiento y Normalización:** La matriz de elevación original, elevación, puede tener dimensiones ligeramente diferentes a la matriz  $n \times n$  deseada ( $n_{\text{filas}} \times n_{\text{columnas}}$ ) debido a la forma en que los DEM se ajustan a las coordenadas geográficas. Para asegurar la coherencia con el resto del conjunto de datos, la matriz de elevación se redimensiona (zoom) a las dimensiones  $n \times n$  utilizando una interpolación de orden 1 (bilineal). Esto garantiza que la matriz de altitud tenga la misma resolución espacial de 30 metros por celda que las otras variables.



**Ilustración 4\_Mapas Elevación\_'Puntacarnero': (36.075818, -5.449429) \_n=200**

Este proceso de obtención de datos de altitud de una fuente fiable y su posterior procesamiento para ajustarse a la estructura de la cuadrícula de simulación garantiza que el modelo de Deep Learning reciba información topográfica precisa y estandarizada, esencial para modelar la influencia de la pendiente en la propagación del fuego.

### 3.1.2 Vegetación (Estático)

La vegetación, también nombrada como combustible, es un factor estático fundamental que determina la velocidad de propagación, la intensidad y el comportamiento general de un incendio forestal. Diferentes tipos de vegetación (coníferas, frondosas, matorrales, pastos) tienen distintas

propiedades de combustión, cargas de combustible y patrones de inflamabilidad, lo que afecta directamente cómo el fuego se propaga.

**Obtención:**

La obtención de los datos de vegetación se realiza a través de la función obtener\_biodiversidad, que interactúa con un servicio de mapas online y procesa los datos para adaptarlos al formato matricial requerido por el modelo.

1. **Definición del Área de Interés (*Bounding Box*):** Primero, la función *calcular\_bounding\_box* utiliza las dimensiones de la cuadrícula (*n*), la resolución deseada (*resolucion*), y las coordenadas de un punto central (*centro\_lat*, *centro\_lon*) para definir las coordenadas geográficas (latitud/longitud) de la región rectangular de estudio.

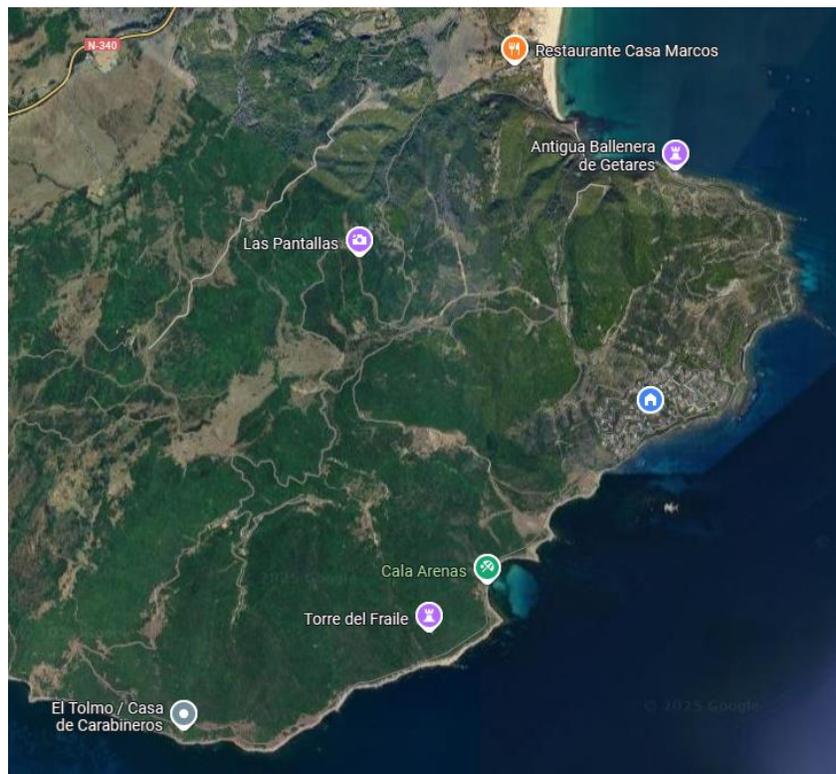


Ilustración5\_'Puntacarnero': (36.075818, -5.449429)

2. **Solicitud de Datos Geoespaciales:** Utilizando la *bounding\_box* calculada, la función realiza una solicitud HTTP a un Servicio de Mapas OGC WMTS específico para datos de biodiversidad: <https://wmts.mapama.gob.es/sig-api/ogc/features/v1/collections/biodiversidad:MFE/items>. Este servicio, gestionado por el Ministerio para la Transición Ecológica y el Reto Demográfico de España (MAPAMA), proporciona información geoespacial detallada, incluyendo tipos de bosque y otros atributos de la vegetación. La solicitud se realiza en formato JSON,
3. **Procesamiento de Datos Vectoriales:** Una vez recibida la respuesta, los datos JSON se cargan en un *GeoDataFrame* de *GeoPandas* (*gdf*) utilizando `gpd.GeoDataFrame.from_features(data["features"])`. Este *GeoDataFrame* contiene geometrías (como polígonos que representan áreas de un tipo de vegetación específico) y

atributos asociados (como tipobosque). En esta etapa, se realizan visualizaciones preliminares para verificar la calidad y el contenido de los datos obtenidos.

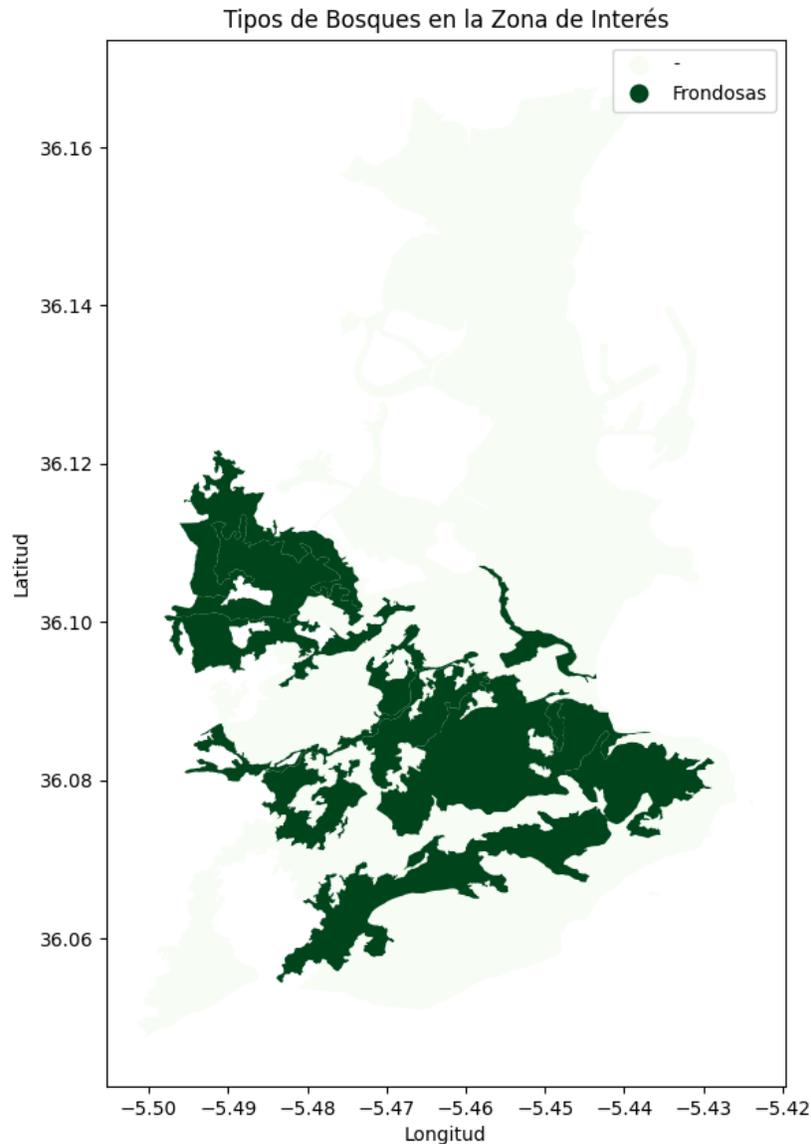


Ilustración 6\_Tipos bosques\_zona\_'Puntacarnero': (36.075818, -5.449429) \_n=200

4. **Rasterización a Matriz (*geo\_to\_matrix\_full\_coverage*):** Esta es la parte más crucial para transformar los datos vectoriales (geometrías) en el formato de matriz rasterizada ( $n \times n$ ) que el modelo de Deep Learning requiere.
  - La función *geo\_to\_matrix\_full\_coverage* crea una cuadrícula interna (*grid\_gdf*) de celdas cuadradas (*box*) que corresponden a la resolución de 30 metros por celda del estudio.

- Luego, utiliza una operación de unión espacial para determinar qué celdas de la cuadrícula se intersectan con las geometrías de vegetación del GeoDataFrame original.
- Para cada celda intersectada, el valor del atributo tipobosque (u otro atributo especificado) se asigna a la celda correspondiente en la matriz resultante. Es importante notar que el índice de fila se invierte ( $n - j - 1, i$ ) para asegurar que la orientación de la matriz sea coherente con la visualización geográfica.

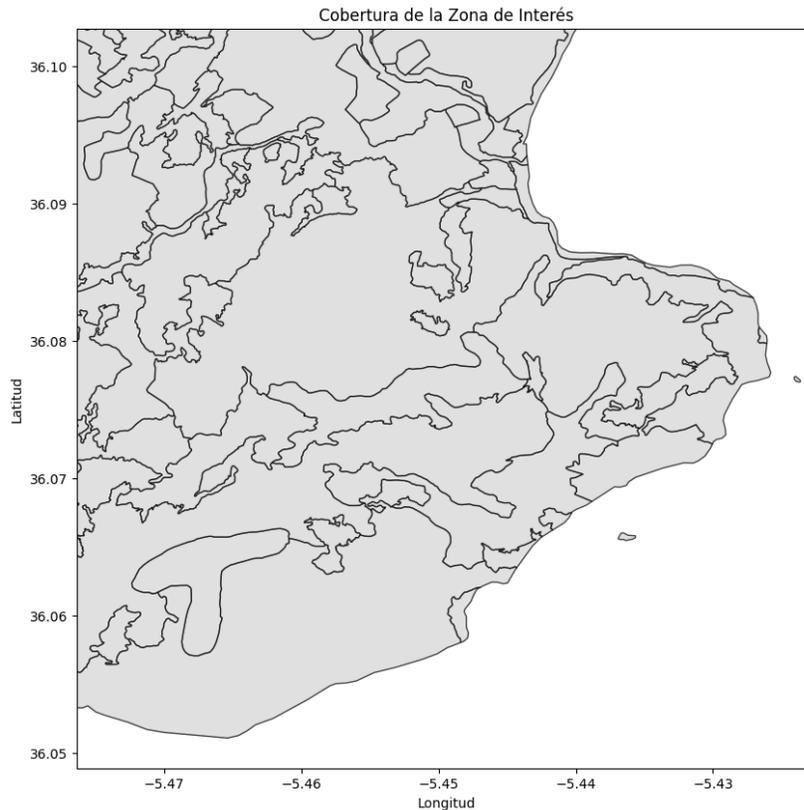
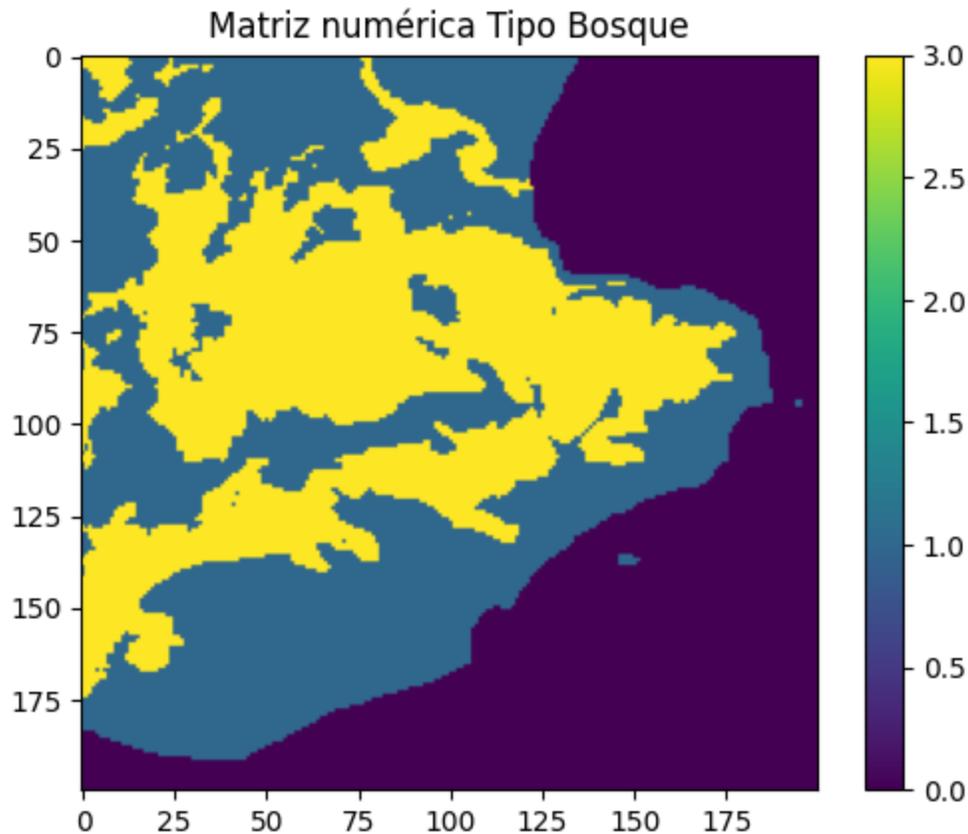


Ilustración 7\_Tipos bosques\_Ajustada\_'Puntacarnero': (36.075818, -5.449429) \_n=200

5. **Transformación a Valores Numéricos** (*transformar\_a\_numerico*): Finalmente, la matriz de vegetación, que inicialmente contiene valores de texto (ej. "Fronosas", "Mixto", "Coníferas"), se convierte a un formato numérico. Esta función asigna un valor numérico único a cada tipo de vegetación. Por ejemplo, "valor\_default" (para áreas sin información o no clasificadas) se mapea a 0, "Fronosas" a 3, "Mixto" a 2, y "Coníferas" a 4. Esta conversión es esencial para que los modelos de Deep Learning puedan procesar esta información como características numéricas.



6.

**Ilustración 8\_Tipos bosque\_Matriz numérica\_'Puntacarnero': (36.075818, -5.449429) \_n=200**

Este proceso garantiza que la variable de vegetación, siendo un factor estático de gran impacto, se integre de manera efectiva en el conjunto de datos rasterizado, permitiendo al modelo de IA aprender cómo los diferentes tipos de combustible influyen en la propagación del fuego.

### 3.1.3 Velocidad y Dirección del Viento (Dinámico)

El viento es un factor determinante en la propagación del fuego. Para este estudio, los datos de velocidad y dirección del viento son sintéticos y se generan como una matriz para cada paso temporal de la simulación.

En un principio se utilizaban valores de  $V$  y  $\theta$  generados de forma arbitraria para el entrenamiento, y se asumía una uniformidad en el viento a lo largo de toda la cuadrícula. Para una simulación más realista del comportamiento de incendios forestales se diseñó la siguiente función que simula la adaptabilidad de los campos de viento al entorno y abre la puerta para futuras exploraciones o funciones más exactas.

#### **Obtención:**

La función *generar\_campo\_viento* simula un campo de viento superficial 2D adaptado a la topografía del terreno, a partir de una dirección y velocidad del viento globales. Este modelo utiliza una aproximación numérica simplificada de las ecuaciones de Navier-Stokes incompresibles.

### Parámetros de Entrada:

1. *elevacion*: matriz 2D ( $n_y \times n_x$ ) que representa la topografía del terreno (en metros).
2. *L*: tamaño de celda (resolución espacial).
3. *wind\_speed*: módulo del viento global (m/s).
4. *wind\_angle\_deg*: dirección global del viento (grados,  $0^\circ$  = norte).
5. *pasos*: número de pasos de simulación para la integración temporal (default = 200).

### Ecuaciones Gobernantes:

El viento superficial se modela a partir del sistema de ecuaciones de conservación de cantidad de movimiento (forma simplificada de Navier-Stokes):

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu_{eff} \nabla^2 \mathbf{u} + \mathbf{F}_{topo} - \mathbf{R}$$

- $\mathbf{u} = (\mathbf{u}, \mathbf{v})$  es el campo de velocidades horizontal (este-oeste y norte-sur),
- $\rho$  es la densidad del aire (constante,  $1.225 \text{ kg/m}^3$ ),
- $\nu_{eff} = \nu + \nu_{turb}$  es la viscosidad cinemática efectiva, incluyendo un componente turbulento constante,
- $\nabla p$  representa el gradiente de presión inducido por el relieve,
- $\mathbf{F}_{topo}$  es la aceleración inducida por la pendiente del terreno (fuerza gravitacional proyectada),
- $\mathbf{R}$  es un término de fricción superficial dependiente de la rugosidad del terreno.

Las ecuaciones se resuelven separadamente para cada componente, vertical y horizontal utilizando una discretización de volúmenes finitos y un esquema de pasos temporales explícitos usando el paquete **FiPy**.<sup>4</sup>

### Términos físicos modelados:

- **Fuerza gravitacional proyectada por la pendiente:**

$$F_g = -g * \alpha * \nabla H(x, y)$$

Donde  $H(x, y)$  es la elevación del terreno y  $\alpha$  es un coeficiente ajustable. Este término acelera el viento cuesta abajo.

---

<sup>4</sup>*FiPy*: Es un solucionador de ecuaciones diferenciales parciales (EDP), orientado a objetos y escrito en Python, basado en un enfoque estándar de volumen finito (FV). Esta combinación proporciona una herramienta extensible, potente y de libre disponibilidad. Una ventaja significativa de Python es el conjunto de herramientas existentes para cálculos con arrays, matrices dispersas y renderizado de datos.

- **Gradiente de presión hidrostático:**

$$\nabla \nabla p \approx -\rho g * \beta * \nabla H(x, y)$$

Simula el efecto de presiones inducidas por diferencias de altura en celdas vecinas.

- **Fricción dependiente de la rugosidad:**

$$R(x, y) = \gamma_0 + \gamma_1 * \|\nabla H(x, y)\|$$

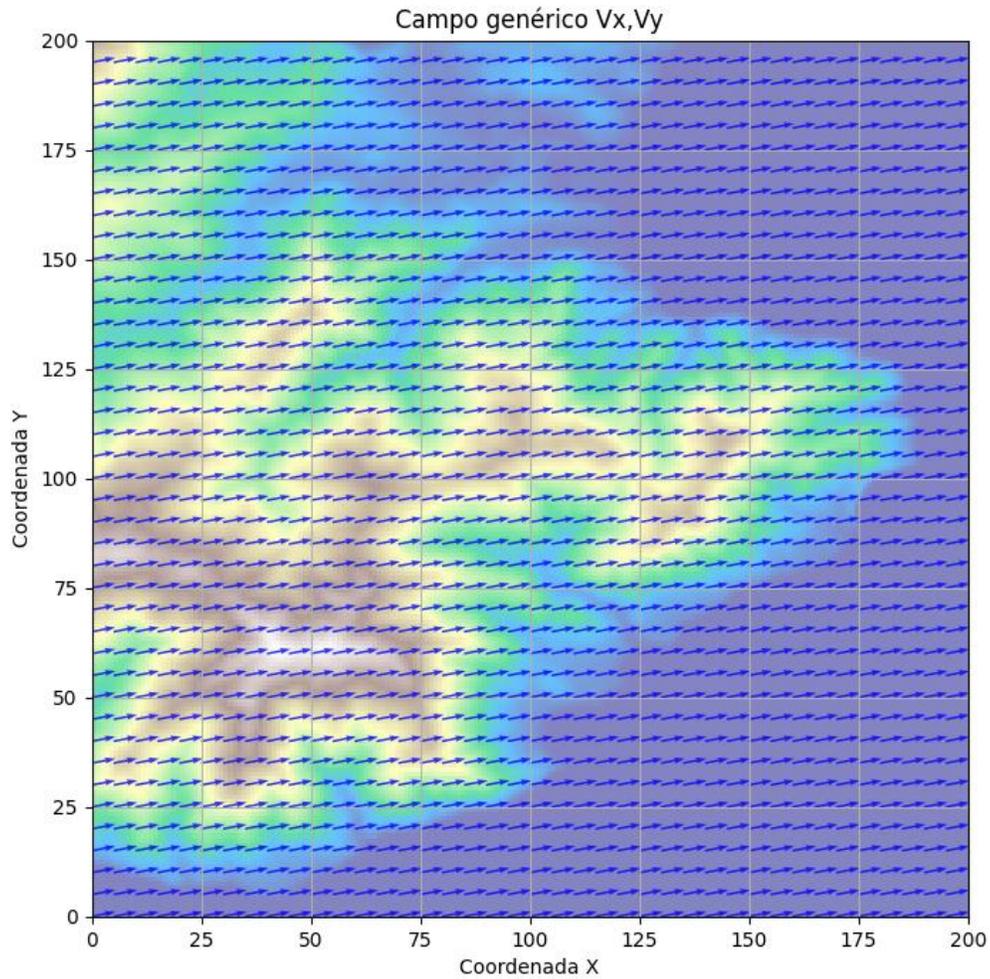
Donde los coeficientes  $\gamma_0 + \gamma_1$  controlan el efecto de la pendiente en la disipación del viento.

- **Difusión turbulenta:**

$$v_{eff} = v + v_t$$

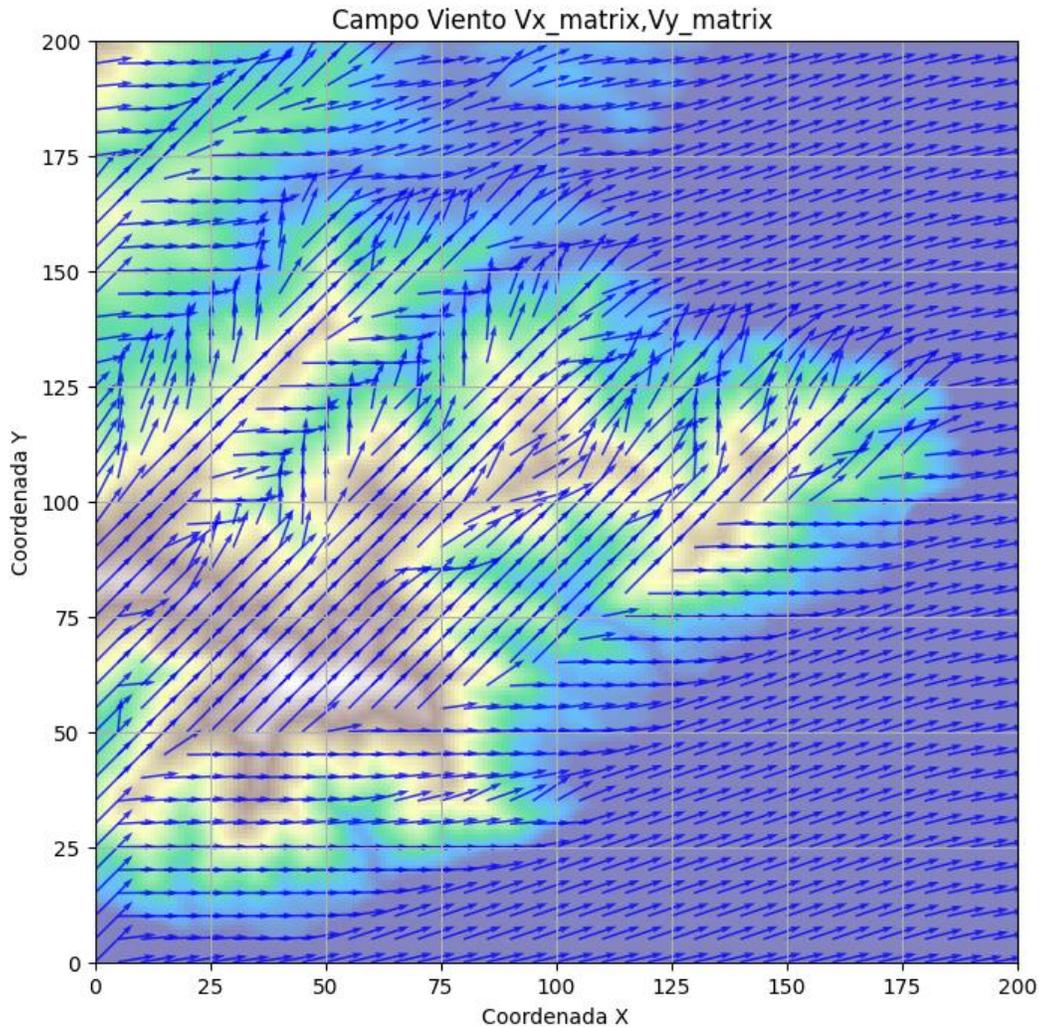
Se incluye mediante un término de difusión isotrópica con una viscosidad efectiva constante:

Se aplican condiciones de contorno tipo Dirichlet en los bordes (entrada constante). Esto se debe a que nos vamos a basar en entradas genéricas del estado del viento en la zona, por ello como aproximación y estado inicial se considera un viento constante a lo largo de toda la malla. Este estado del viento se irá modificando a casa paso, las condiciones de contorno no se modifican.



**Ilustración 9\_ Condiciones Contorno e Iniciales Campo viento Punta Carnero Angulo 79° vel 5m/s\_n=200**

El programa definido se trata de una aproximación intermedia entre modelos puramente empíricos y simulaciones CFD completas con un bajo coste computacional, sin embargo la función es capaz de capturar los efectos de aceleración, freno y modificaciones de la dirección del viento debido a la topografía de un área, (pendientes, valles y cumbres).



**Ilustración 10\_Post-Procesamiento Campo viento Punta Carnero Angulo 79° vel 5m/s\_n=200**

Esta función implica un punto muy importante que se deberá tener en cuenta para el resto del proyecto y por el cual se van a tomar un tipo de modelo de redes neuronales y se estudiarán diferentes metodologías:

**La implementación de esta función implica un efecto espacial global y no local en el comportamiento de un fuego forestal.**

La visión a futuro incluiría la integración de fuentes de datos más sofisticadas, como mapas de viento dados por herramientas de CFD (derivados de modelos microclimáticos) y la incorporación de predicciones de viento de la AEMET (Agencia Estatal de Meteorología). Esto enriquecerá significativamente el realismo de las simulaciones y la capacidad del modelo para predecir el comportamiento del fuego bajo condiciones de viento más complejas y variadas.

### 3.1.4 Estado del Fuego (Dinámico)

Esta es la variable central de nuestro modelo y representa la progresión temporal del incendio. Los datos para el estado del fuego se obtienen directamente de un programa de simulación basado en algoritmos celulares desarrollado específicamente para este proyecto. Este simulador, aunque de ejecución lenta, permite generar una secuencia de "instantáneas" del incendio, mostrando qué celdas están quemadas, ardiendo o en riesgo de ignición en cada paso temporal. Esto proporciona el detalle necesario sobre la extensión y la intensidad simulada del fuego en cada celda a lo largo del tiempo.

#### Obtención:

La obtención del estado del fuego se realiza a través de la función actualizar\_Q, que implementa la lógica central del algoritmo celular. Esta función opera sobre una matriz Q que representa el "estado de inflamabilidad" o "probabilidad de quemado" de cada celda en la cuadrícula en un momento dado.

1. **Inicialización:** La simulación comienza con una matriz Q inicializada por ejemplo, con un punto de ignición central o múltiples focos, en nuestro caso, como la idea es ver la evolución de un foco, se ha inicializado siempre con una celda central "encendida". Existe una función encargada de esta inicialización.
2. **Determinación del Potencial de Propagación (a):** Al inicio de cada paso de tiempo, la función actualizar\_Q calcula una matriz auxiliar a. Esta matriz a determina el "potencial de propagación" de una celda, basándose en su estado actual en Q. Los valores de Q se utilizan para categorizar el estado de la celda (quemada, ardiendo, en riesgo, etc.), y a asigna un valor a este potencial. Por ejemplo,  $Q[i,j] = -1$  significa "no se puede quemar",  $Q[i,j] > 1$  indica una celda activa o con alta probabilidad de propagación y  $Q[i,j]=3$  indica celda quemada.
3. **Cálculo de la Tasa de Propagación (R):** Para cada celda (i, j) que podría propagar el fuego, la función considera sus 8 celdas vecinas (vecindad de Moore<sup>5</sup>). Para cada vecino, se calcula una "tasa de propagación" o "energía transferida" ( $R[di+1, dj+1]$ ). Este cálculo incorpora una serie de factores complejos que simulan el comportamiento físico del fuego:
  - Propiedades del Combustible (RR00): Se utiliza una ecuación que depende de la temperatura (T) y la humedad (H) (por sencillez ambas consideradas constantes en la matriz) de la celda vecina, junto con el potencial calculado previamente. Esta parte del modelo simula cómo las condiciones del combustible influyen en la propagación inicial.
  - Influencia del Viento (KV00): El factor KV00 ajusta la tasa de propagación basándose en la velocidad del viento ( $W\_V$ ) y la dirección del viento (theta) en relación con el ángulo entre la celda central y la celda vecina (ángulo). Esto modela el efecto direccional del viento, que es crucial para la forma del frente de fuego.
  - Influencia de la Pendiente (KH): El factor KH considera la diferencia de altitud ( $S\_L$ ) entre la celda central y la celda vecina, simulando cómo la pendiente acelera o ralentiza la propagación cuesta arriba o cuesta abajo.

---

<sup>5</sup>Autómatas bidimensionales: Hay 4 u 8 vecinos dependiendo que tipo de vecindad se use. La vecindad de Von Neumann solo tiene en cuenta los vecinos situados arriba, abajo, a la izquierda y a la derecha, mientras que la vecindad de Moore tiene en cuenta también los vecinos de las diagonales

- Factor de Vegetación o tipo de combustible ( $K_s$ ): La variable  $K_s$  introduce el efecto de la vegetación en la celda vecina, lo que influye en la inflamabilidad y la velocidad de propagación.
4. **Actualización del Estado de la Celda ( $Q_{nueva}$ ):** La matriz  $Q_{nueva}$  se actualiza para cada celda sumando la contribución de propagación de sus vecinos durante un pequeño intervalo de tiempo ( $dif\_t$ ). La ecuación de actualización da un peso diferente a las celdas adyacentes ortogonalmente y a las diagonales, reflejando cómo el fuego se propaga en una cuadrícula. Si la "probabilidad de quemado" de una celda supera un umbral (por ejemplo,  $Q_{nueva}[i,j] \geq 3$ ), se clasifica como "quemada" o "totalmente consumida" y se establece en un valor máximo (3 en este caso).
  5. **Generación de Secuencias Temporales:** El proceso se repite por un número predefinido de  $num\_steps$ , generando una secuencia de matrices  $Q$  que representan la evolución del incendio a lo largo del tiempo. Estas matrices  $Q$  forman el conjunto de datos dinámico para el entrenamiento de la red neuronal.

Este enfoque de simulación permite crear un conjunto de datos controlable y reproducible que captura la dinámica de la propagación del fuego, incluso en ausencia de datos reales de tan alta resolución, sirviendo como una base fundamental para entrenar el modelo de Deep Learning.

A continuación, se observa en las Ilustraciones 11-15 una simulación de un fuego, con una inicialización en el centro de la imagen y cambio de viento en el paso 200.

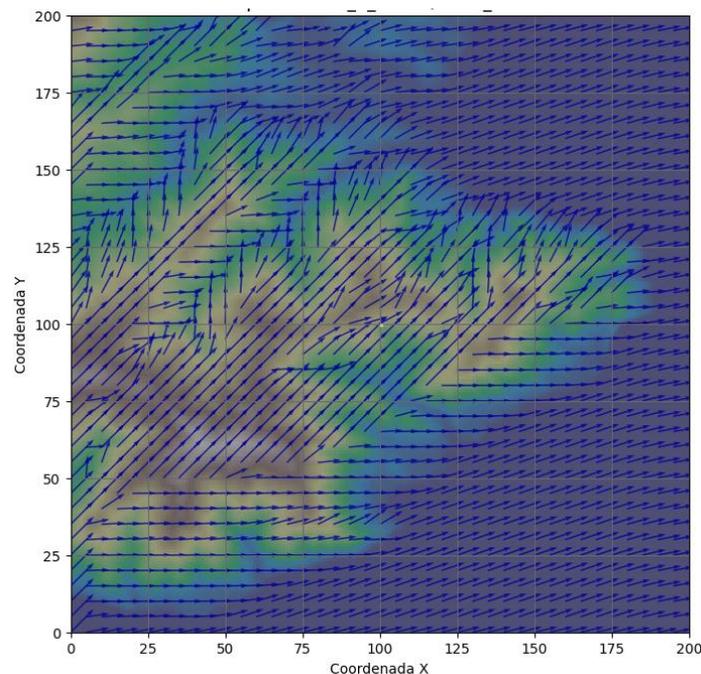
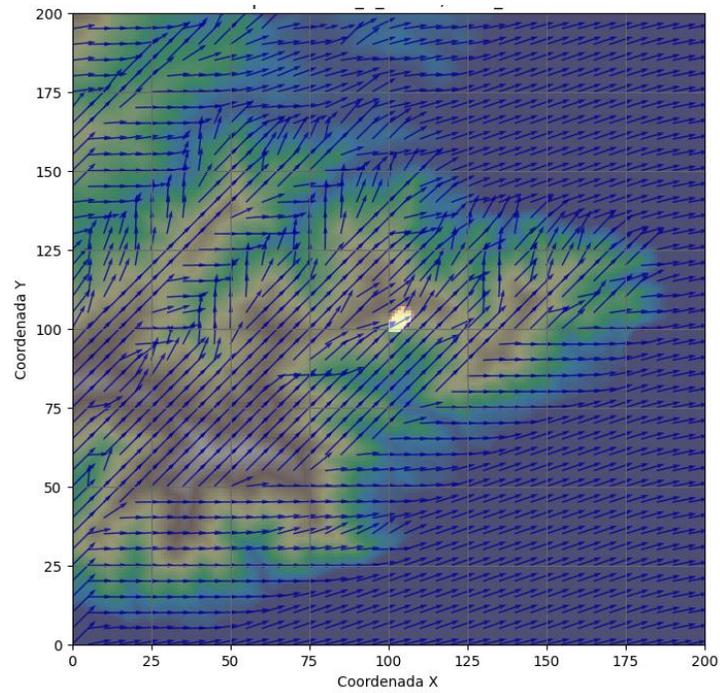
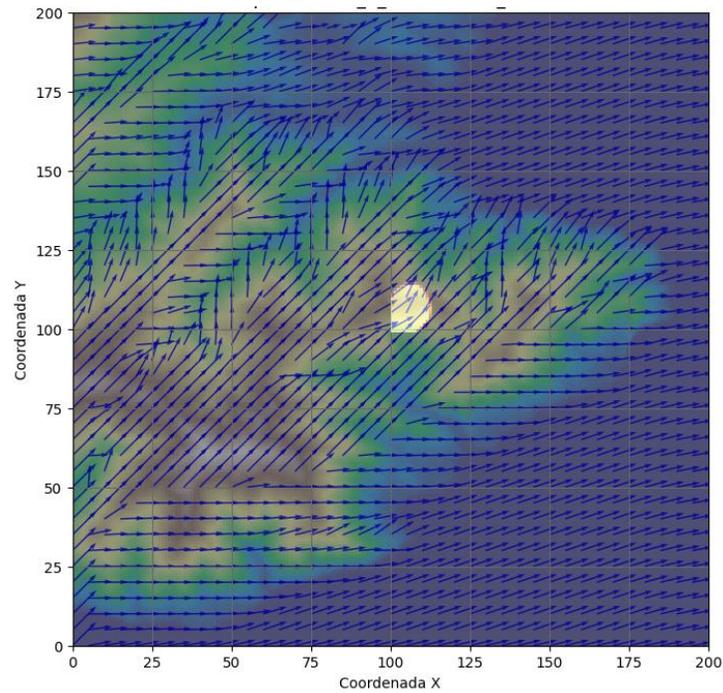


Ilustración 11\_Inicialización Fuego Campo viento Punta Carnero Angulo 79° vel 5m/s\_n=200



**Ilustración 12\_Paso 100 Fuego Campo viento Punta Carnero Angulo 79° vel 5m/s\_n=200**



**Ilustración 13\_Paso 200 Fuego Campo viento Punta Carnero Angulo 79° vel 5m/s\_n=200**

Modificación de la dirección del viento en Paso =200.

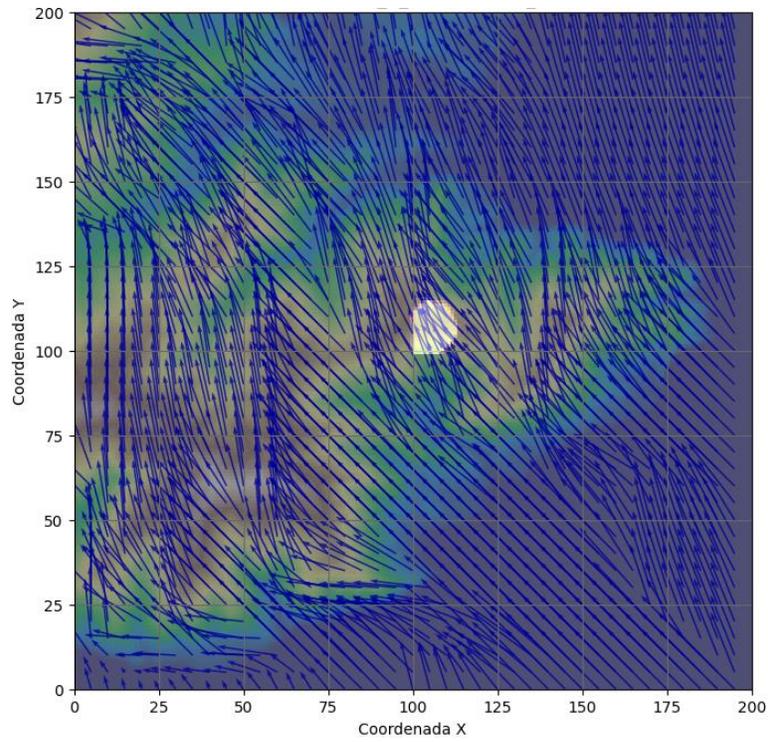


Ilustración 14\_Paso 201 Fuego Campo viento Punta Carnero Angulo 347° vel 11.8m/s\_n=200

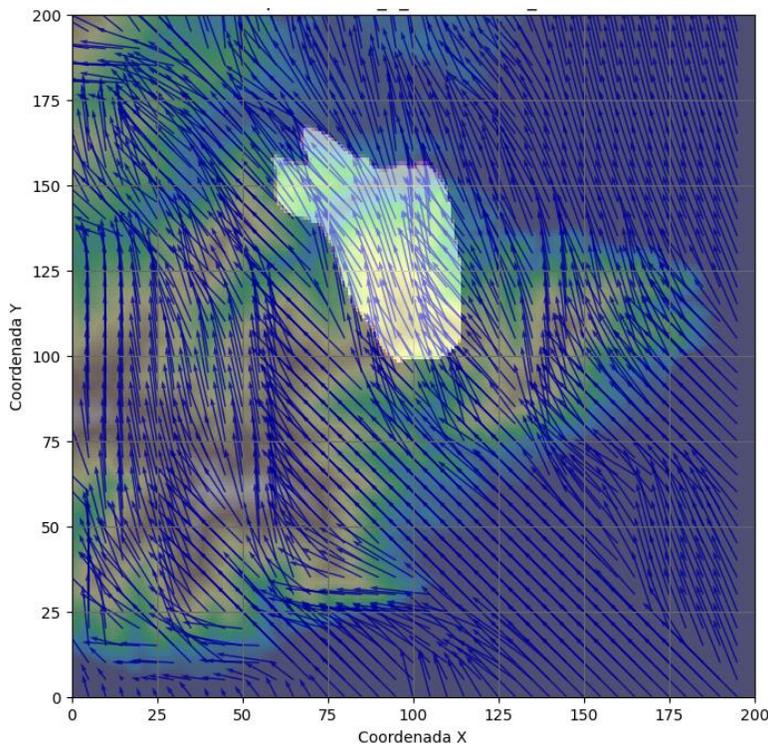


Ilustración 15\_Paso 300 Fuego Campo viento Punta Carnero Angulo 347° vel 11.8m/s\_n=200

## 3.2 Procesamiento y normalización de datos.

### 3.2.1 Formato de los datos

Los datos utilizados en este trabajo están organizados como secuencias temporales de matrices bidimensionales multicanal, representando una ventana espacial de terreno. Cada entrada tiene la forma:

(batch\_size, time steps, height, width, channels)

Puesto que el trabajo finalmente se ha limitado a predicciones a partir de 1 solo step, el conjunto de datos de entrada se simplifica a:

(batch\_size, height, width, channels)

Aunque se tiene la posibilidad de entrenar los diferentes modelos con diferentes capas y resoluciones, en este caso particular:

- height = width = 50 (resolución espacial de 50x50 píxeles)
- channels = 5, correspondientes a:
  - Estado del fuego (dinámico)
  - Velocidad del viento en eje X (dinámico)
  - Velocidad del viento en eje Y (dinámico)
  - Vegetación (estático)
  - Altura del terreno (estático)

Todas las capas tienen valores Float.

### 3.2.2 Pre-procesamiento

El pre-procesamiento consistió en varios pasos:

- Interpolación y ajuste espacial:  
En caso de diferencias en resolución entre capas, se utilizó interpolación bilineal para unificar las dimensiones espaciales nxn.
- Segmentación temporal:  
Se crearon pares de datos (entrada, salida esperada) que representen el estado del terreno en t y la predicción deseada en t+1.
- Máscaras y valores nulos:  
Se aplicaron máscaras para excluir zonas sin datos o no relevantes (por ejemplo, cuerpos de agua o áreas fuera del área de estudio). Los valores nulos fueron reemplazados por ceros o por el valor medio de la variable según el caso.

### 3.2.3 Normalización

La normalización es crítica para mejorar la estabilidad y la eficiencia del entrenamiento del modelo. Se aplicaron dos estrategias distintas según el tipo de canal:

#### **Normalización Min-Max para Características Continuas:**

Para características como la biodiversidad, la elevación y las velocidades del viento ( $V_x$  y  $V_y$ ), hemos utilizado una técnica de normalización llamada Min-Max Scaling.

La fórmula general para esta normalización es:

$$X_{norm} = \frac{(x - x_{min})}{(x_{max} - x_{min})}$$

Donde:

- $x$  es el valor original de la característica.
- $x_{min}$  es el valor mínimo de esa característica observado durante la fase de entrenamiento del modelo.
- $x_{max}$  es el valor máximo de esa característica observado durante la fase de entrenamiento del modelo.

Rangos de entrenamiento:

- **Biodiversidad:** Se normaliza entre un mínimo de 0 y un máximo de 4.
- **Elevación:** Se normaliza entre un mínimo de 0 y un máximo de 2500 metros.
- **Velocidad del viento  $V_y$  (vertical):** Se normaliza entre un mínimo de -25 y un máximo de 25.
- **Velocidad del viento  $V_x$  (horizontal):** Se normaliza entre un mínimo de -25 y un máximo de 25.

### Normalización Específica para el Estado del Incendio (Q)

La matriz de estado del incendio (Q) representa un tipo de dato diferente, que podría indicar la intensidad o fase de un incendio. Para esta característica, aplicamos una lógica de normalización personalizada a través de la función `normalize_fire`:

1. Ajuste de límites (Clipping): Primero, nos aseguramos de que todos los valores en la matriz estén dentro del rango de 0.0 a 3.0. Esto evita que valores atípicos o erróneos (que pudieran ser muy altos o muy bajos) distorsionen el proceso.
2. Manejo de valores específicos: Cualquier valor de -1.0 se interpreta como un estado "sin incendio" o "no aplicable" y se convierte a 0.0.
3. Escalado final: Finalmente, todos los valores se dividen por 3.0. Esto escala el rango de 0.0 a 3.0 a un rango de 0.0 a 1.0, de manera similar a la normalización Min-Max, pero adaptado a la naturaleza de esta característica específica.

### 3.2.4 División del conjunto de datos

El total de datos usado es de 41600 datos o matrices 50x50x5.

Se han utilizado 20 escenarios diferentes para tener mayor variabilidad de datos en la simulación, en cada uno de estos escenarios se han generado varios mapas de viento aleatorio ( $W_V$  y  $\Theta$  aleatorios), y para cada mapa de viento se ha generado una simulación de 77 y 144 steps de evolución de un fuego forestal, en la cual, a mitad del proceso se ha generado un nuevo mapa de viento aleatorio.

Se ha hecho hincapié en la aleatoriedad en el orden de los datos en todos los subconjuntos.

Para garantizar una evaluación robusta del modelo, se dividió el conjunto total de datos en:

- **Entrenamiento:**66%
- **Validación:**16,66%
- **Prueba:**16,66%

Esta división se realizó a su vez de forma temporal y geográfica, asegurando que el modelo no vea interrelaciones durante el entrenamiento de zonas o tiempos específicos que luego se evalúan.

#### Explicación de cada conjunto:

- **Conjunto de Entrenamiento:** Es la parte más grande de los datos y se utiliza para "enseñar" al modelo, ajustando sus parámetros y aprendiendo los patrones subyacentes.
- **Conjunto de Validación:** Se utiliza durante el entrenamiento para ajustar los hiperparámetros del modelo (por ejemplo, la tasa de aprendizaje, el número de capas, etc.) y para monitorear el rendimiento del modelo en datos no vistos durante el entrenamiento directo. Esto ayuda a prevenir el sobreajuste (overfitting) y a seleccionar el mejor modelo. No se utiliza para el entrenamiento de los parámetros del modelo, sino para la evaluación y ajuste iterativo de los hiperparámetros.
- **Conjunto de Prueba:** Es un conjunto de datos completamente independiente que se utiliza una sola vez al final del proceso de desarrollo del modelo para evaluar su rendimiento final y generalización en datos completamente nuevos y no vistos. Es crucial no ajustar el modelo basándose en los resultados del conjunto de prueba para obtener una estimación imparcial de su desempeño en el mundo real.

## 4 Metodología

Con el objetivo de abordar el problema de predicción de la propagación de incendios forestales de forma eficaz y progresiva, adoptamos una metodología basada en la experimentación incremental con modelos de complejidad creciente. Esta estrategia permite analizar, paso a paso, cómo distintos niveles de complejidad influyen en la capacidad predictiva del modelo.

Partimos de arquitecturas sencillas (convolucionales estáticas) y avanzamos hacia modelos más sofisticados, como redes U-Net, y variantes híbridas. Esta evolución controlada permite identificar qué componentes arquitectónicos aportan mejoras significativas, evitar sobreajustes innecesarios y mantener interpretabilidad en el proceso.

Además, este enfoque modular y progresivo favorece una validación más robusta, comparaciones más justas entre variantes y una mejor comprensión del funcionamiento de las diferentes arquitecturas. En última instancia, esta metodología permite construir una base sólida sobre la cual optimizar modelos más complejos, con miras a aplicaciones reales en la gestión y monitoreo de incendios.

Todo el modelo está implementado con clases personalizadas (Layer) que permiten evitar el uso de Lambda, asegurando compatibilidad total con exportación y producción.

Para los entrenamientos se han usado:

- Detención anticipada (early stopping)
- ReducLROnPlateau

Nota: El **Modelo VII** se obvió por no aportar información relevante al igual que otros modelos intermedios y finales sin resultados significantes.

## 4.1 Modelo I - CNN estática (Conv2D)

El objetivo es establecer una línea base con un modelo convolucional sencillo, sin incorporar información temporal. Este modelo busca aprender relaciones espaciales directas entre las variables actuales (estado del fuego, viento, vegetación, altura) y la evolución inmediata del fuego.

Entrada:

- Forma: (alto, ancho, canales)

alto = ancho = n = 50

canales=5

- Canales:
  - Estado del fuego actual
  - Velocidad del viento en eje X
  - Velocidad del viento en eje Y
  - Vegetación (estática)
  - Elevación o altitud (estática)

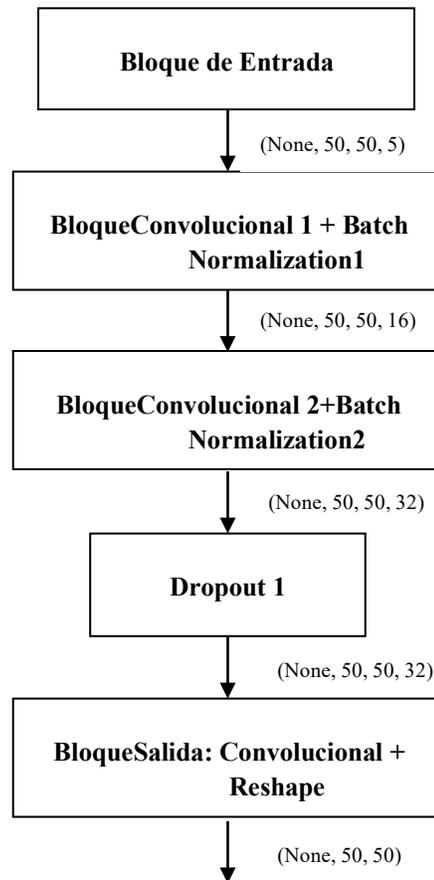
No se utiliza historial temporal; el modelo recibe un único instante como entrada.

### 4.1.1 Arquitectura del modelo

Capa	Descripción
Input	<ul style="list-style-type: none"> <li>• Forma: (H, W, C). Input layer: Representa la cuadrícula de entrada al modelo.</li> </ul>
Conv2D (3×3, 16)	Extrae características espaciales simples con activación ReLU y padding='same'.
BatchNormalization	Normaliza la salida del primer bloque convolucional
Conv2D (3×3, 32)	Extrae patrones más complejos, con activación ReLU y padding='same'.
BatchNormalization	Normaliza la salida del segundo bloque
Dropout (0.2)	Apaga aleatoriamente el 20% de neuronas para evitar sobreajuste
Conv2D (1×1, 1)	Capa de salida: predicción por celda (activación lineal)
Reshape	Elimina la dimensión de canal → (batch_size, alto, ancho)

Tabla 1\_Descripción Arquitectura Modelo I

Salida: Mapa 2D con la predicción del estado del fuego en el siguiente instante, con forma (alto, ancho).



#### 4.1.2 Características principales

- No capta dinámicas temporales.
- Ideal como base para comparar con modelos más complejos.
- Eficiente y rápido de entrenar.

### 4.1.3 Entrenamiento

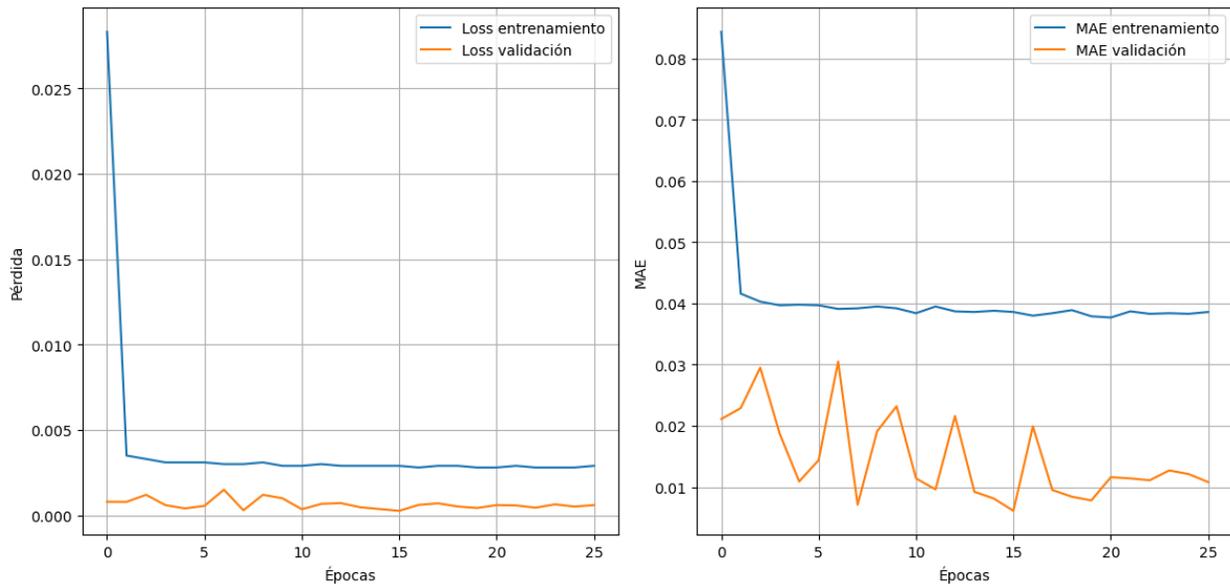


Ilustración 16\_Evolución entrenamiento Modelo I

#### Evolución del entrenamiento:

El modelo muestra una reducción rápida del error en las primeras épocas. A partir de la época 4, la mae se estabiliza en torno a 0.038–0.039 en entrenamiento. La val\_mae presenta más variabilidad, alcanzando un mínimo de 0.0061 en la época 16. El mecanismo de ReduceLRonPlateau se activa en las épocas 12, 17 y 22, lo que permite al modelo seguir afinando su rendimiento. La val\_loss es desde el inicio mucho menor que la loss (0.00079 vs. 0.0283), esto puede deberse al dropout del modelo

#### Posibles errores detectados:

No se observan errores significativos. La val\_mae es frecuentemente menor que la mae de entrenamiento, lo que indica buena generalización. Las oscilaciones en validación son normales y no reflejan sobreajuste.

#### Conclusión:

Modelo con rendimiento sólido y estable. Presenta buena capacidad de generalización y no muestra sobreajuste. La activación progresiva del ajuste de tasa de aprendizaje indica una convergencia controlada. Apto como baseline fiable.

### 4.1.4 Resultados obtenidos

Métrica	Valor
Loss (MSE)	0.00033
MAE	0.00767
RMSE	0.01826

<b>R<sup>2</sup> Score</b>	0.99024
----------------------------	---------

Tabla2\_Resultados Modelo I

Interpretación: Este modelo logra una capacidad predictiva en términos de error cuadrático y correlación con los datos reales, pese a su simplicidad. Aun así, no es suficiente para simular dinámicas complejas en secuencias largas.

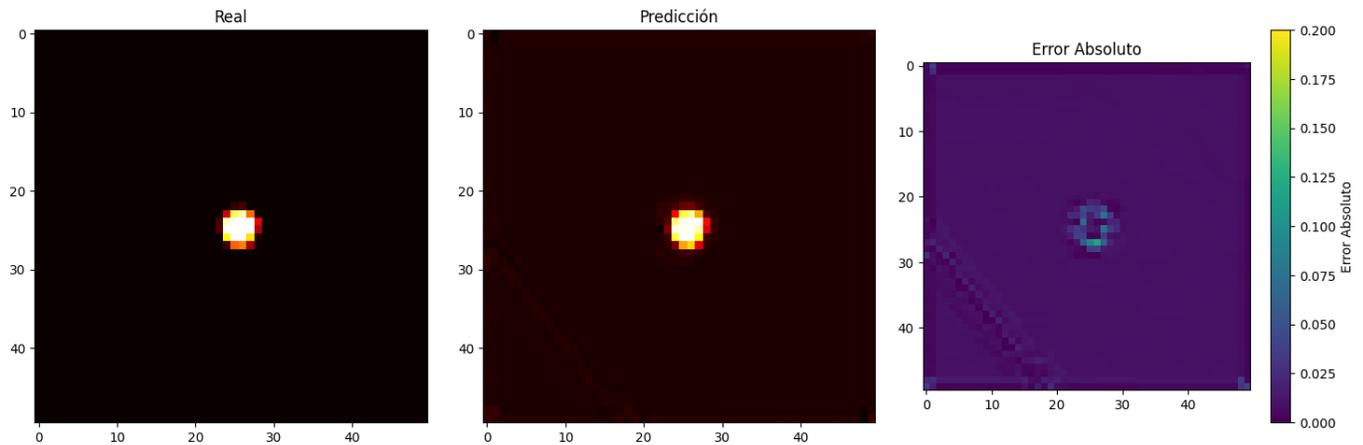


Ilustración 17\_Comparación Visual\_Model I

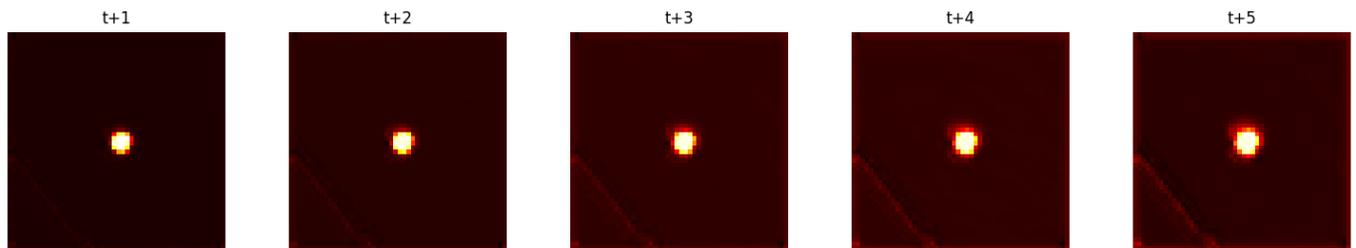


Ilustración 18\_Evolución Autoregresiva\_Model I

En la comparativa visual del modelo I se puede observar un error generado no solo en la zona de activación del fuego, también en el perímetro. El modelo no es capaz de predecir de forma adecuada los contornos de la capa Q (fuego) e introduce interferencias provocadas por otras capas.

En la imagen de evolución autoregresiva se observa como el mapa completo de va “encendiendo” e incrementa el valor Q en celdas fuera del perímetro.

Se decide incrementar la complejidad del modelo dando más capas convolucionales.

## 4.2 Modelo II - Improved\_Sequential\_Conv2D

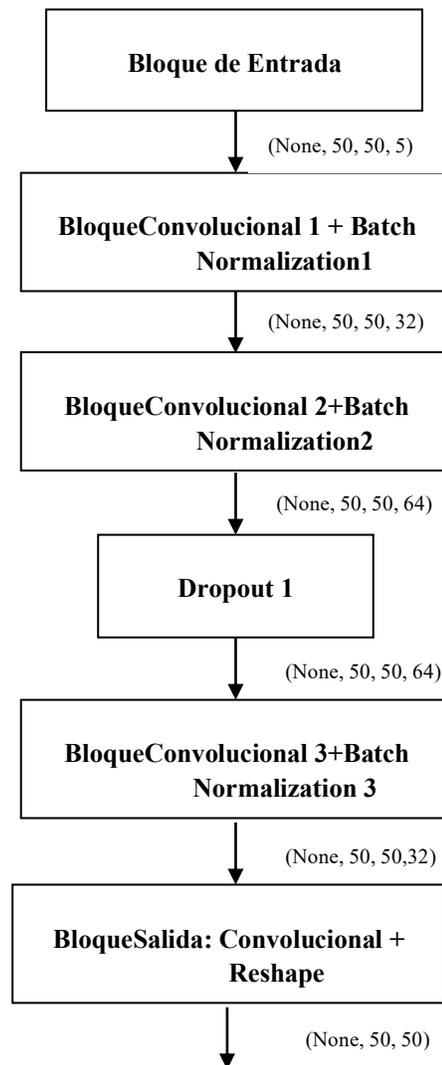
El **Modelo II** es una versión mejorada del Modelo I, diseñado para predecir la propagación o intensidad del fuego en una cuadrícula 2D. Mantiene una arquitectura Sequential exclusivamente convolucional, pero incrementa la capacidad expresiva al usar más filtros, un mayor número de capas y un bloque convolucional adicional. Su objetivo es capturar patrones espaciales más complejos manteniendo un tiempo de entrenamiento razonable y evitando aún el uso de historial temporal.

Al incorporar una capa adicional de convolución y más filtros, se logra una mayor extracción de patrones espaciales complejos, lo que permite una mejor aproximación de la dinámica del fuego en el paso siguiente.

### 4.2.1 Arquitectura del modelo

Capa	Descripción
Input(50, 50, 5)	Forma de entrada: (alto, ancho, canales)
Conv2D (3×3, 32)	Extrae características espaciales simples con activación ReLU y padding='same'.
BatchNormalization	Normalización de batch posterior a la primera convolución
Conv2D (3×3, 64)	Extrae patrones más complejos, con activación ReLU y padding='same'.
BatchNormalization	Normalización posterior a la segunda convolución
Dropout(0.25)	Dropout del 25% para mitigar el sobreajuste
Conv2D (3×3, 32)	Extrae patrones complejos, con activación ReLU y padding='same'.
BatchNormalization	Normalización posterior a la tercera convolución
Conv2D (1×1, 1)	Capa de salida: predicción por celda (activación lineal)
Reshape	Remueve la dimensión final → salida de forma (batch_size, alto, ancho)

Tabla3\_Descripción ArquitecturaModelo II



#### 4.2.2 Características principales

- Aumento de profundidad: Se añade una tercera capa convolutacional para extraer características más refinadas.
- Mayor capacidad: Se incrementa el número de filtros en las dos primeras capas convolucionales.
- Regularización: Dropout del 25% ayuda a evitar el sobreajuste.
- Predicción completamente espacial: Cada celda del mapa es predicha individualmente, sin aplanamiento ni capas densas.

### 4.2.3 Entrenamiento

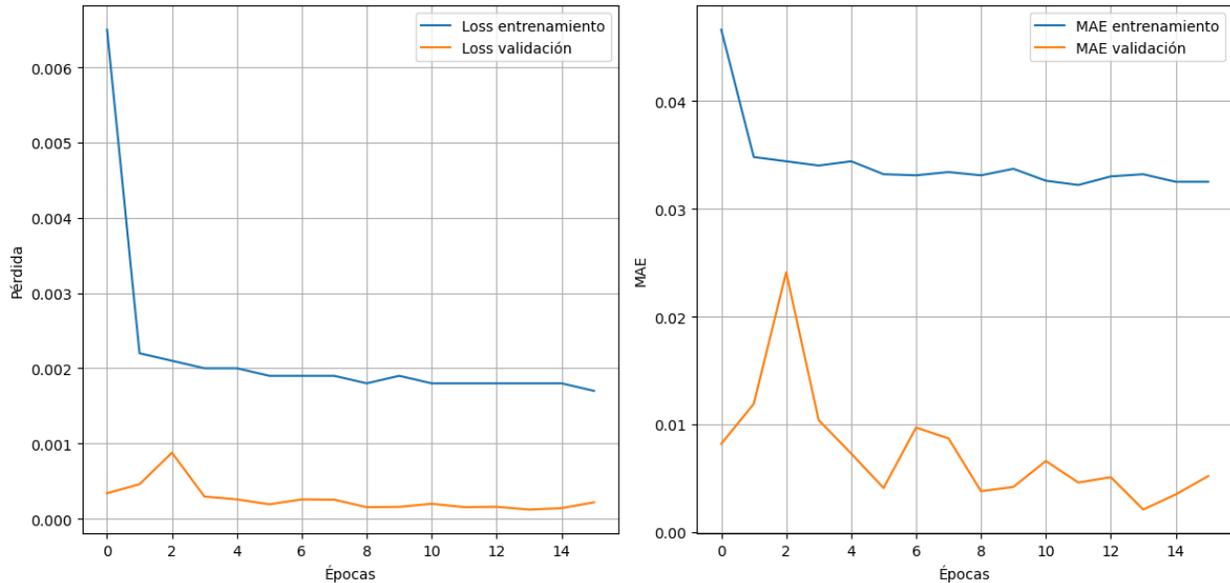


Ilustración 19\_Evolución entrenamiento Modelo II

#### Evolución del entrenamiento:

El modelo reduce el error de manera eficaz desde la primera época, con una *mae* de validación que alcanza un mínimo de 0.0021 en la época 14. La *mae* de entrenamiento se mantiene estable entre 0.032–0.034, mientras que la de validación fluctúa entre 0.0021 y 0.0241. El ajuste del *learning\_rate* se activa dos veces (épocas 11 y 16), y el entrenamiento finaliza por *early stopping* tras la segunda reducción. La *val\_loss* inicia mucho más baja que la *loss*, esto puede deberse a los dos *dropout* del modelo

#### Posibles errores detectados:

No se detecta sobreajuste. Al contrario, el modelo muestra una mejor generalización en varias épocas, con *val\_mae* consistentemente inferior a la de entrenamiento. Las oscilaciones observadas en validación son pequeñas y no afectan la estabilidad global.

#### Conclusión:

Modelo eficiente con aprendizaje robusto y generalización notable. Supera en consistencia y precisión al Modelo I. Se adapta bien al conjunto de validación sin señales de sobreajuste, lo que lo posiciona como una opción fuerte como base.

#### 4.2.4 Resultados obtenidos:

Métrica	Valor
Loss (MSE)	0.00024
MAE	0.00597
RMSE	0.01539
R <sup>2</sup> Score	0.99307

Tabla4\_Resultados Modelo II

Comparado con el Modelo I, mejora tanto el error cuadrático medio como el R<sup>2</sup> como resto de métricas, lo que indica que el modelo captura mejor la estructura del fenómeno.

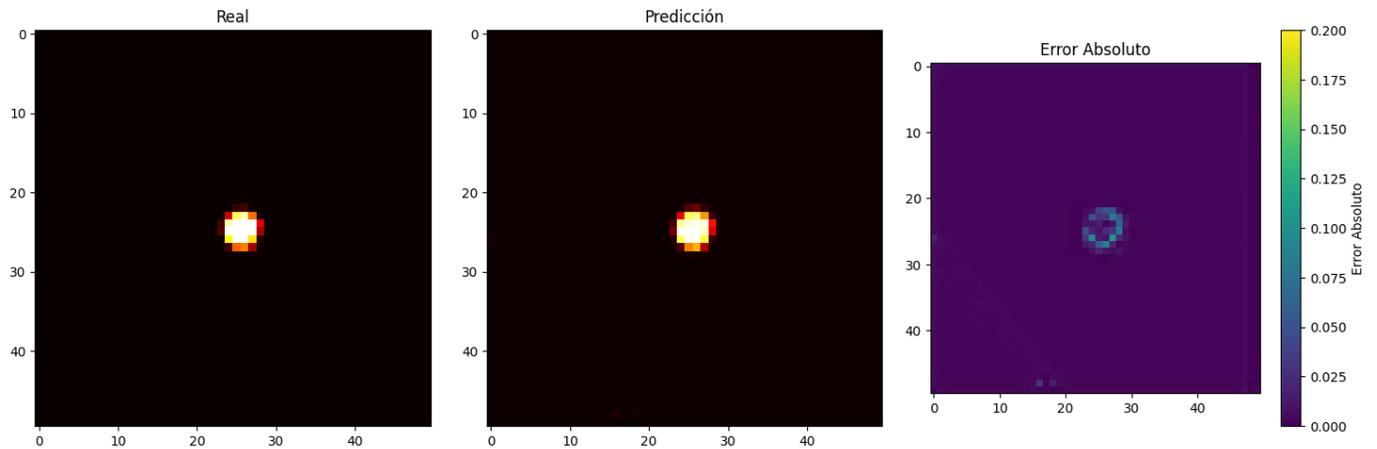


Ilustración 20\_Comparación Visual\_Model II

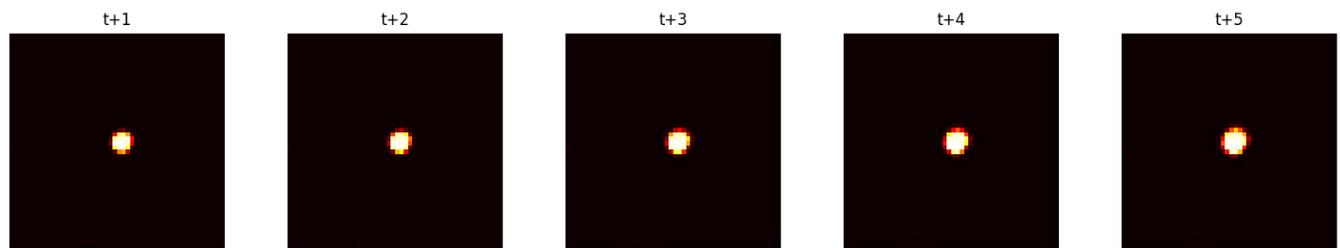


Ilustración 21\_Evolución Autoregresiva\_Model II

El modelo parece que ha sido capaz de minimizar el error en el exterior del área “prendida”, aunque se siguen observando puntos aislados, además mantiene un error significativo en esta misma área. Teniendo en cuenta que este error se origina en solo un paso, el error se multiplicará conforme la evolución se haga autorregresiva provocando una evolución del fuego muy diferente a la “teórica”.

### 4.3 Modelo III - ClassBased

Dados los problemas detectados en el anterior modelo, se proponen nuevas estrategias para poder mejorar el contorno de propagación. Este modelo introduce una estrategia más estructurada para procesar la información, separando explícitamente los canales de entrada en dos grupos:

- Canal clave: el estado actual del fuego, que recibe un tratamiento preferencial con mayor profundidad, filtros y regularización.
- Canales secundarios: velocidad del viento (X/Y), vegetación y altura, que se procesan en una rama paralela con bloques convolucionales inspirados en modelos anteriores.

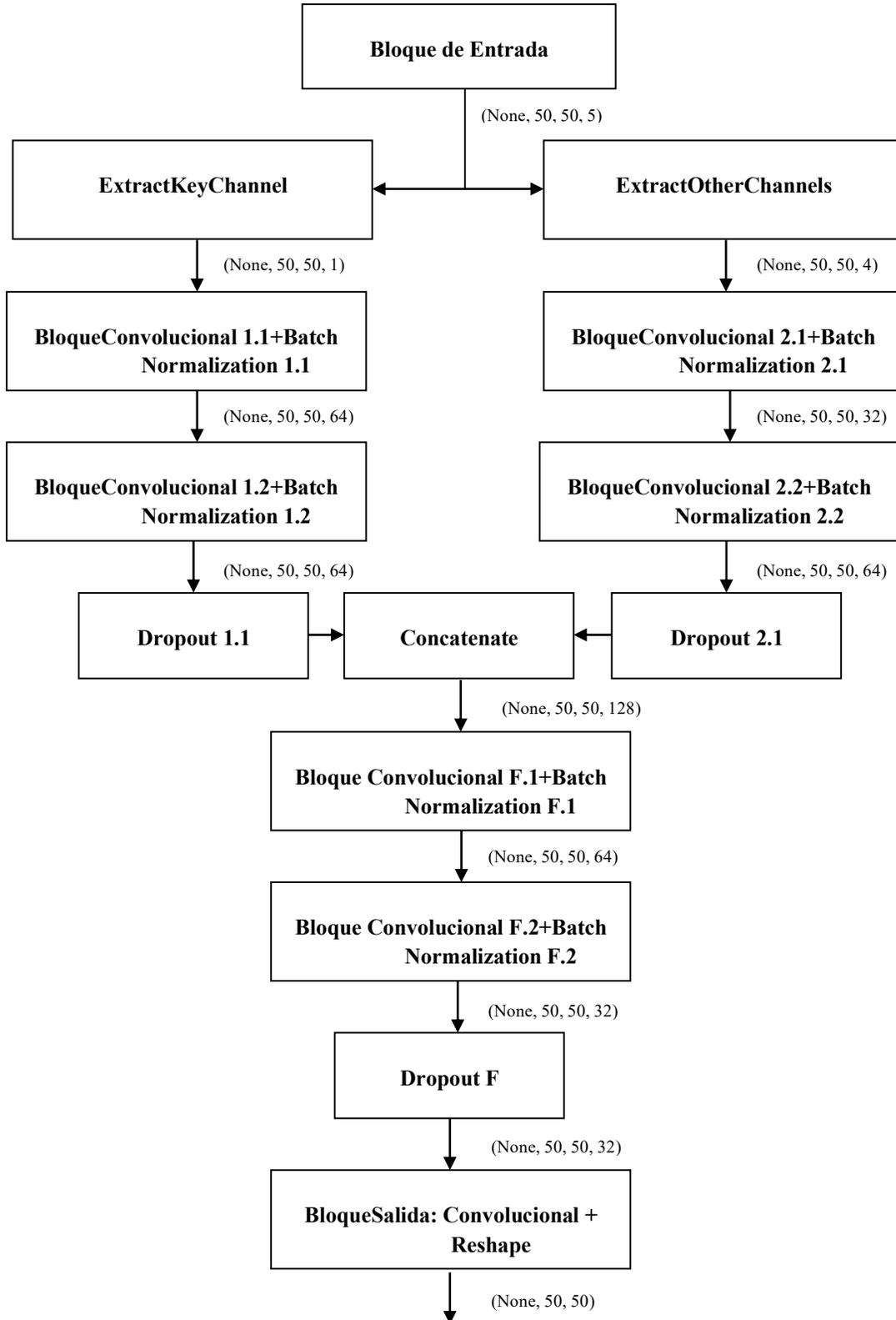
La salida del modelo de nuevo es un mapa 2D que representa la predicción espacial del fuego en el instante siguiente, sin pérdida de resolución.

#### 4.3.1 Arquitectura del modelo

Capa	Descripción
Input (50, 50, 5)	Forma de entrada: (alto, ancho, canales)
ExtractKeyChannel	Extrae canal clave (índice 0): estado del fuego → forma (alto, ancho, 1)
ExtractOtherChannels	Extrae canales secundarios (índices 1 a 4): viento(Vx,Vy), vegetación, altura
Conv2D_1.1 (3×3, 64)	Extrae características espaciales simples con activación ReLU ,padding='same' y regularización l2(0.001)
BatchNormalization_1.1	Normalización posterior a la 1. <sup>a</sup> convolución de KeyChannel
Conv2D_1.2 (3×3, 64)	Extrae características mas complejoscon activación ReLU ,padding='same' y regularización l2(0.001)
BatchNormalization_1.2	Normalización posterior a la 2. <sup>a</sup> convolución de fuego
Dropout (0.30)	Dropout del 30% para mitigar el sobreajuste
Conv2D_2.1 (3×3, 32)	Extrae características espaciales simples con activación ReLU y padding='same'.
BatchNormalization_2.1	Normalización posterior a la 1. <sup>a</sup> convolución de otros canales
Conv2D_2.2 (3×3, 64)	Extrae patrones más complejos, con activación ReLU y padding='same'.
BatchNormalization_2.2	Normalización posterior a la 2. <sup>a</sup> convolución de otros canales
Dropout (0.25)	Dropout del 25% para mitigar el sobreajuste
Concatenate	Combina las salidas de ambas ramas → forma (alto, ancho, 128)
Conv2D_F.1 (fusión) (3×3, 64)	Extrae características espacialescombinando las salidas de ambas ramas, padding 'same', activación 'relu', regularización l2(0.001)
BatchNormalization_F.1	Normalización posterior a la fusión
Conv2D_F.2 (fusión)	Reducción de los canales, padding 'same', activación 'relu'
BatchNormalization_F.2	Normalización posterior a la segunda convolución de fusión
Dropout (fusión)	Dropout del 25% para mitigar el sobreajuste
Conv2D (salida)	1 filtro 1×1, padding 'same', predicción por celda activación

	'linear'
Reshape	Remueve la dimensión final → salida de forma (batch_size, alto, ancho)

Tabla5\_Descripción ArquitecturaModelo III



### 4.3.2 Características principales

- Separación explícita de canales: El modelo distingue entre el canal clave y los canales secundarios para un procesamiento especializado.
- Mayor profundidad para el canal clave: Dos capas convolucionales con 64 filtros y mayor dropout aseguran un mejor modelado del estado del fuego.
- Arquitectura inspirada en el Modelo II para canales secundarios: Preserva profundidad y regularización sin sobreajustar esta rama.
- Fusión posterior: Las dos ramas se combinan y refinan mediante convoluciones adicionales para generar una salida coherente.
- Regularización consistente: Dropout y BatchNormalization en ambas ramas y en el bloque final.
- Predicción completamente espacial: La salida conserva la estructura espacial del input, sin capas densas ni aplanamientos.

### 4.3.3 Entrenamiento

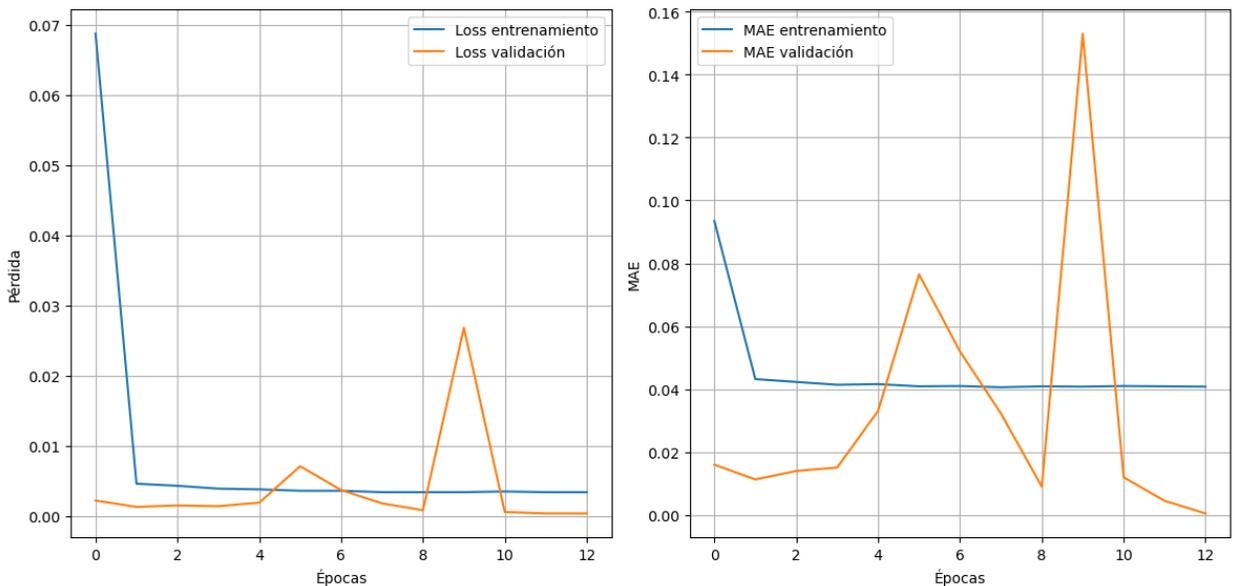


Ilustración 22\_Evolución entrenamiento Modelo III

#### Evolución del entrenamiento:

El modelo arranca con `mae` de validación muy baja (0.0160), pero presenta una evolución irregular. Aunque reduce el error en entrenamiento de forma progresiva (`mae` ~0.040–0.043), la `val_mae` oscila con brusquedad, alcanzando valores tan altos como 0.1530 y tan bajos como 0.0005. El `learning_rate` se ajusta una vez (época 7), y no hay *early stopping* hasta la época 13.

#### Posibles errores detectados:

El comportamiento errático de la `val_mae` sugiere inestabilidad en la generalización. Las fluctuaciones extremas (de 0.0045 a 0.1530 y de vuelta a 0.0005) no se corresponden con los

valores estables del conjunto de entrenamiento. Esto apunta a un sobreajuste intermitente o sensibilidad a ciertas muestras del conjunto de validación. La reducción de `learning_rate` ayuda parcialmente, pero no elimina la variabilidad.

**Conclusión:**

Modelo potente pero inestable en validación. Puede alcanzar errores mínimos impresionantes, pero tiene picos erráticos que comprometen su fiabilidad. Requiere mejoras en regularización o gestión del `learning_rate`. A pesar de su potencial, no es tan consistente como el Modelo II.

**4.3.4 Resultados obtenidos**

Métrica	Valor
Loss (MSE)	0.00021
MAE	0.00377
RMSE	0.01441
R <sup>2</sup> Score	0.99393

Tabla6\_ Resultados Modelo III

El modelo muestra una muy ligera mejora las métricas en comparación con el Modelo II (por ejemplo, el MAE pasa de 0.00597 a 0.00377 y el R<sup>2</sup> disminuye de 0.99307 a 0.99393), lo que sugiere que la mayor complejidad o profundidad ha aportado una pequeña mejora en precisión.

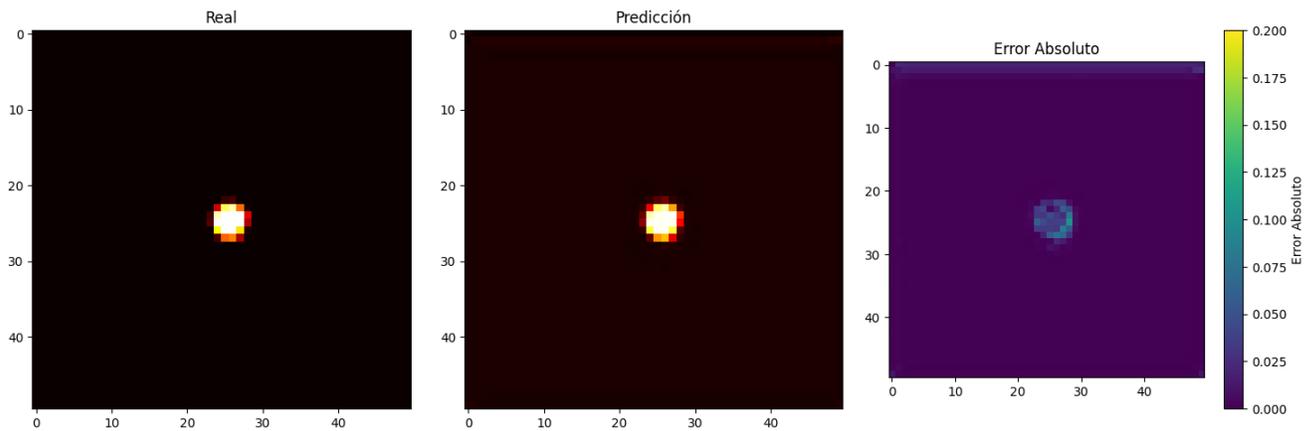


Ilustración 23\_Comparación Visual\_Model III

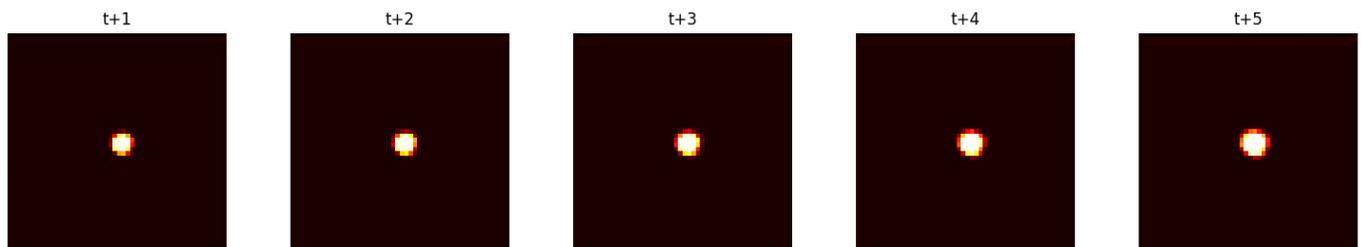


Ilustración 24\_Comparación Visual\_Model III

Dar una atención al canal clave puede estar ayudando a la mejora de las métricas, pero observando la comparación visual se puede observar que el error absoluto ahora se encuentra esparcido por toda el área quemada, y no solo en el contorno.

El modelo es más organizado, modular, y desde el punto de vista predictivo puro (métricas), el Modelo III ha sido superior. No obstante, este modelo no responde de forma adecuada al modelado de incendios con varios steps como se verá en el apartado 4.3.

Este modelo puede ser un buen punto de partida para probar nuevas estrategias, de acuerdo con nuestra metodología propuesta. Se proponen posibles mejoras como:

- Incluir atención espacial o mecanismos de ponderación entre ramas.
- Añadir convoluciones dilatadas en el bloque final.
- Aplicar un `FusionBlock` intermedio para mejorar la combinación de ramas.

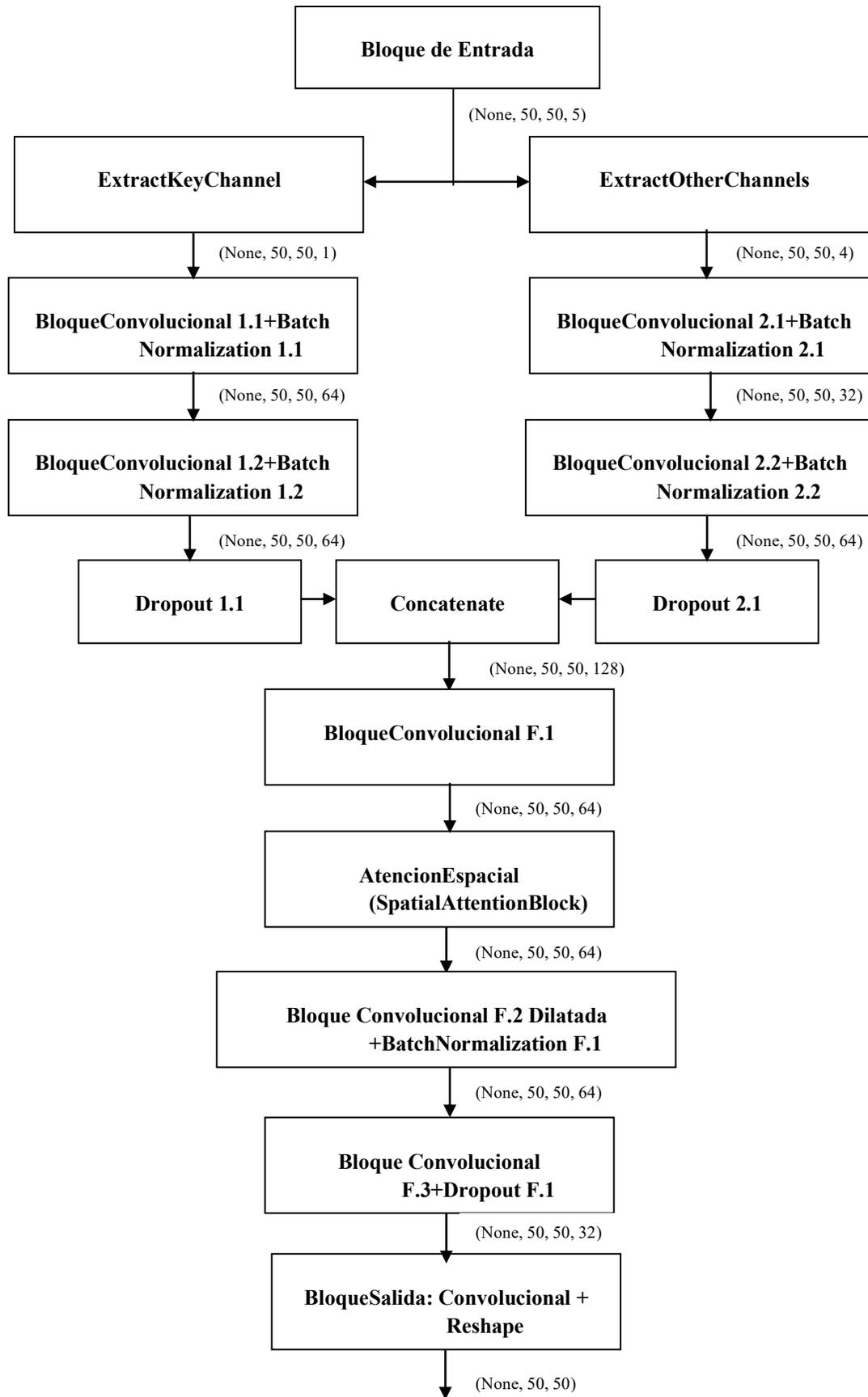
## 4.4 Modelo IV – Attn\_Fusion\_Dilated

Este modelo representa un enfoque avanzado y más expresivo para predecir la propagación espacial de incendios forestales. Es la evolución del modelo III: incorpora atención espacial, una fusión profunda entre ramas especializadas y convoluciones dilatadas para capturar contexto ampliado.

### 4.4.1 Arquitectura del modelo

Capa	Descripción
Input(50, 50, 5)	Forma de entrada: (alto, ancho, canales)
ExtractKeyChannel	Extrae canal clave (índice 0): estado del fuego → forma (alto, ancho, 1)
ExtractOtherChannels	Extrae canales secundarios (índices 1 a 4): viento(Vx,Vy), vegetación, altura
Conv2D_1.1(3×3, 64)	Extrae características espaciales simples con activación ReLU ,padding='same' y regularización l2(0.001)
BatchNormalization_1.1	Normalización posterior a la 1. <sup>a</sup> convolución de KeyChannel
Conv2D_1.2 (3×3, 64)	Extrae características mas complejos con activación ReLU ,padding='same' y regularización l2(0.001)
BatchNormalization_1.2	Normalización posterior a la 2. <sup>a</sup> convolución de fuego
Dropout_1.1(0.20)	Dropout del 20% para mitigar el sobreajuste
Conv2D_2.1(3×3, 32)	Extrae características espaciales simples con activación ReLU y padding='same'.
BatchNormalization_2.1	Normalización posterior a la 1. <sup>a</sup> convolución de otros canales
Conv2D_2.2(3×3, 64)	Extrae patrones más complejos, con activación ReLU y padding='same'.
BatchNormalization_2.2	Normalización posterior a la 2. <sup>a</sup> convolución de otros canales
Dropout_2.1 (0.25)	Dropout del 25% para mitigar el sobreajuste
Concatenate	Combina las salidas de ambas ramas → forma (alto, ancho, 128)
Conv2D_F.1 (fusión) (3×3, 64)	Extrae características espaciales combinando las salidas de ambas ramas, padding 'same', activación 'relu',
spatial_attention_block	Atención espacial estilo Squeeze& Excite con compresión y expansión por canal
Conv2D_F_2 (dilatada) (fusión) (3×3, 64)	64 filtros 3×3, activación ReLU, dilationrate = 2, regularización L2
BatchNormalization F.1	
Conv2D_F.3 (fusión) (3×3, 32)	32 filtros 3×3, activación ReLU
Dropout(0.25)_F.1	25% de apagado final
Conv2D (salida)	1 filtro 1×1, activación lineal, padding 'same'
Reshape	Remueve dimensión de canal final → (alto, ancho)

Tabla7\_Descripción ArquitecturaModelo IV



#### 4.4.2 Características principales

- Ramas especializadas: Separación explícita entre el canal clave (estado del fuego) y los canales secundarios (viento, vegetación, altura), permitiendo un procesamiento diferencial.
- Fusión intermedia enriquecida: Uso de un bloque de fusión con convolución  $3 \times 3$  posterior a la concatenación de ramas para integración de características complementarias.
- Mecanismo de atención espacial (Squeeze& Excite): Enfatiza regiones espacialmente relevantes tras la fusión, modulando la activación por canal con base en su importancia global.
- Convoluciones dilatadas: Ampliación del campo receptivo sin pérdida de resolución espacial, útil para capturar interacciones espaciales más extensas.
- Regularización robusta: Dropout en ambas ramas y regularización L2 en capas clave para mitigar el sobreajuste.
- Predicción completamente espacial: La salida conserva las dimensiones espaciales sin aplanamiento, permitiendo una predicción detallada por celda.

#### 4.4.3 Entrenamiento

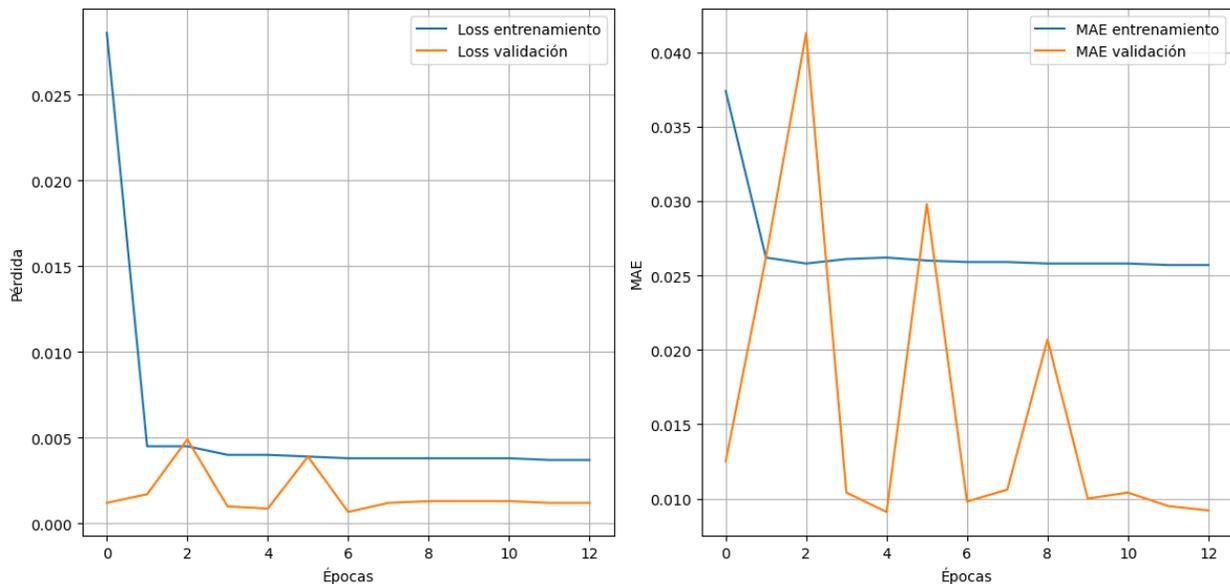


Ilustración 25\_Evolución entrenamiento Modelo IV

#### Evolución del entrenamiento:

El modelo parte de un  $mae$  de validación inicial bajo (0.0125) y mejora progresivamente el  $mae$  en entrenamiento hasta estabilizarse cerca de 0.0257–0.0258. La  $val\_mae$ , aunque algo ruidosa (fluctúa entre 0.0091 y 0.0413), tiende a estabilizarse en torno a 0.009–0.010. El  $learning\_rate$  no se ajusta en estas primeras 13 épocas, y no se activa *early stopping*.

**Posibles errores detectados:**

Aunque hay picos ocasionales en `val_mae`, no hay señales claras de sobreajuste. El error en validación vuelve a bajar consistentemente y permanece próximo al entrenamiento. El ruido parece controlado, y la red muestra señales de haber encontrado un mínimo estable.

**Conclusión:**

Modelo estable y confiable, con generalización sólida tras unas pocas fluctuaciones iniciales. Aunque el `mae` de validación no alcanza mínimos tan extremos como en el Modelo III, su comportamiento es más consistente, lo que lo hace más fiable para despliegue o evaluación comparativa.

**4.4.4 Resultados obtenidos**

Métrica	Valor
Loss (MSE)	0.00019
MAE	0.00324
RMSE	0.01373
R <sup>2</sup> Score	0.99448

Tabla8\_ Resultados Modelo IV

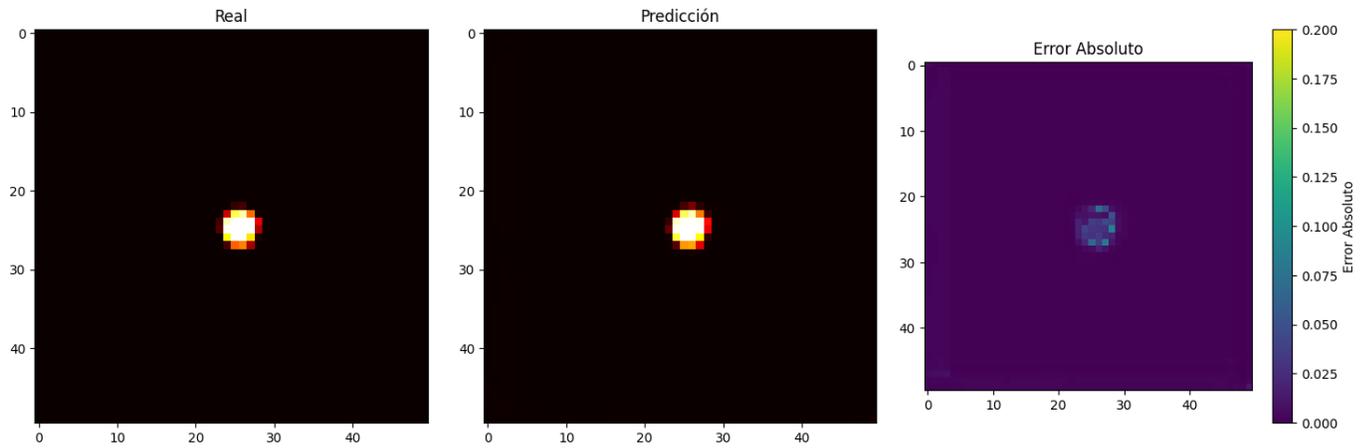
Este modelo ha obtenido los mejores resultados hasta ahora, con un  $R^2$  de 0.9947, lo que indica una predicción extremadamente ajustada a los datos reales.

El uso de atención espacial y convoluciones dilatadas parece haber sido clave para mejorar la captura de relaciones espaciales, especialmente en patrones de propagación más complejos.

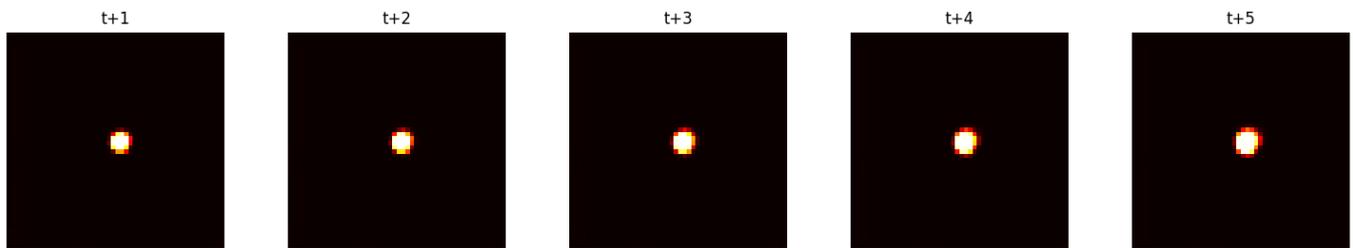
A pesar de la complejidad añadida, el modelo no ha sufrido sobreajuste significativo, lo cual sugiere que la combinación de regularización L2 y Dropout fue bien balanceada.

La reducción del MAE a menos de 0.004 demuestra una mejora tangible en la precisión por celda, lo cual es crucial para aplicaciones de predicción geoespacial.

Se podría conservar esta arquitectura como base avanzada y explorar variantes de atención (por ejemplo, CBAM o atención mixta espacial + canal) o integración con contextual residual blocks para mayor rendimiento.



**Ilustración 26\_Comparación Visual\_Model IV**



**Ilustración 27\_Evolución Autoregresiva\_Model IV**

Visualmente se ve una mejora con respecto a modelos anteriores, sin embargo, el error en la zona interior del área de fuego persiste, esto puede originar fuegos en zonas ya combustionadas.

Habrà que estudiar el comportamiento del modelo a largo plazo para ver su idoneidad.

## 4.5 Modelo V - Global\_Flatten\_Dense

Este modelo explora la incorporación de una **relación global espacial** utilizando un bloque Flatten + Dense, buscando capturar dependencias más amplias en todo el mapa.

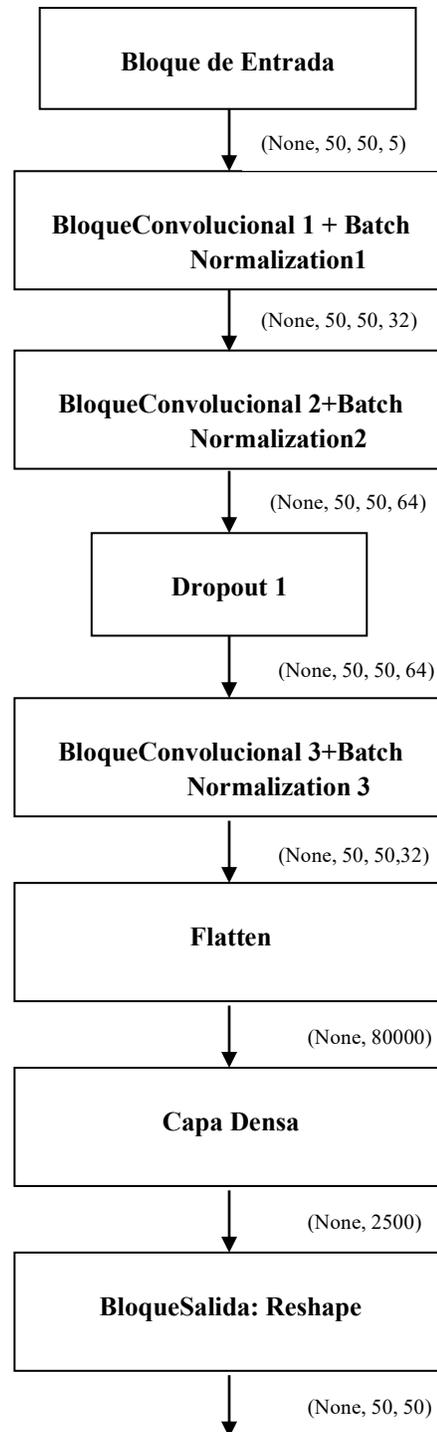
A diferencia de los modelos anteriores que son puramente convolucionales, este diseño introduce una capa densa conectada a todo el mapa para integrar información espacial completa.

El motivo del estudio de búsqueda de dependencias globales y no solo locales, se debe, sobre todo, a la capa viento, cuya dependencia de la topografía global de nuestro escenario repercute en la propagación del fuego.

### 4.5.1 Arquitectura del modelo

Capa	Descripción
Input(50, 50, 5)	Forma de entrada: (alto, ancho, canales)
Conv2D (3×3, 32)	Extrae características espaciales simples con activación ReLU y padding='same'.
BatchNormalization	Normalización de batch posterior a la primera convolución
Conv2D (3×3, 64)	Extrae patrones más complejos, con activación ReLU y padding='same'.
BatchNormalization	Normalización posterior a la segunda convolución
Dropout(0.25)	Dropout del 25% para mitigar el sobreajuste
Conv2D (3×3, 32)	Extrae patrones complejos, con activación ReLU y padding='same'.
BatchNormalization	Normalización posterior a la tercera convolución
Flatten	Aplanamiento total del mapa (globalización de relaciones espaciales)
Dense(input_shape[0] * input_shape[1])=(50*50)	Aplica transformaciones no lineales sobre el vector global, permitiendo al modelo aprender relaciones a gran escala entre todas las características. Capa totalmente conectada con activación ReLU, salida: (alto × ancho)
Reshape	Reconstrucción de la forma 2D → (alto, ancho)

Tabla 9 Descripción Arquitectura Modelo V



#### 4.5.2 Características principales

- Arquitectura híbrida: Combina capas convolucionales locales con una capa densa global, intentando mezclar detalle local con contexto general.

- Relación global explícita: El uso de Flatten + Dense permite que la red tenga una vista "global" de toda la entrada, lo que podría ser útil para capturar patrones amplios.
- Sin ramas ni separación de canales: Todos los canales (clave y secundarios) son tratados de manera unificada, sin especialización.
- Predicción espacial reconstruida: Tras la capa densa, la salida es remodelada a la forma original del mapa para conservar la estructura espacial.
- Regularización ligera: Solo se aplica dropout en una capa intermedia, con posible insuficiencia para prevenir sobreajuste o capturar variabilidad compleja.

### 4.5.3 Entrenamiento

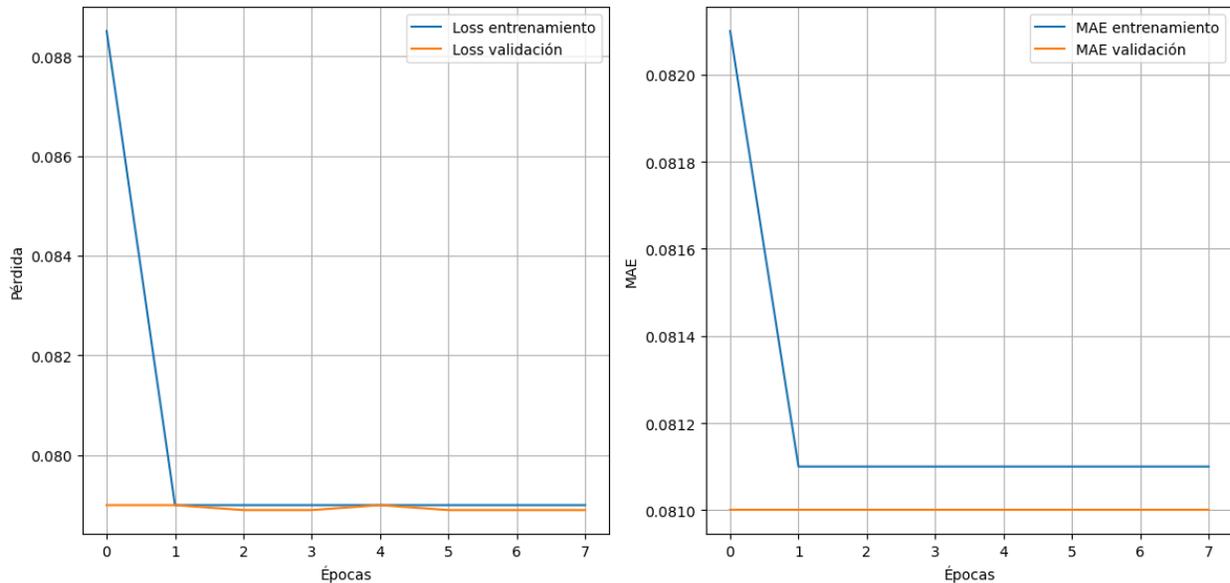


Ilustración 28\_Evolución entrenamiento Modelo V

#### Evolución del entrenamiento:

El modelo muestra una clara estancación desde el principio. Aunque la primera época parte de una pérdida y MAE aceptables ( $val\_mae = 0.0810$ ), no se observa ninguna mejora sustancial ni en entrenamiento ni en validación durante las épocas entrenadas. Los valores se mantienen virtualmente constantes, lo que sugiere que el modelo ha quedado bloqueado o no ha aprendido nada relevante tras la primera época.

#### Posibles problemas detectados:

- Sin aprendizaje real tras la primera época: Las métricas no mejoran en absoluto, lo que indica un modelo congelado o una configuración ineficaz.
- Arquitectura posiblemente mal ajustada: o bien está sobredimensionada y no generaliza, o bien carece de capacidad para capturar la señal relevante.
- Tiempo de entrenamiento excesivo:
- Pérdida y MAE altas (muy lejos del rendimiento del Modelo IV, por ejemplo).

**Conclusión:**

El Modelo V es altamente ineficiente tanto en rendimiento como en costo computacional. Aunque numéricamente estable, no logra aprender nada útil tras la primera época, lo que lo descarta como candidato competitivo.

**4.5.4 Resultados obtenidos**

Métrica	Valor
Loss (MSE)	0.03557
MAE	0.03715
RMSE	0.18859
R <sup>2</sup> Score	-0.04038

Tabla10\_Resultados Modelo V

Este modelo tuvo un rendimiento significativamente inferior comparado con los modelos anteriores, obteniendo un R<sup>2</sup> negativo (-0.04038), lo cual indica que predice peor que simplemente promediar los valores reales.

La inclusión de una capa Flatten + Dense no resultó beneficiosa en este caso; puede haber introducido una pérdida de estructura espacial importante o una incapacidad para aprender patrones locales cruciales para la dinámica del fuego.

El tratamiento homogéneo de todos los canales sin separación de funciones ni priorización del canal clave también pudo haber limitado su capacidad predictiva.

Este enfoque no parece adecuado para tareas altamente espaciales como la predicción de mapas de propagación de incendios, donde las dependencias locales y la estructura espacial son fundamentales.

Servirá como línea base negativa para evaluar la importancia de preservar las estructuras convolucionales y el tratamiento diferenciado de los canales en este tipo de problemas.

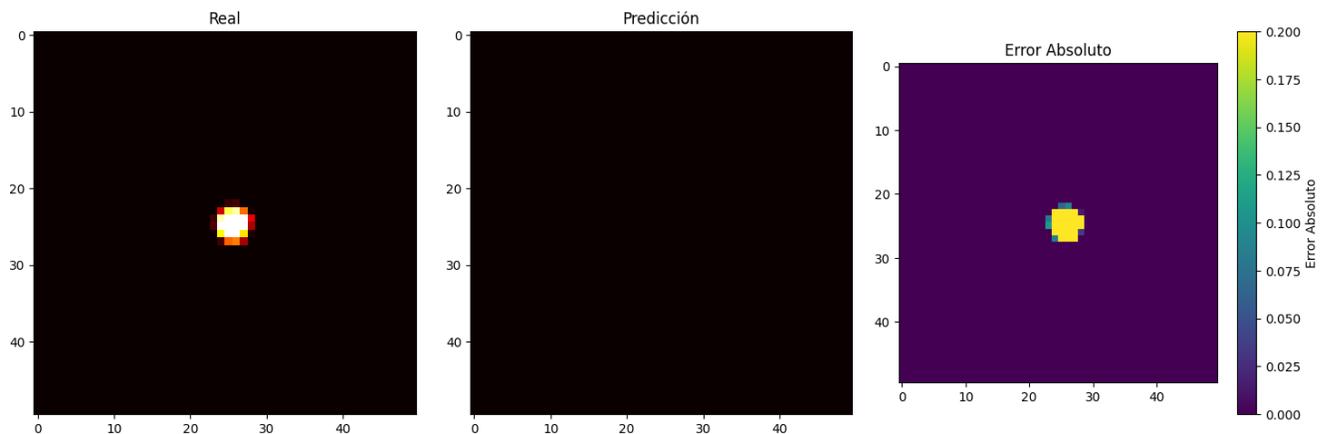


Ilustración 29\_Comparación Visual\_Model V

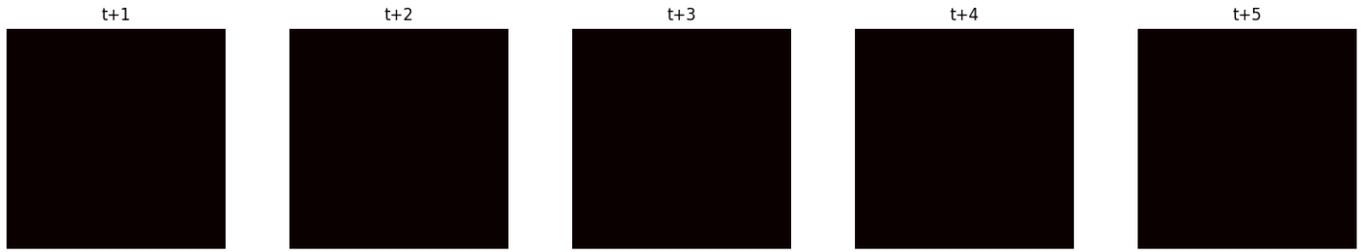


Ilustración 30\_Evolución Autoregresiva\_Model V

Error absoluto en el modelo.



## 4.6 Modelo VI – Local\_Global\_Fusion

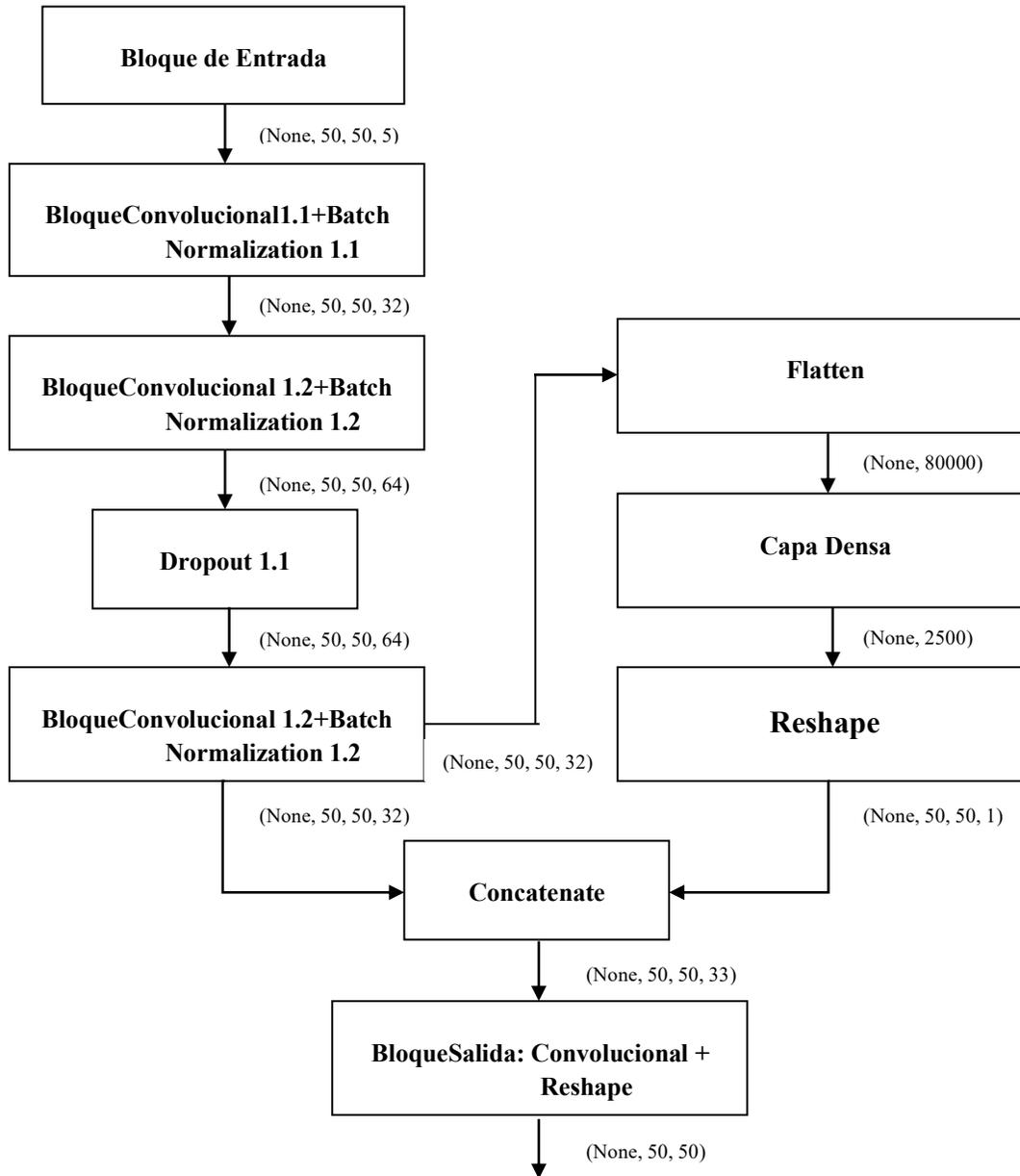
Como lección aprendida del modelo previo se propone no dejar de lado el estudio global (dense) por completo, pero sin perder el contexto o dependencia espacial que tiene nuestro modelado.

Este modelo implementa una arquitectura híbrida que combina el aprendizaje local basado en convoluciones con una ruta global de Flatten + Dense. Su objetivo es capturar patrones locales finos (como bordes, cambios suaves en el terreno) y relaciones espaciales globales (por ejemplo, el impacto amplio del viento o la orografía).

### 4.6.1 Arquitectura del modelo

Capa	Descripción
Input (50, 50, 5)	Forma de entrada: (alto, ancho, canales)
Conv2D_1.1 (3×3, 32)	Extrae características espaciales simples con activación ReLU y padding='same'.
BatchNormalization_1.1	Normalización de batch posterior a la primera convolución
Conv2D_1.2 (3×3, 64)	Profundiza en la extracción de características locales., padding 'same', activación ReLU
BatchNormalization_1.2	Normalización de batch posterior a la segunda convolución
Dropout_1.1 (0.25)	Reduce el sobreajuste. Dropout del 25% para regularización
Conv2D_1.3 (3×3, 32)	Refina las características locales antes de la, padding 'same', activación ReLU
BatchNormalization_1.3	Normalización de batch posterior a la tercera convolución
Flatten	Ruta global: aplanamiento completo del mapa
Dense(input_shape[0] * input_shape[1])=(50*50)	Capa densa global con activación ReLU, salida: (alto × ancho)
Reshape	Reconstrucción de la forma 2D → (alto, ancho)
Concatenate	Fusión de ruta local y global a lo largo del canal
Conv2D (salida)	1 filtro 1×1, activación lineal, preserva estructura espacial
Reshape	Salida final: (alto, ancho)

Tabla 11 Descripción Arquitectura Modelo VI



#### 4.6.2 Características principales

- Ruta local convolutacional: Tres bloques convolutacionales con `BatchNormalization` y `Dropout`, lo que permite detectar patrones locales robustos y prevenir sobreajuste.
- Ruta global densamente conectada: La salida del bloque convolutacional se aplanan, pasa por una `capa Dense` y se reconstruye a la forma espacial. Esto permite capturar dependencias a largo alcance entre regiones del mapa.

- Fusión multimodal: Las rutas local y global se concatenan, integrando lo aprendido en ambas perspectivas antes de generar la predicción.
- Salida espacial estructurada: Usa una Conv2D (1x1) y un Reshape para mantener la coherencia espacial en la salida final.
- Buena regularización: Uso moderado de Dropout y BatchNormalization permite estabilidad sin pérdida de capacidad.

### 4.6.3 Entrenamiento

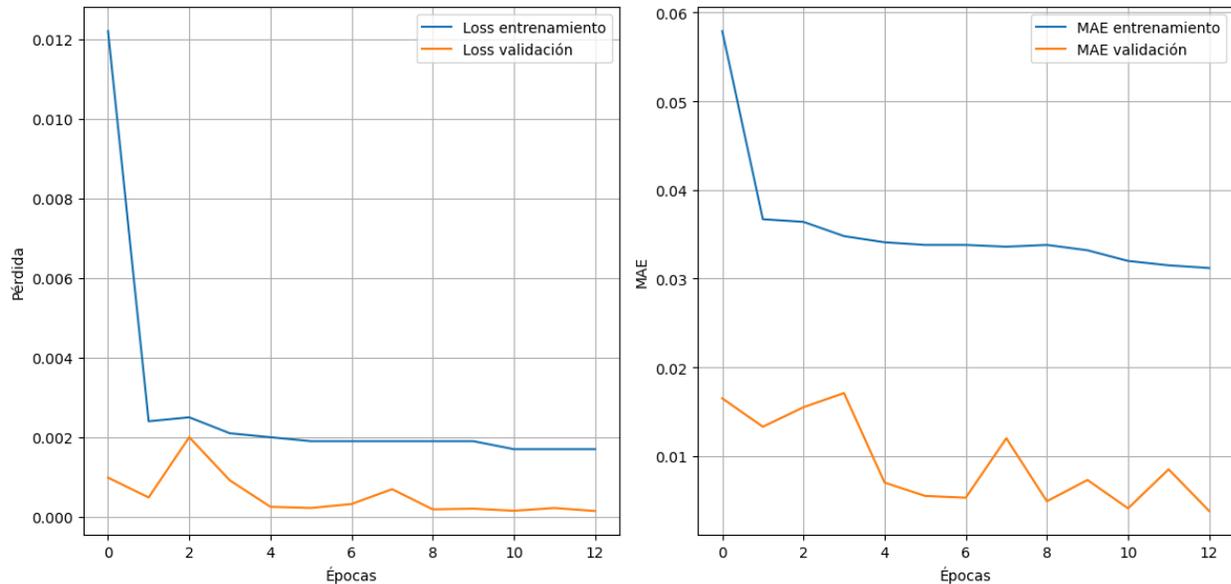


Ilustración 31\_Evolución entrenamiento Modelo VI

#### Evolución del entrenamiento:

Este modelo comienza con un `val_mae` inicial de 0.0165, que desciende rápidamente a valores bajos ( $< 0.01$ ) a partir de la época 4. Durante las épocas 4–12, el `val_mae` cae hasta un mínimo de 0.0038 (época 12) mientras el `mae` en entrenamiento desciende de 0.0579 a 0.0312.

El `ReduceLRonPlateau` se activa en la época 10, lo que sugiere un intento de estabilizar el entrenamiento, tras una ligera subida en `val_mae` en la época 7. El descenso posterior en `val_mae` tras la reducción de tasa de aprendizaje indica que esta estrategia fue efectiva.

#### Posibles errores detectados:

No hay señales claras de sobreajuste. El `val_mae` y `mae` están bien acoplados. Se observan pequeñas fluctuaciones esperables, pero el modelo mejora progresivamente con una tendencia claramente decreciente en validación, lo que sugiere una buena capacidad de generalización.

#### Conclusión:

Modelo muy sólido. Muestra un comportamiento suave y controlado. Su coste computacional es mayor, pero los beneficios en precisión lo justifican.

#### 4.6.4 Resultados obtenidos

Métrica	Valor
Loss (MSE)	0.00021
MAE	0.00424
RMSE	0.01439
R <sup>2</sup> Score	0.99395

Tabla12\_Resultados Modelo VI

Este modelo logró un muy buen rendimiento, con un **R<sup>2</sup> de 0.99695**, comparable al mejor modelo previo (Modelo IV), mostrando que la combinación de rutas local y global puede ser sinérgica.

El uso de la ruta `Dense` aporta relación espacial global sin eliminar totalmente la estructura espacial como ocurría en el Modelo V. Esto demuestra que la integración es efectiva cuando se mantiene el contexto local como base.

La fusión explícita de rutas permite que el modelo aprenda cuándo usar información local o global, lo que puede ser útil ante patrones del fuego que dependen tanto de condiciones locales (combustible, elevación puntual) como globales (viento, pendiente amplia).

Aunque los resultados son muy buenos, la complejidad del modelo aumenta en cuanto a parámetros debido a la capa densa. En escenarios con menos datos o mayor ruido, podría ser más penoso al sobreajuste si no se ajusta el `Dropout` o la regularización.

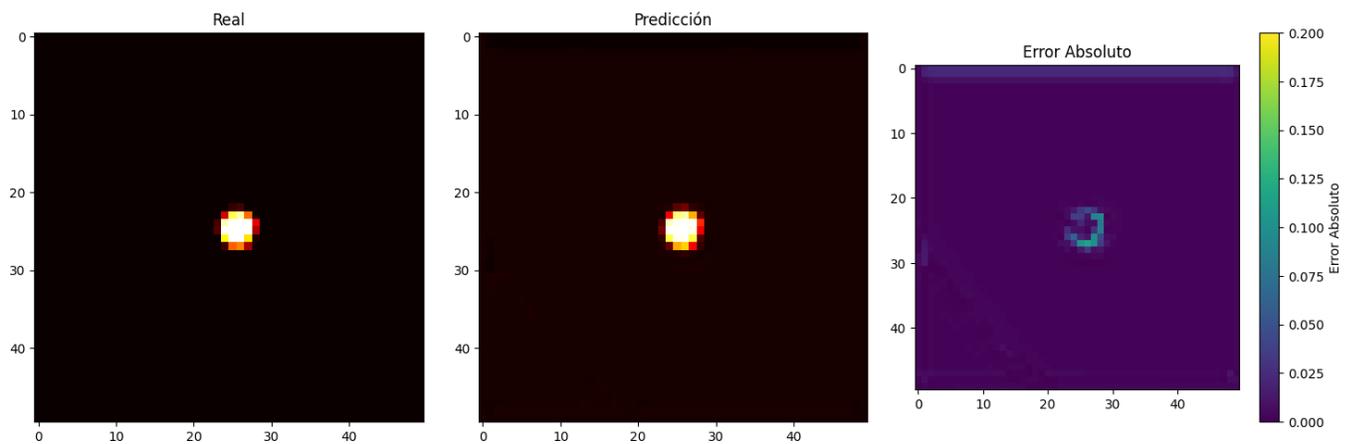
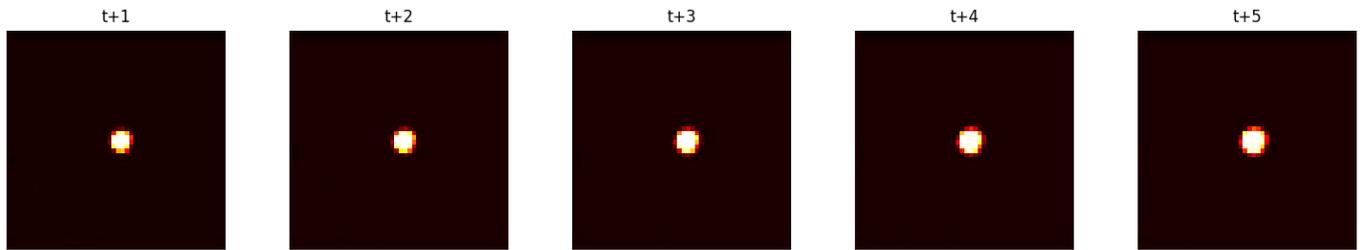


Ilustración 32\_Comparación Visual\_Model VI



**Ilustración 33\_Evolución Autoregresiva\_Model VI**

Visualmente Los resultados parecen bastante alentadores, el error se reduce al contorno de la zona de fuego y se reduce tanto en el interior de este área como en el exterior.

Viendo los buenos resultados obtenidos, estudiaremos otros modelos que nos permitan combinar dependencias locales y globales.

## 4.7 Modelo VIII - UNet\_Segmentation

Este modelo representa una implementación mejorada de U-Net, una arquitectura ampliamente utilizada para tareas de segmentación semántica.

Su objetivo es producir mapas de fuego espacialmente coherentes, capturando tanto **características de contexto global** como **detalles espaciales finos**.

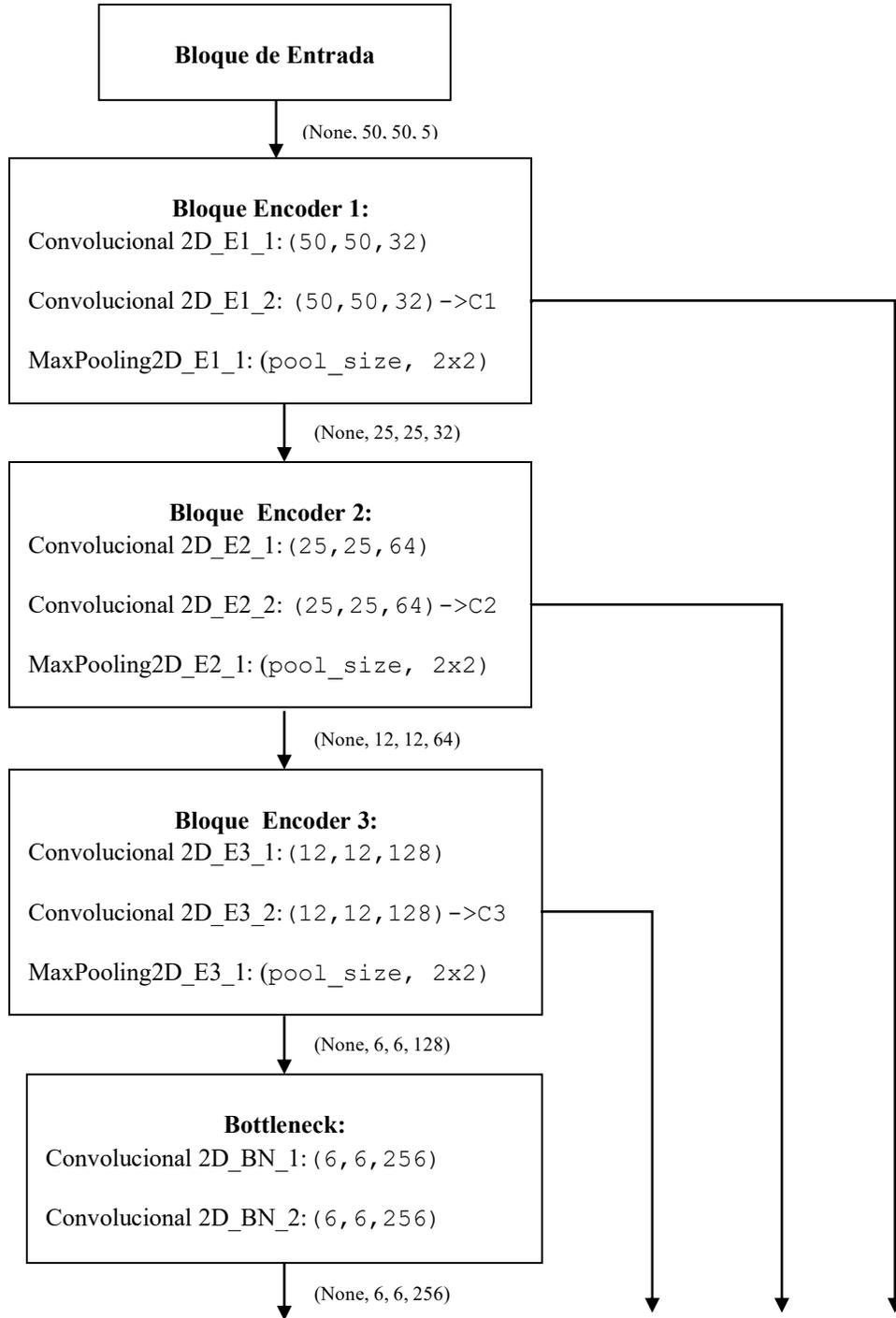
Se caracteriza por su encoder (ruta de contracción), decoder (ruta de expansión) y conexiones de salto que permiten la fusión de características de los diferentes niveles de abstracción para una localización precisa

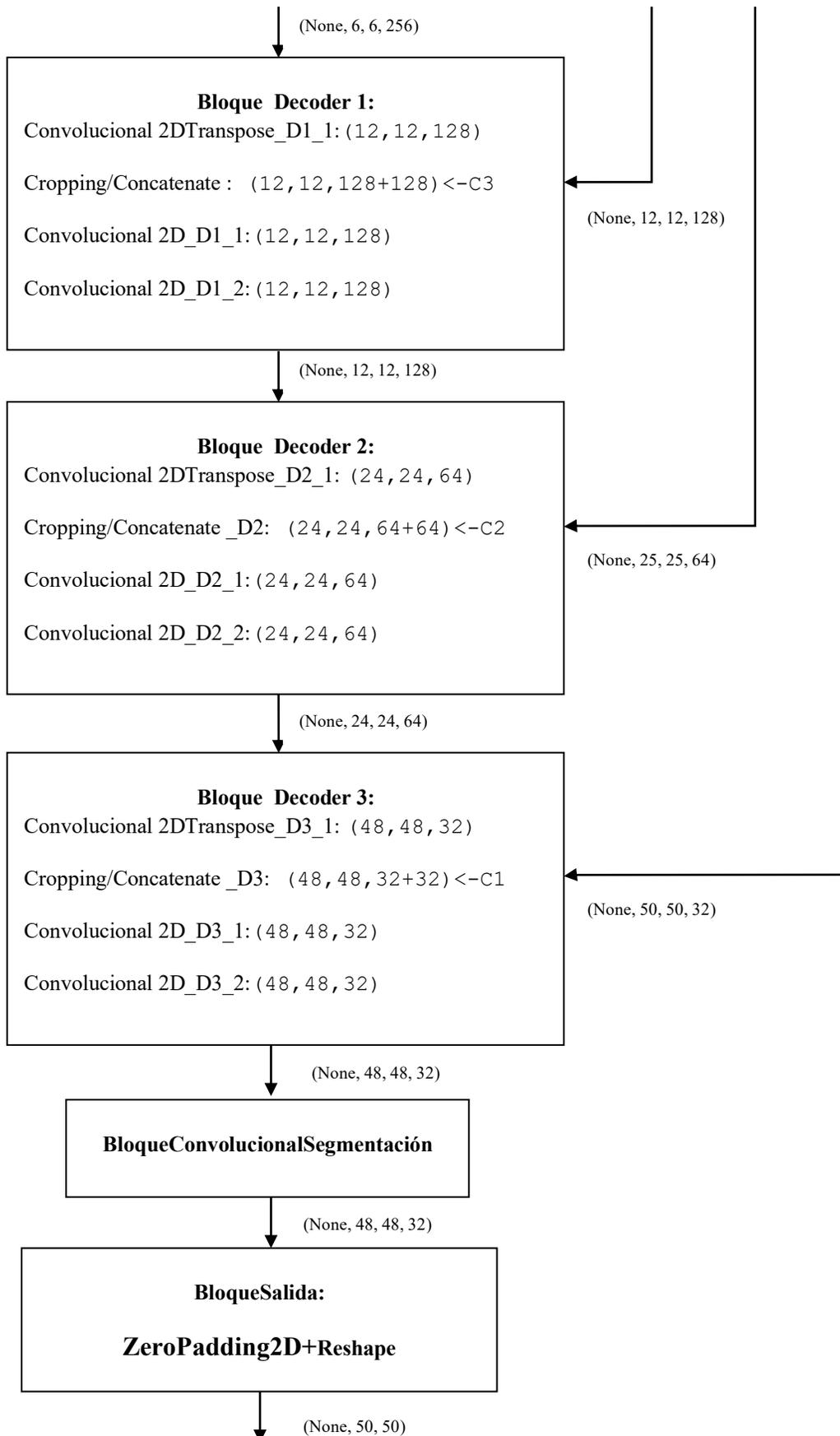
### 4.7.1 Arquitectura del modelo

Capa	Descripción
Input (50, 50, 5)	Forma de entrada: (50, 50, canales)
Conv2D_E1_1 (3×3, 32)	Extrae características espaciales simples con activación ReLU ,padding='same'(enc_conv1_1)
Conv2D_E1_2 (3×3, 32)	Extrae características mas complejas con activación ReLU ,padding='same' (enc_conv1_2)
MaxPooling2D_E1_1 (2×2)	Reducción de la resolución espacial. Pooling 2×2 (enc_pool1) – Reducción espacial a 25×25
Conv2D_E2_1 (3×3, 64)	Extrae características espaciales, padding='same', activación ReLU (enc_conv2_1)
Conv2D_E2_2 (3×3, 64)	Extrae características espaciales, padding='same', activación ReLU (enc_conv2_2)
MaxPooling2D_E2_1 (2×2)	Pooling 2×2 (enc_pool2) – Reducción espacial a 12×12
Conv2D_E3_1 (3×3, 128)	Extrae características espaciales 128 filtros 3×3, padding='same', activación ReLU (enc_conv3_1)
Conv2D_E3_2 (3×3, 128)	Extrae características espaciales 128 filtros 3×3, padding='same', activación ReLU (enc_conv3_2)
MaxPooling2D_E3_1 (2×2)	Pooling 2×2 (enc_pool3) – Reducción espacial a 6×6
Conv2D_BN_1 (3×3, 256)	256 filtros 3×3, padding='same', activación ReLU (bottleneck_conv1)
Conv2D_BN_2 (3×3, 256)	256 filtros 3×3, padding='same', activación ReLU (bottleneck_conv2)
Conv2DTranspose_D1_1 (2×2, 128)	Upsampling (aumento de resolución) en un tensor de entrada. 128 filtros 2×2, strides=2, padding='same' (dec_deconv1) – Upsampling a 12×12
Cropping2D (opcional)	Ajuste de $c_3$ si difiere de $u_3$ para concatenación
Concatenate	Combina las salidas de ambas ramas , fusión $u_3 + c_3$ (dec_concat1)
Conv2D_D1_2 (3×3, 128)	Extrae características espaciales 128 filtros 3×3, padding='same', activación ReLU (dec_conv1_1)

Conv2D_D1_3 (3×3, 128)	Extrae características espaciales 128 filtros 3×3, padding='same', activación ReLU (dec conv1 2)
Conv2DTranspose_D2_1 (2×2, 64)	Upsampling (aumento de resolución) en un tensor de entrada.64 filtros 2×2, strides=2, padding='same' (dec deconv2) – Upsampling a 25×25
Cropping2D_D2_1	Ajuste de c2 si difiere de u2 para concatenación. Opcional.
Concatenate_D2_1	Combina las salidas de ambas ramas , fusión u2 + c2 (dec concat2)
Conv2D_D2_2 (3×3, 64)	Extrae características espaciales.64 filtros 3×3, padding='same', activación ReLU (dec conv2 1)
Conv2D_D2_3 (3×3, 64)	Extrae características espaciales.64 filtros 3×3, padding='same', activación ReLU (dec conv2 2)
Conv2DTranspose_D3_1 (2×2, 32)	Upsampling (aumento de resolución) en un tensor de entrada.32 filtros 2×2, strides=2, padding='same' (dec deconv3) – Upsampling a 50×50
Cropping2D (opcional)	Ajuste de c1 si difiere de u1 para concatenación
Concatenate	Combina las salidas de ambas ramas , fusión u1 + c1 (dec concat3)
Conv2D_D3_2 (3×3, 32)	32 filtros 3×3, padding='same', activación ReLU (dec conv3 1)
Conv2D_D3_3 (3×3, 32)	32 filtros 3×3, padding='same', activación ReLU (dec conv3 2)
Conv2D (salida) (1×1, 1)	1 filtro 1×1, activación <b>sigmoide</b> , padding='same' (output segmentation raw)
ZeroPadding2D (opcional)	Si la salida es menor que 50×50, se rellena (final padding)
Reshape	Salida final: (batch_size, 50, 50) (output fire_mask)

Tabla 13\_Descripción Arquitectura Modelo VIII





## 4.7.2 Características principales

- Encoder-Decoder simétrico: Usa una ruta de contracción (encoder) para aprender representaciones profundas y abstractas, y una ruta de expansión (decoder) para reconstruir la resolución espacial.
- Conexiones de salto (skipconnections): Permiten recuperar información espacial perdida durante el downsampling, facilitando la segmentación precisa en los bordes.
- Ajuste espacial robusto: Se implementa lógica condicional para recorte (`Cropping2D`) o padding (`ZeroPadding2D`) que asegura la alineación entre los mapas encoder-decoder, y la salida final a  $50 \times 50$ .
- Salida sigmoïdal con reshape: Se utiliza `sigmoid` para generar una máscara continua entre  $[0, 1]$ , y luego se proyecta a una imagen de salida (`None, 50, 50`).

## 4.7.3 Entrenamiento

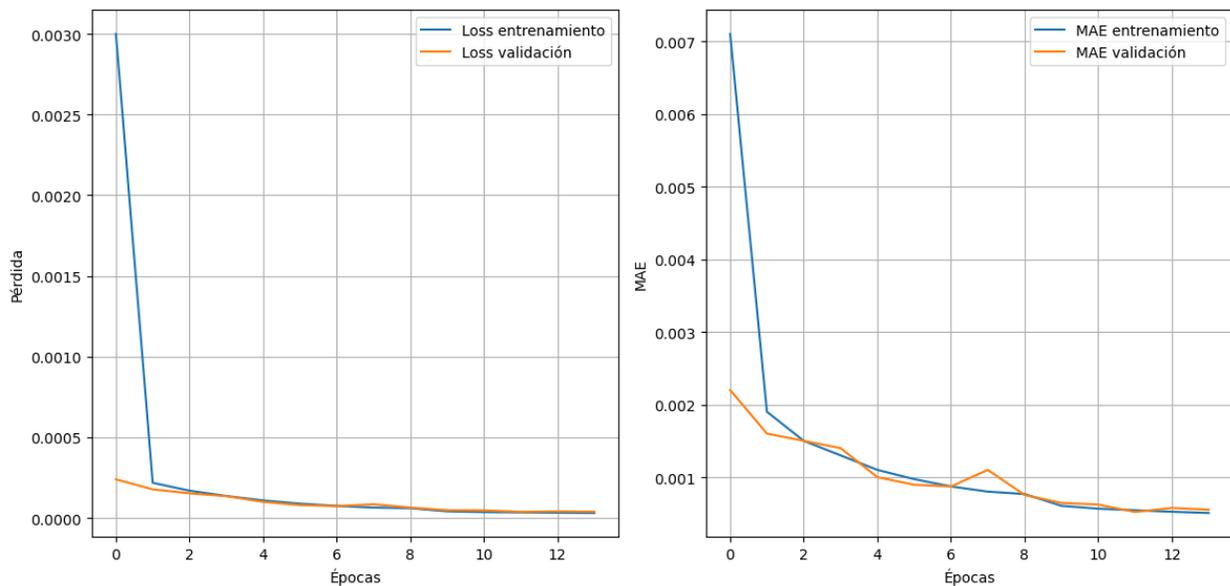


Ilustración 34\_Evolución entrenamiento Modelo VIII

### Evolución del entrenamiento:

Este modelo parte con un `val_mae` ya bajo y mejora rápidamente durante las primeras 5 épocas. Alcanza un mínimo de `val_mae`  $\approx 0.000521$  en la época 11, con una `mae` de entrenamiento de 0.000545. El `ReduceLRonPlateau` se activa dos veces: En la época 9, tras un leve estancamiento y en la época 14, seguido de *early stopping* al no mejorar tras esa bajada.

Curvas suaves y consistentes, con progresiva disminución tanto de `mae` como de `val_mae`, y valores extremadamente bajos en comparación con los modelos anteriores. El aprendizaje es más rápido: solo 14 épocas y tiempos por época mucho menores ( $\approx 750$  s), gracias a una arquitectura más eficiente.

**Posibles errores detectados:**

Ninguno evidente. La curva de validación es estable, sin picos. La diferencia entre  $mae$  y  $val\_mae$  es mínima en todo momento, lo cual es un excelente indicio de buena generalización y bajo riesgo de sobreajuste.

**Conclusión:**

Este modelo es, hasta ahora, el más preciso y eficiente. En conjunto, el Modelo VII logra la mejor precisión con menor coste computacional, lo que lo convierte en un fuerte candidato a modelo final.

**4.7.4 Resultados obtenidos**

Métrica	Valor
Loss (MSE)	0.00011
MAE	0.00107
RMSE	0.01064
R <sup>2</sup> Score	0.99669

Tabla14\_ Resultados Modelo VIII

Este modelo logró un buen desempeño, superando incluso a versiones anteriores como la U-Net base (Modelo IV) y al modelo híbrido VI, con el menor **MAE** registrado (**0.00107**) y un **R<sup>2</sup>** muy alto (**0.99669**).

La arquitectura U-Net mejorada captura tanto información de contexto multi-escala como detalles espaciales, lo que es clave para tareas como la predicción de propagación de incendios, donde las condiciones locales y el entorno global se combinan por las acciones del viento.

El uso explícito de operaciones de alineamiento espacial condicional (recortes o paddings) garantiza estabilidad dimensional incluso con imágenes no perfectamente divisibles por 2, una buena práctica para entornos reales.

La arquitectura es más pesada que los modelos puros convolucionales simples, lo cual ha aumentado el tiempo de entrenamiento y la demanda de memoria, pero en este caso se traduce en beneficios significativos de precisión.

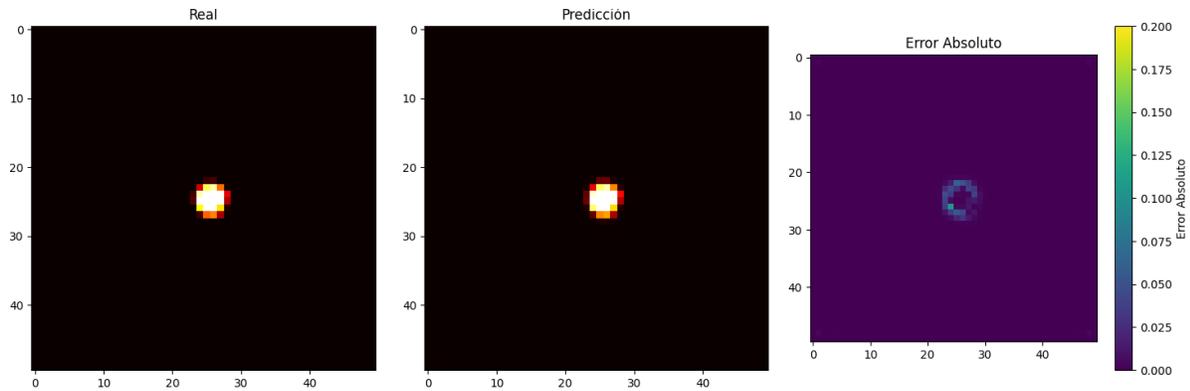


Ilustración 35\_Comparación Visual\_Model VIII

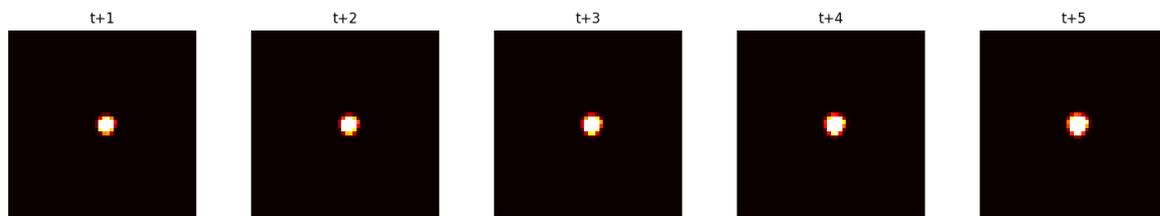


Ilustración 36\_Evolución Autoregresiva\_Model VIII

De nuevo, visualmente los resultados parecen bastante buenos, el error se reduce al contorno de la zona de fuego y se reduce tanto en el interior de esta área como en el exterior. Se continuará el estudio de resultados en el apartado 4.7. para ver el comportamiento tras varios steps temporales.

Continuaremos la línea de investigación con los modelos U-Nets.

## 4.8 Modelo IX - UNet\_Transformer2D

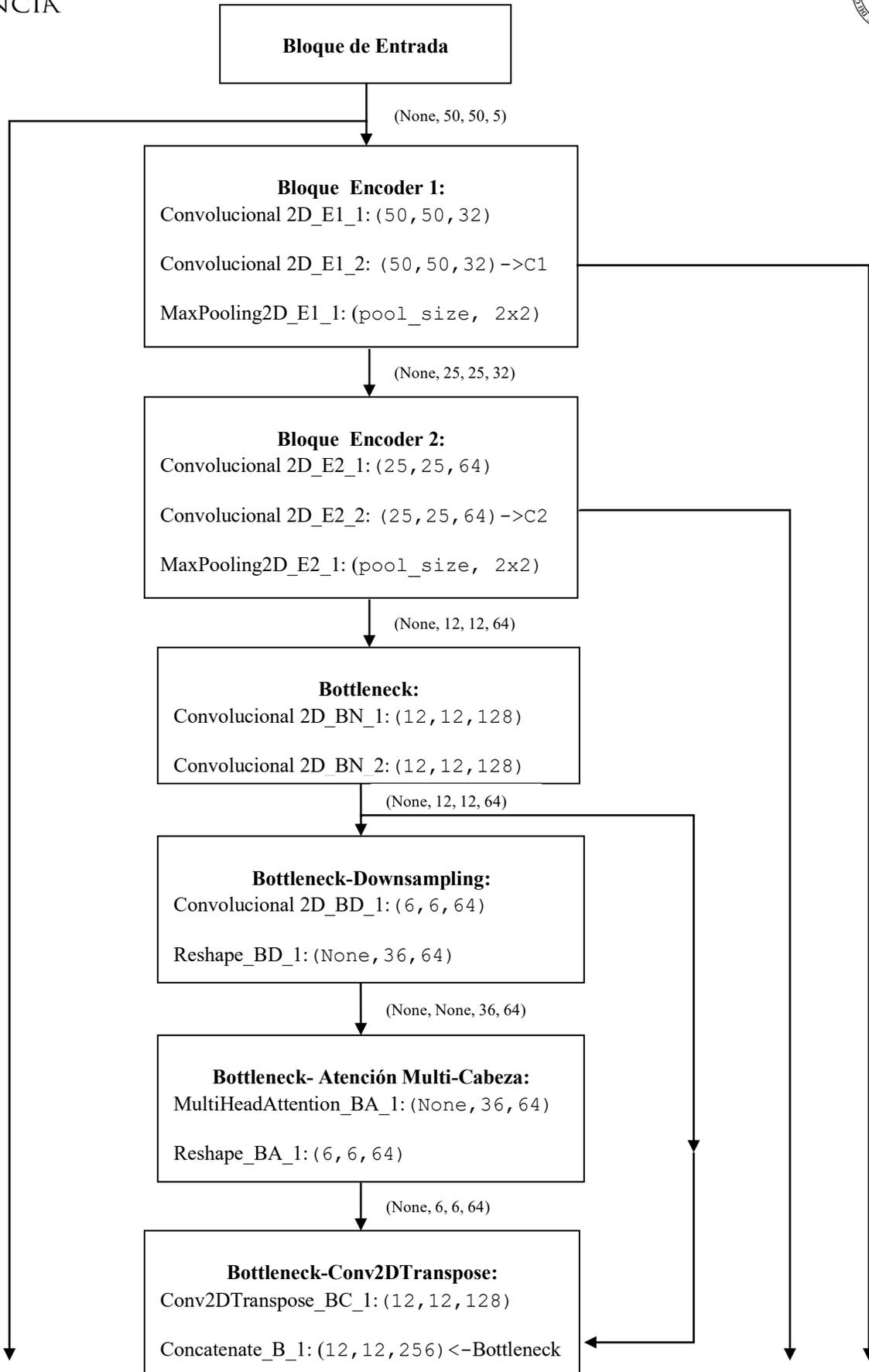
Este modelo representa una implementación avanzada y ligera de una U-Net con atención, diseñada para capturar dependencias espaciales no locales utilizando un bloque tipo **Transformer 2D** insertado en el cuello de la arquitectura. Su objetivo es producir mapas de fuego espacialmente precisos, integrando información contextual profunda con detalles finos a través de un decodificador enriquecido y conexiones residuales.

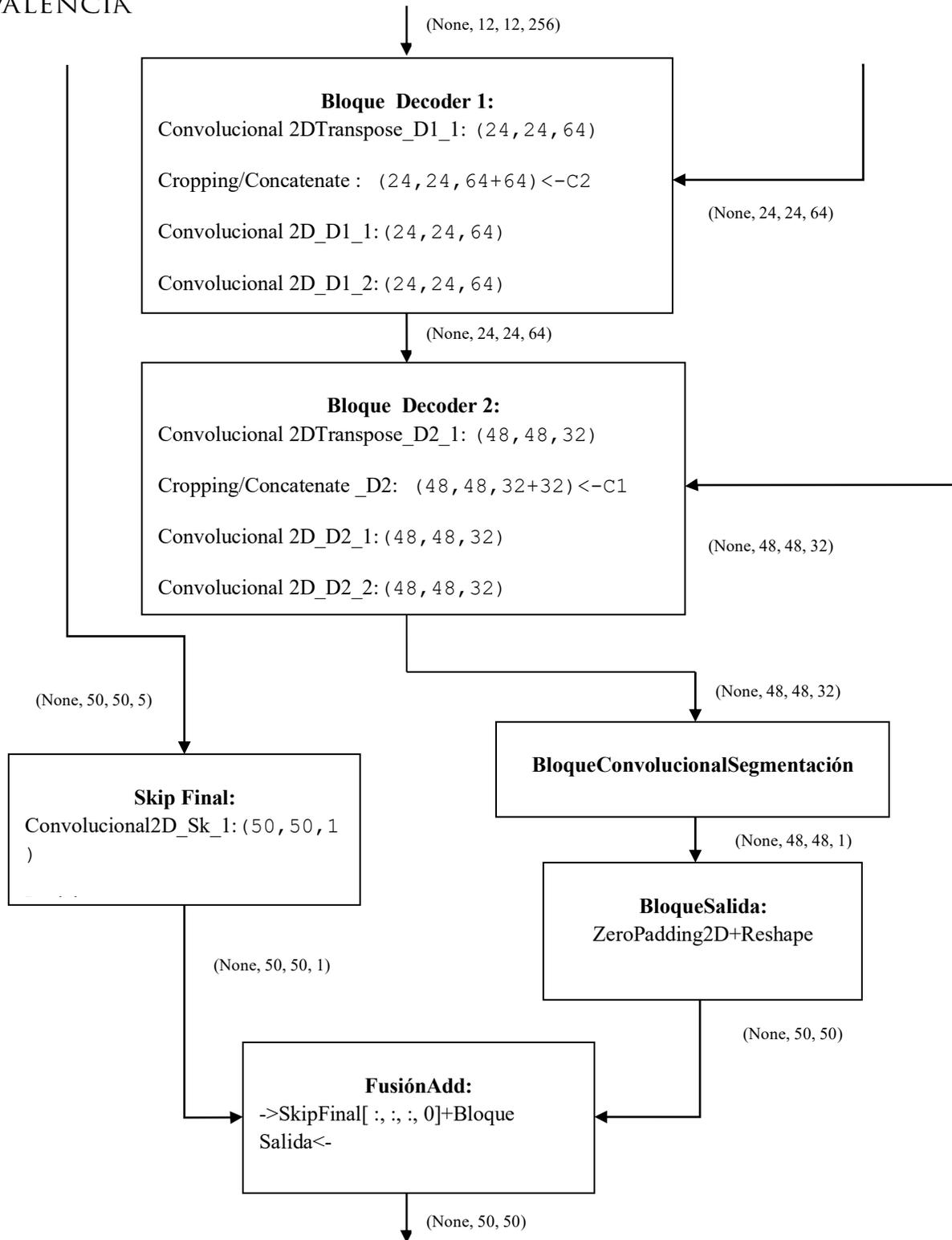
### 4.8.1 Arquitectura del modelo

Capa	Descripción
Input (50, 50, 5)	Forma de entrada: (50, 50, canales)
Conv2D_E1_1 (3×3, 32)	Extrae características espaciales. 32 filtros 3×3, padding='same', activación ReLU (c1)
Conv2D_E1_2 (3×3, 32)	Extrae características espaciales 32 filtros 3×3, padding='same', activación ReLU (c1)
MaxPooling2D_E1_1 (2×2)	Pooling 2×2 (p1) – Reducción espacial a 25×25
Conv2D_E2_1 (3×3, 64)	Extrae características espaciales 64 filtros 3×3, padding='same', activación ReLU (c2)
Conv2D_E2_2 (3×3, 64)	Extrae características espaciales 64 filtros 3×3, padding='same', activación ReLU (c2)
MaxPooling2D_E2_1 (2×2)	Pooling 2×2 (p2) – Reducción espacial a 12×12
Conv2D_BN_1 (3×3, 128)	Extrae características espaciales 128 filtros 3×3, padding='same', activación ReLU (bottleneck)
Conv2D_BN_2 (3×3, 128)	Extrae características espaciales 128 filtros 3×3, padding='same', activación ReLU (bottleneck)
Conv2D_BD_1 (att entrada) (3×3, 64)	Extrae características espaciales y reducción de dimensionalidad. 64 filtros 3×3, strides=2, padding='same', activación ReLU – Reducción a 6×6
Reshape	Reshape a secuencia: (36, 64) para atención
MultiHeadAttention_BA_1 (2)	Atención a diferentes partes de su entrada. 2 cabezas, dimensión clave 16 (att)
Reshape_BA_1	Reconstrucción a (6, 6, 64)
Conv2DTranspose_BC_1	Upsampling (aumento de resolución) en un tensor de entrada. 128 filtros 3×3, strides=2, padding='same', activación ReLU – Upsampling a 12×12
Concatenate_BC_1	Fusión con salida del bottleneck (b + att)
Conv2DTranspose_D1_1	Upsampling (aumento de resolución) en un tensor de entrada. 64 filtros 2×2, strides=2, padding='same' (u2) – Upsampling a 24×24
Cropping2D_D2	Ajuste de c2 para emparejar con u2 si difiere. Opcional
Concatenate_D2	Fusión u2 + c2
Conv2D_D1_1 (3×3, 64)	Extrae características espaciales. 64 filtros 3×3, padding='same', activación ReLU (c4)
Conv2D_D1_2 (3×3, 64)	Extrae características espaciales. 64 filtros 3×3,

	padding='same', activación ReLU (c4)
Conv2DTranspose_D2_1 (2×2, 32)	Upsampling (aumento de resolución) en un tensor de entrada. 32 filtros 2×2, strides=2, padding='same' (u1) – Upsampling a 48×48
Cropping2D_D2	Ajuste de c1 para emparejar con u1 si difiere. Opcional
Concatenate_D2	Fusión u1 + c1
Conv2D_D2_1 (3×3, 32)	Extrae características espaciales. 32 filtros 3×3, padding='same', activación ReLU (c5)
Conv2D_D2_2 (3×3, 32)	Extrae características espaciales. 32 filtros 3×3, padding='same', activación ReLU (c5)
Conv2D_S_1 (salida) (1×1, 1)	1 filtro 1×1, activación sigmoide, padding='same'
ZeroPadding2D_S_1	Padding de 1 píxel para llegar a 50×50 (output)
Reshape_S_1	Salida: (batch_size, 50, 50)
Conv2D_Sk_1 (salto input) (1×1, 1)	Skip final desde inputs: 1 filtro 1×1, padding='same'
Resizing_Sk_1	Redimensionado a 50×50 (skip input)
Add	Suma del output con skip_input[:, :, :, 0] como conexión residual final

Tabla 15\_Descripción Arquitectura Modelo IX





#### 4.8.2 Características principales

- Encoder-Decoder simétrico: La arquitectura utiliza un encoder profundo que reduce la dimensionalidad espacial mientras extrae representaciones abstractas, seguido de un decoder que recupera la resolución original mediante upsampling y convoluciones.

- Bloque Transformer ligero en el cuello: Se incluye un bloque de atención multi-cabeza en la parte más profunda de la red, lo que permite capturar dependencias espaciales amplias con una proyección compacta ( $6 \times 6$ ) para eficiencia.
- Conexiones de salto (skipconnections): Se utilizan conexiones U-Net clásicas entre encoder y decoder, lo que mejora la precisión espacial, especialmente en bordes y estructuras finas del fuego.
- Ajuste espacial robusto: Se incorporan capas condicionales (`Cropping2D`, `ZeroPadding2D`) para asegurar que los tamaños espaciales sean compatibles, incluso si hay discrepancias por divisiones no exactas.
- Salida sigmoideal + fusión residual: La salida se pasa por una activación sigmoide y se combina con una versión redimensionada de la entrada original mediante una suma residual, agregando robustez y facilitando el aprendizaje de correcciones sutiles.

### 4.8.3 Entrenamiento

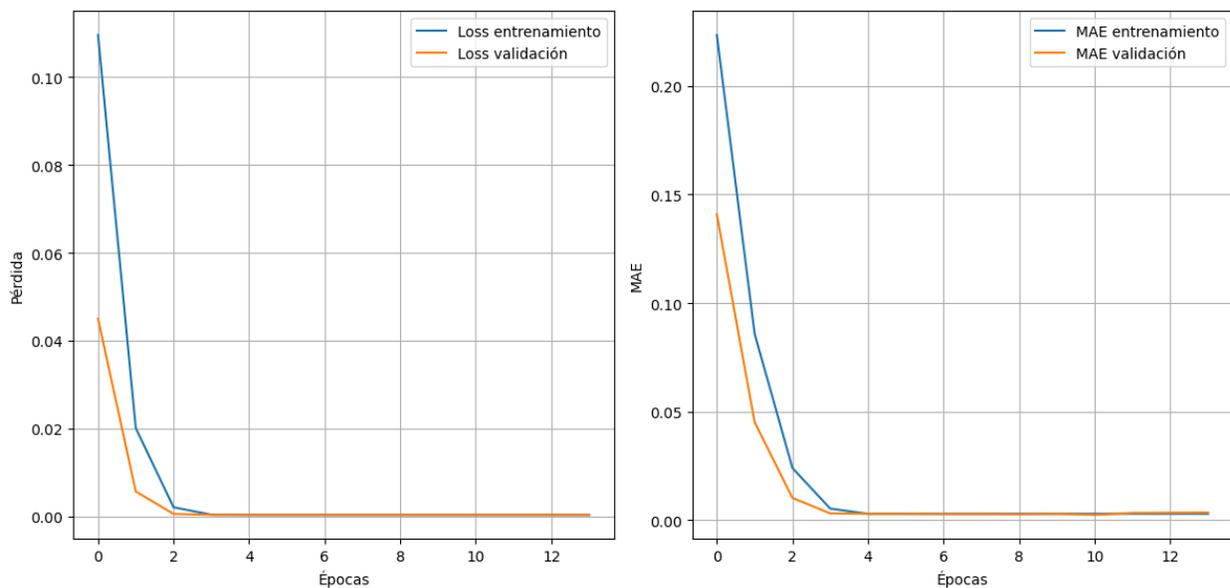


Ilustración 37\_Evolución entrenamiento Modelo IX

#### Evolución del entrenamiento:

Este modelo comienza con un error alto en la época 1 ( $val\_mae = 0.1408$ ), pero mejora drásticamente en las primeras 3 épocas, reduciendo el  $val\_mae$  a 0.0102 (época 3), y luego hasta  $\approx 0.0027$  en la época 9. A partir de ahí, el modelo entra en una meseta prolongada, con cambios mínimos tanto en  $mae$  como en  $val\_mae$ .

Estancamiento claro a partir de la época 5, donde los valores de pérdida ( $loss$ ,  $val\_loss$ ,  $mae$ ,  $val\_mae$ ) prácticamente no cambian, lo que indica que el modelo ha alcanzado su capacidad máxima para esta configuración. Se activan dos veces los mecanismos de `ReduceLROnPlateau`, y se activa *early stopping* en la época 14. El modelo ha mostrado alta eficiencia computacional. La  $val\_mae$  mínimo es:  $\sim 0.0024$  y  $mae$  en entrenamiento muy similar: 0.0029, sugiriendo mínimo sobreajuste.

### Posibles problemas detectados:

Estancamiento temprano desde la época 4. La pérdida se mantiene casi constante, lo que puede indicar un cuello de botella en la capacidad de representación del modelo o que el learning rate no permitió mayor refinamiento pese a los ajustes.

### Conclusión:

El Modelo XIX es eficiente y confiable, muestra una rápida mejora inicial y buena generalización temprana, pero se estabiliza demasiado pronto, sin alcanzar la precisión de modelos como el VIII.

#### 4.8.4 Resultados obtenidos

Métrica	Valor
Loss (MSE)	0.00010
MAE	0.00258
RMSE	0.00984
R <sup>2</sup> Score	0.99717

Tabla16\_Resultados Modelo IX

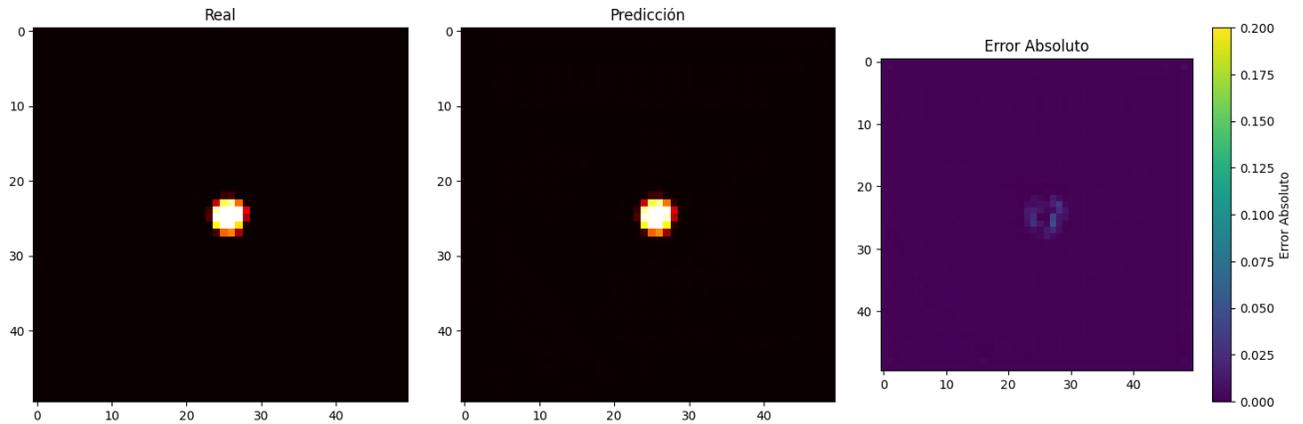
Este modelo logró un desempeño excelente, con un **MAE de 0.00258** y un **R<sup>2</sup> de 0.99717**, posicionándose como uno de los modelos más precisos hasta la fecha.

La combinación de convoluciones profundas con un bloque de atención Transformer 2D en el cuello ofrece una mejora sustancial en la captación de contexto global sin perder detalle espacial, un aspecto crucial para la propagación de incendios.

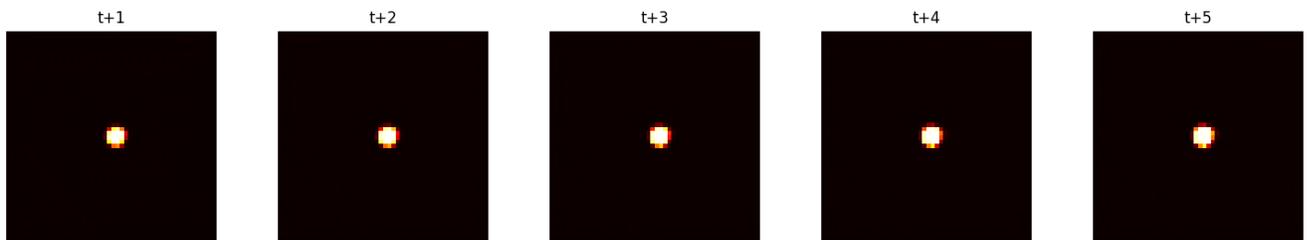
El uso explícito de ajuste espacial condicional (recorte o padding) garantiza que las salidas mantengan integridad dimensional sin errores de forma, lo cual es fundamental para inferencias consistentes.

La suma residual con la entrada original ayuda a estabilizar el aprendizaje y refinar predicciones, una técnica muy útil cuando el estado inicial del fuego es informativo.

Aunque más ligero que un Transformer completo, este modelo introduce cierta complejidad computacional adicional respecto a modelos puramente convolucionales, pero lo compensa con mejoras notables en precisión y capacidad de generalización.



**Ilustración 38\_Comparación Visual\_Model IX**



**Ilustración 39\_Evolució Autoregresiva\_Model IX**

Se observa una reducción del error considerable, mejorando bastante los resultados del anterior modelo, ya de por si buenos.

## 4.9 Modelo X – Unet\_ Attention

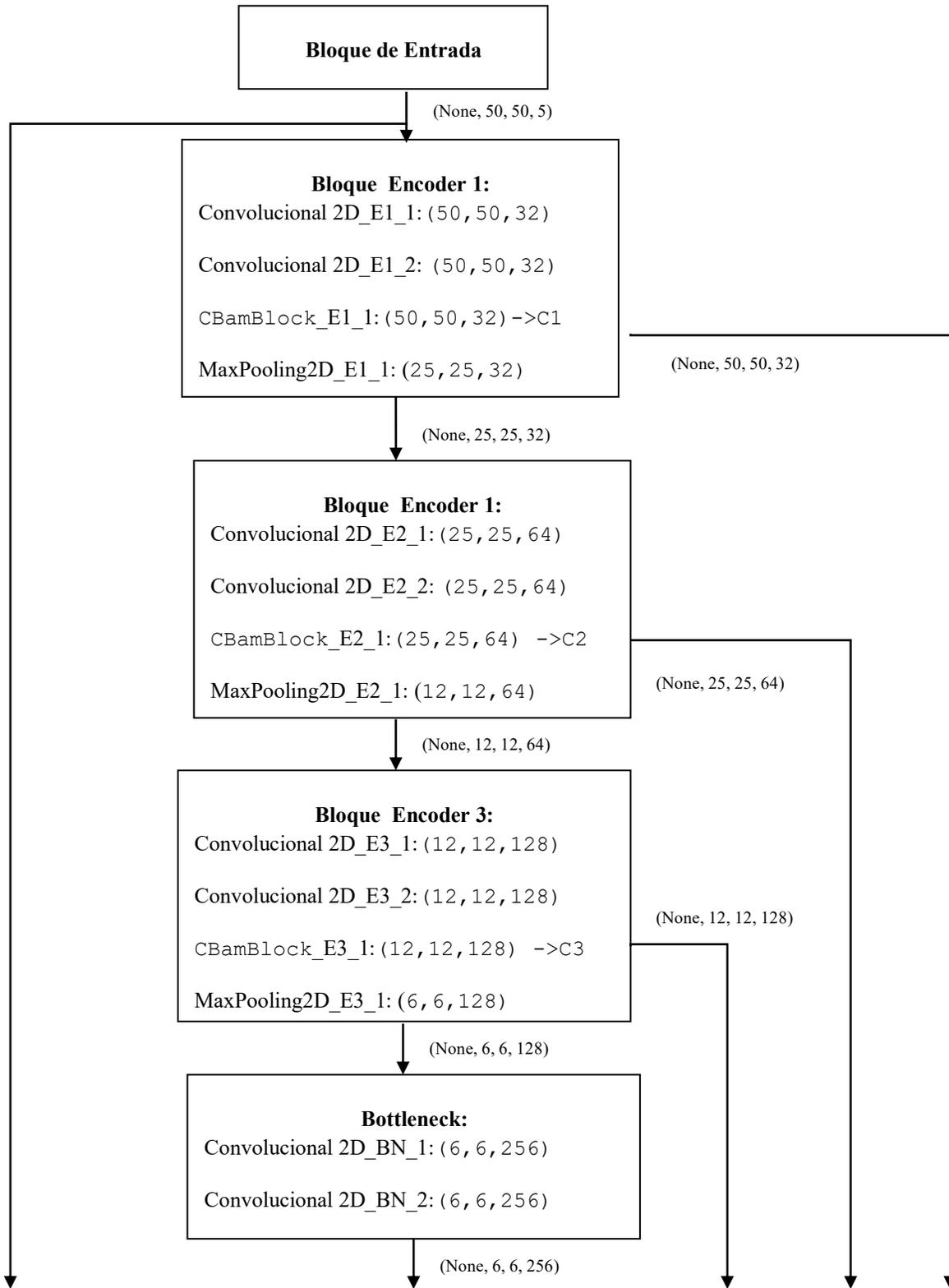
Este modelo es una versión avanzada de la arquitectura U-Net, reforzada con módulos de atención espacial y de canal tipo **CBAM (Convolutional Block Attention Module)**. Su diseño está orientado a capturar tanto la estructura global como las regiones más relevantes del mapa de entrada, ideal para tareas donde el contexto espacial y la discriminación de regiones activas son críticas, como la predicción de la propagación del fuego.

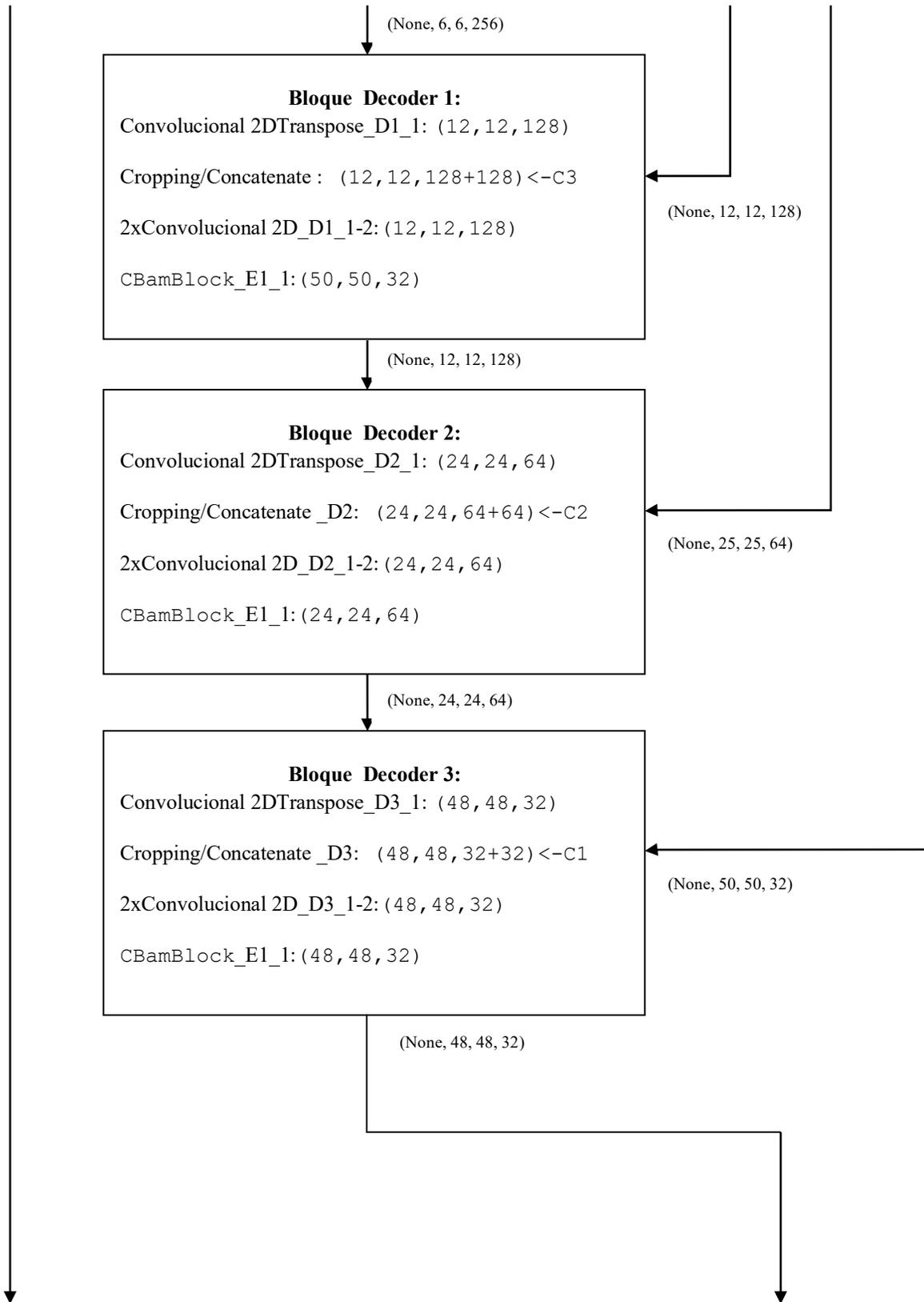
### 4.9.1 Arquitectura del modelo

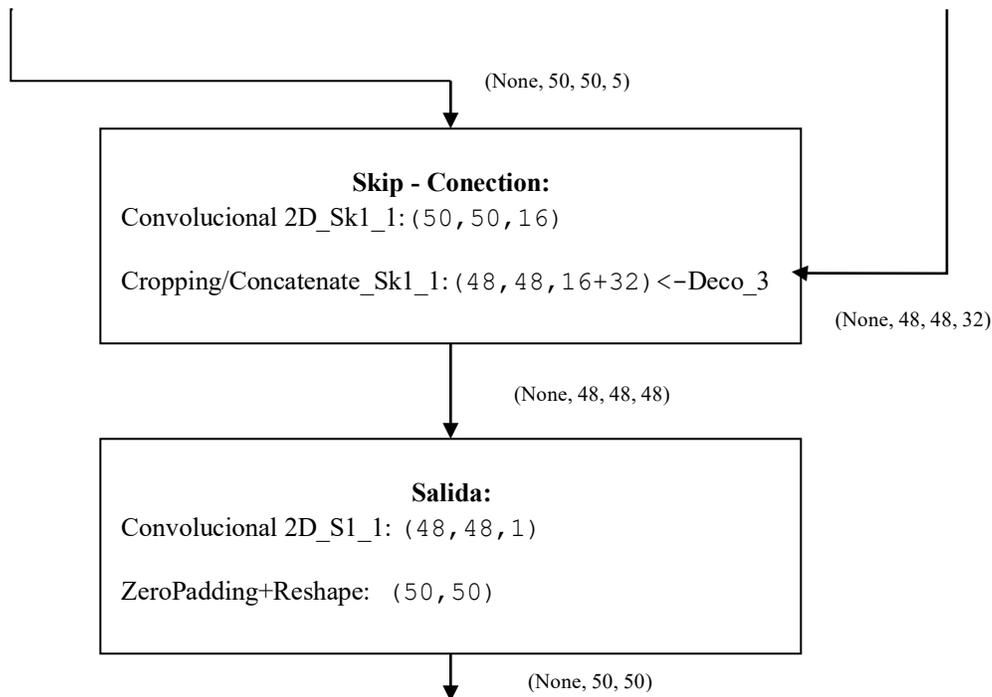
Capa	Descripción
Input (50, 50, 5)	Forma de entrada: (50, 50, canales)
Conv2D_E1_1 (3×3, 32)	Extrae características espaciales. 32 filtros 3×3, activación ReLU, padding 'same' (Encoder Block 1)
Conv2D_E1_2 (3×3, 32)	Extrae características espaciales. 32 filtros 3×3, activación ReLU, padding 'same' (Encoder Block 1)
CBAMBlock_E1_1	Atención espacial y de canal (Encoder Block 1)
MaxPooling2D_E1_1	Reducción espacial a (25, 25)(Encoder Block 1)
Conv2D_E2_1 (3×3, 64)	Extrae características espaciales. 64 filtros 3×3, activación ReLU, padding 'same' (Encoder Block 2)
Conv2D_E2_2 (3×3, 64)	Extrae características espaciales. 64 filtros 3×3, activación ReLU, padding 'same' (Encoder Block 2)
CBAMBlock_E2_1	Atención CBAM (Encoder Block 2)
MaxPooling2D_E2_1	Reducción espacial a (12, 12)(Encoder Block 2)
Conv2D_E3_1 (3×3, 128)	Extrae características espaciales. 128 filtros 3×3, activación ReLU, padding 'same' (Encoder Block 3)
Conv2D_E3_2 (3×3, 128)	Extrae características espaciales. 128 filtros 3×3, activación ReLU, padding 'same' (Encoder Block 3)
CBAMBlock_E3_1	Atención CBAM (Encoder Block 3)
MaxPooling2D_E3_1	Reducción espacial a (6, 6)(Encoder Block 3)
Conv2D_BN_1 (3×3, 256)	Extrae características espaciales. 256 filtros 3×3, activación ReLU, padding 'same' (Bottleneck)
Conv2D_BN_2 (3×3, 256)	Extrae características espaciales. 256 filtros 3×3, activación ReLU, padding 'same' (Bottleneck)
Conv2DTranspose_D1_1 (2×2, 64)	Upsampling (aumento de resolución) en un tensor de entrada. 128 filtros 2×2, stride 2, upsampling a (12, 12) (Decoder Block 3)
Cropping2D_D1_1	Ajuste espacial con skip connection (Decoder Block 3)
Concatenate_D1_1	Unión con salida del encoder (s3) (Decoder Block 3)
Conv2D_D1_1 (3×3, 256)	Extrae características espaciales. 128 filtros 3×3, activación ReLU(Decoder Block 3)
Conv2D_D1_2 (3×3, 256)	Extrae características espaciales. 128 filtros 3×3, activación ReLU(Decoder Block 3)
CBAMBlock_D1_1	Atención CBAM (Decoder Block 3)
Conv2DTranspose_D2_1 (2×2, 64)	Upsampling (aumento de resolución) en un tensor de entrada. 64 filtros 2×2, stride 2, upsampling a (24, 24) (Decoder

	Block 2)
Cropping2D_D2_1	Ajusteespacial con skip connection (Decoder Block 2). Opcional
Concatenate_D2_1	Unión con salida del encoder (s2) (Decoder Block 2)
Conv2D_D2_2 (3×3, 128)	Extrae características espaciales. 64 filtros 3×3, activación ReLU(Decoder Block 2)
Conv2D_D2_3 (3×3, 128)	Extrae características espaciales. 64 filtros 3×3, activación ReLU(Decoder Block 2)
CBAMBlock_D2_1	Atención CBAM (Decoder Block 2)
Conv2DTranspose_D3_1 (2×2, 32)	Upsampling (aumento de resolución) en un tensor de entrada. 32 filtros 2×2, stride 2, upsampling a (48, 48) (Decoder Block 1)
Cropping2D_D3_1	Ajusteespacial con skip connection (Decoder Block 1). Si aplica
Concatenate_D3_1	Unión con salida del encoder (s1) (Decoder Block 1)
Conv2D_D3_2 (3×3, 32)	Extrae características espaciales. 32 filtros 3×3, activación ReLU(Decoder Block 1)
Conv2D_D3_3 (3×3, 32)	Extrae características espaciales. 32 filtros 3×3, activación ReLU(Decoder Block 1)
CBAMBlock_D3_1	Atención CBAM (Decoder Block 1)
Conv2D_Sk_1 (3×3, 16)	Extrae características espaciales de la entrada 16 filtros 3×3, activación ReLU(Skip desde la entrada)
Cropping2D_Sk_1	Recorte de 1 píxel por lado → (48, 48)(Skip desde la entrada)
Concatenate_Sk_1	Fusión entre decoder final y rama residual (Post-decoder)
Conv2D_S_1 (3×3, 1)	Salida. 1 filtro 3×3, activación sigmoide, salida (48, 48, 1)(Output)
ZeroPadding2D_S_1	Padding simétrico de 1 píxel → salida final (50, 50, 1)(Output)

Tabla 17\_ Descripción Arquitectura Modelo X







#### 4.9.2 Características principales

- Encoder-Decoder profundo con simetría U-Net: Consta de una ruta de contracción para extraer abstracciones multiescala y una ruta de expansión para reconstruir la resolución espacial, favoreciendo la localización precisa del fuego.
- Atención CBAM en múltiples niveles: Cada bloque encoder y decoder aplica el módulo CBAM, que combina atención espacial y de canales para focalizar la atención de la red en las regiones más relevantes, mejorando la discriminación entre zonas activas e inactivas.
- Bloques auxiliares compactos: Usa bloques auxiliares `conv_block` reutilizables con dos capas convolucionales ReLU, y `decoder_block` que incluye upsampling, alineamiento condicional, skipconnections y atención.
- Ajuste espacial robusto: Incluye operaciones condicionales `Cropping2D` en cada etapa del decoder para garantizar que las dimensiones entre las ramas encoder-decoder coincidan, especialmente en resoluciones impares.

- Skip final desde la entrada original: Se incorpora un canal adicional derivado de la entrada original, tras convolución y recorte, que es fusionado con la etapa final del decoder para reforzar los detalles originales.
- Salida sigmoïdal final con padding: Genera una máscara continua  $[0, 1]$ , luego se reescala a una forma final de salida  $(50, 50)$  mediante `ZeroPadding2D`, asegurando consistencia con las dimensiones originales.

### 4.9.3 Entrenamiento

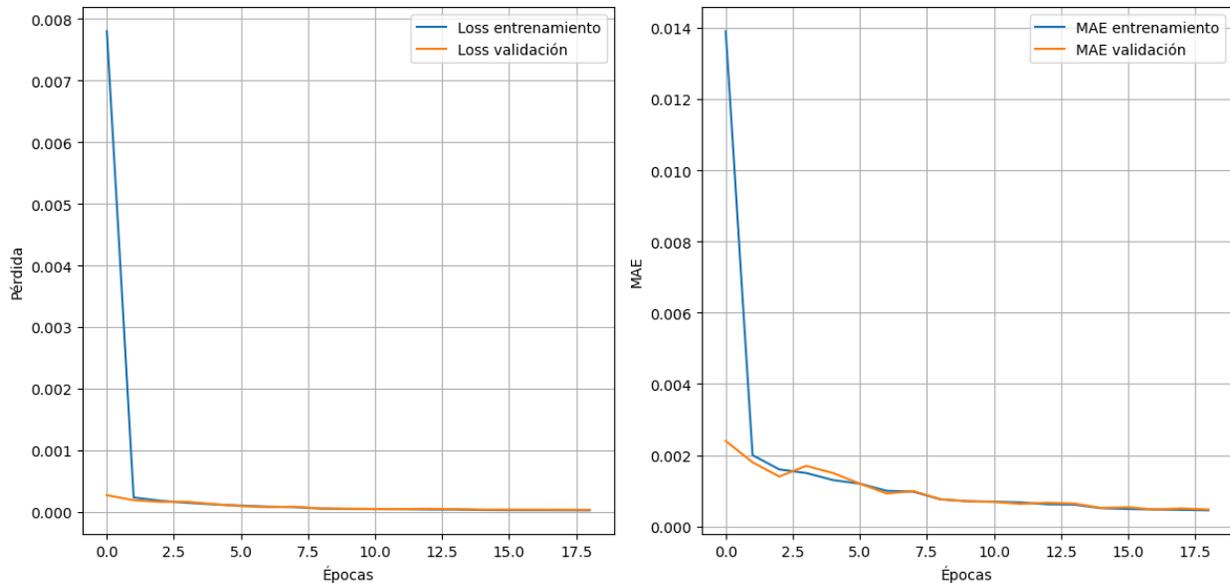


Ilustración 40\_Evolución entrenamiento Modelo X

#### Evolución del entrenamiento:

Este modelo muestra un rendimiento altamente eficiente y progresivo desde la primera época. Comienza con un `val_mae` muy bajo y lo reduce rápidamente alcanzando un mínimo final de `val_mae`  $\approx 4.74e-4$  (época 18), uno de los mejores logrados hasta ahora.

El Modelo sigue mejorando de forma constante hasta el final del entrenamiento, sin señales prematuras de estancamiento. Solo al final, las mejoras se hacen menores, y se activa *early stopping* tras varios ajustes de *learning rate*.

Excelente consistencia entre entrenamiento y validación, con cero sobreajuste. Tiempo por época notablemente más alto que los dos modelos previos. Reducción efectiva del error con cada cambio de tasa de aprendizaje (*ReduceLRonPlateau*)

#### Posibles problemas detectados:

El rendimiento mejora únicamente tras múltiples reducciones del learning rate; esto podría indicar una necesidad de mejor ajuste inicial.

Margen de mejora residual pequeño: el modelo ya alcanza un rendimiento muy bajo en error, por lo que mejoras futuras podrían ser marginales a menos que se modifique la arquitectura o los datos.

**Conclusión:**

El Modelo X es estable, preciso y altamente generalizable, aunque con mayor costo computacional por época. Se trata de un modelo exitoso y sólido, ideal como referencia o base para mejoras adicionales.

**4.9.4 Resultados obtenidos**

Métrica	Valor
Loss (MSE)	0.00005
MAE	0.00061
RMSE	0.00694
R <sup>2</sup> Score	0.99859

Tabla18\_ Resultados ModeloX

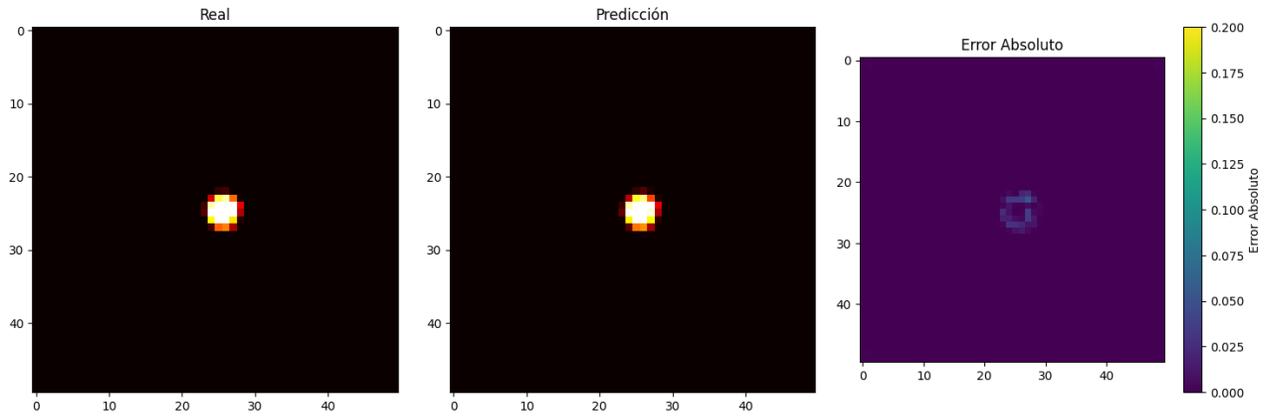
Este modelo obtuvo el **mejor rendimiento global** hasta ahora, con el **MAE más bajo (0.00061)** y un **R<sup>2</sup> de 0.99859**, lo que lo convierte en el más preciso de todos los modelos evaluados.

La inclusión del módulo CBAM en todos los bloques encoder y decoder potencia significativamente la capacidad de la red para enfocar su atención en zonas clave del mapa, permitiendo una segmentación extremadamente precisa.

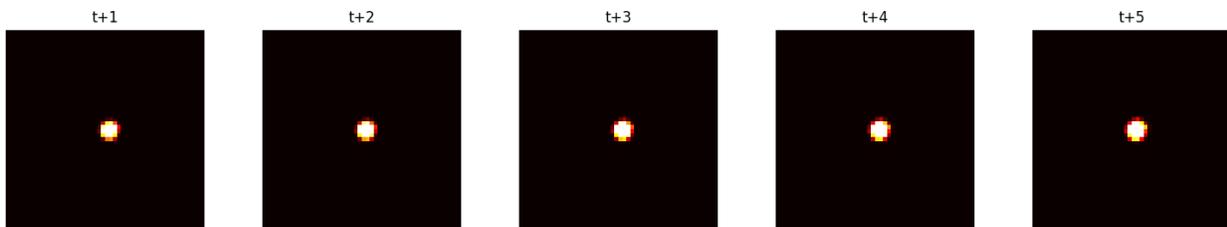
La arquitectura es profunda pero bien balanceada, reutilizando bloques internos y manteniendo estabilidad espacial en cada etapa, lo que facilita el entrenamiento y reduce errores por desalineamiento.

El uso de una conexión final desde la entrada original, combinada con la salida del decoder, actúa como una forma de refinamiento residual, particularmente útil cuando el estado inicial del fuego ya contiene una señal valiosa.

El uso repetido de atención CBAM implica un mayor costo computacional respecto a modelos CNN puros, pero la ganancia en precisión lo justifica ampliamente en tareas sensibles como la predicción de incendios.



**Ilustración 41\_Comparación Visual\_Model X**



**Ilustración 42\_Evolució Autoregresiva\_Model X**

Al igual que en el modelo previo, se observa una reducción general del error, siguiendo la mejora progresiva de los modelos.

## 5 Resultados

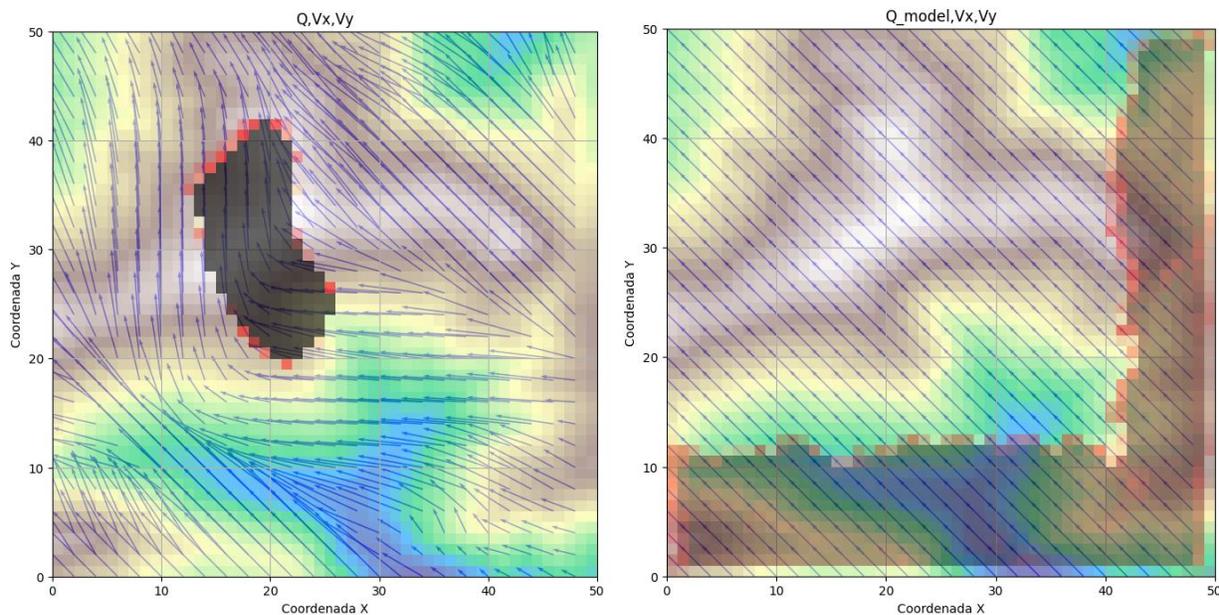
En este apartado se van a analizar los diferentes modelos comparándolos con el resultado del simulador tras 25 y/o 95 pasos. De esta forma se podrá ver de una forma más gráfica el error y las desviaciones de los modelos con respecto a lo que sería el simulador.

Se debe tener en cuenta que las predicciones de cada paso se basan en la predicción previa del modelo, y que por tanto el error no se incrementa de una forma lineal, si no que dependerá de la complejidad del modelo.

Para la comparativa se ha optado por un escenario realista y no un entorno lineal o hipotético, de forma que las acciones del viento, vegetación y orografía generan una mayor complejidad en la predicción. Se ha usado de base el escenario “PuntaCarnero” con un viento genérico de  $-45^\circ$  Noroeste y 14.2 m/s de velocidad.

En los mapas se puede observar el estado del Incendio donde: Negro= Zona completamente Calcinada, Rojo=Foco activo, Amarillo=Foco muy activo.

### 5.1 Modelo I – CNN estática (Conv2D)



**Ilustración 43\_Paso 95 Punta Carnero Angulo  $-45^\circ$  vel 14.2m/s\_n=50\_Model I**

El modelo no es capaz de reproducir el fuego provocado en la celda central. Sin embargo, genera fuego espontaneo en el perímetro. Esto ya se observaba en la evolución autorregresiva.

## 5.2 Modelo II - Improved\_Sequential\_Conv2D

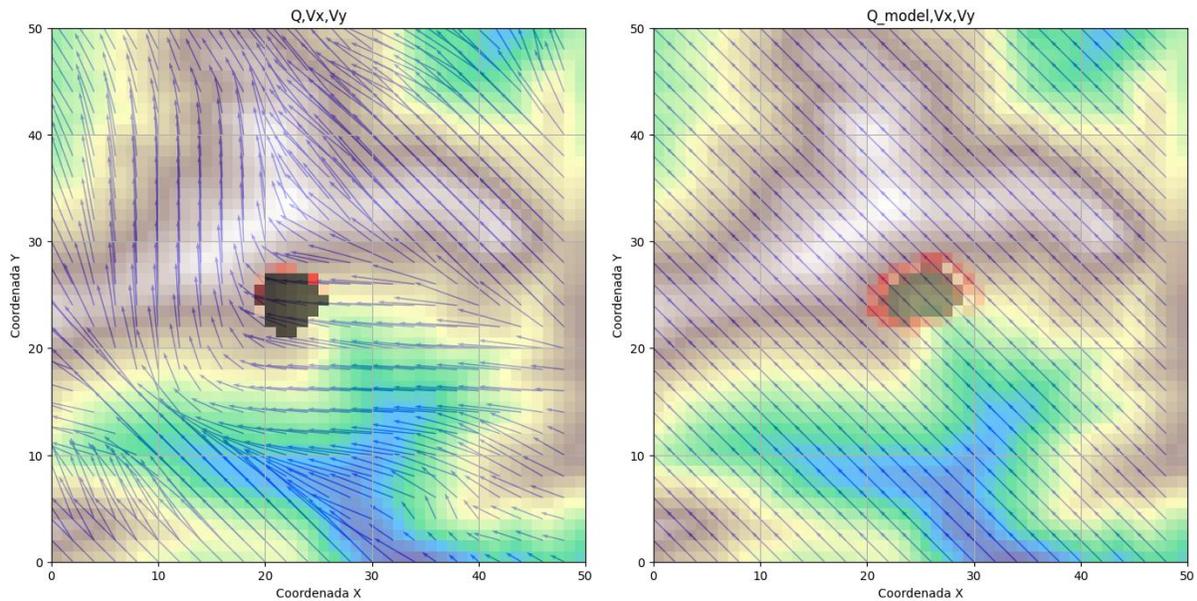


Ilustración 44\_Paso 25 Punta Carnero Angulo -45° vel 14.2m/s\_n=50\_Model II

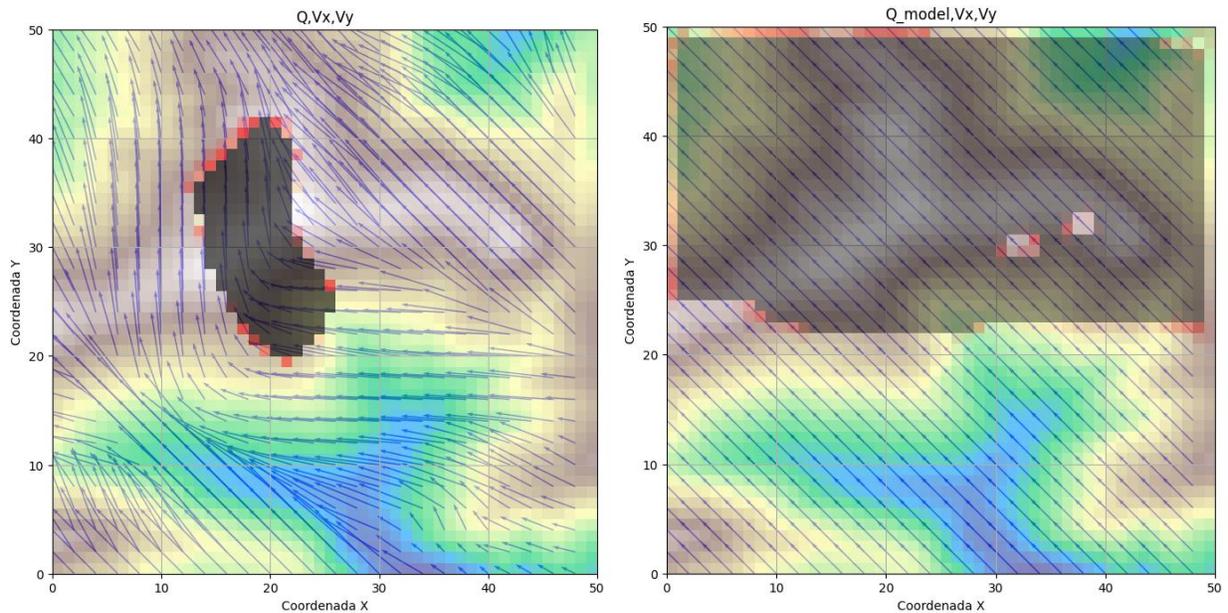


Ilustración 45\_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s\_n=50\_Model II

El modelo parece que en los estados iniciales logra una expansión o propagación del fuego, sin embargo, no logra captar bien los márgenes o contornos de propagación iniciando el fuego en varias capas de celdas a la vez lo que genera una aceleración de la propagación no acorde con el modelo “teórico”.

### 5.3 Modelo III - ClassBased

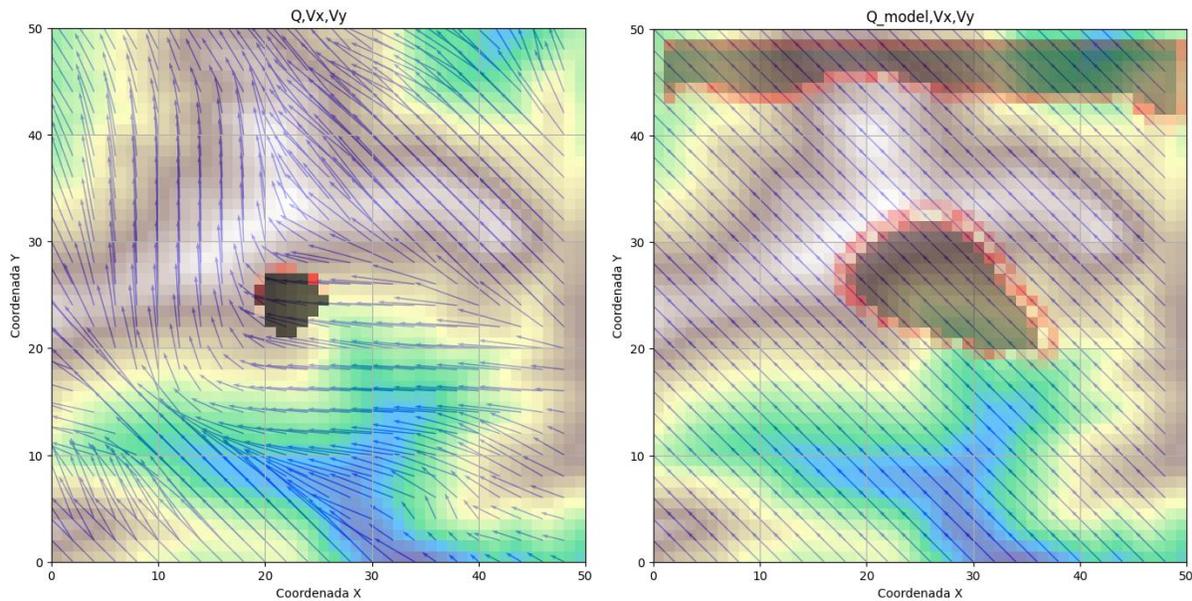


Ilustración 46\_Paso 25 Punta Carnero Angulo -45° vel 14.2m/s\_n=50\_Model III

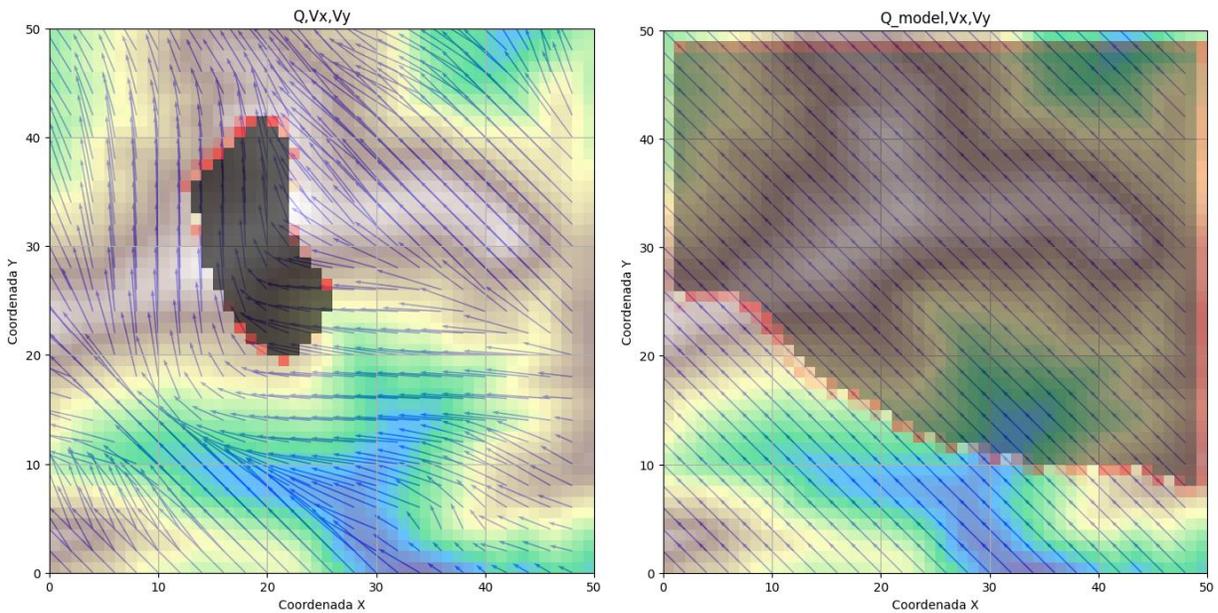


Ilustración 47\_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s\_n=50\_Model III

Se observa que hay inicialización y desarrollo del fuego, sin embargo el modelo no parece desarrollarse de acuerdo a los principios físicos dados por las capas de viento y altura. Además tenemos una inicialización de fuego espontánea en toda la parte superior de nuestro escenario.

Esto puede explicarse por varios factores:

- El modelo puede estar sobre-cualificando el canal clave (mayor profundidad y dropout) mientras que el resto de los canales reciben un tratamiento más superficial.
- La separación de ramas y posterior combinación puede introducir cierta fricción o pérdida de sinergias entre variables que interaccionan espacialmente de forma conjunta.

## 5.4 Modelo IV - Attn\_Fusion\_Dilated

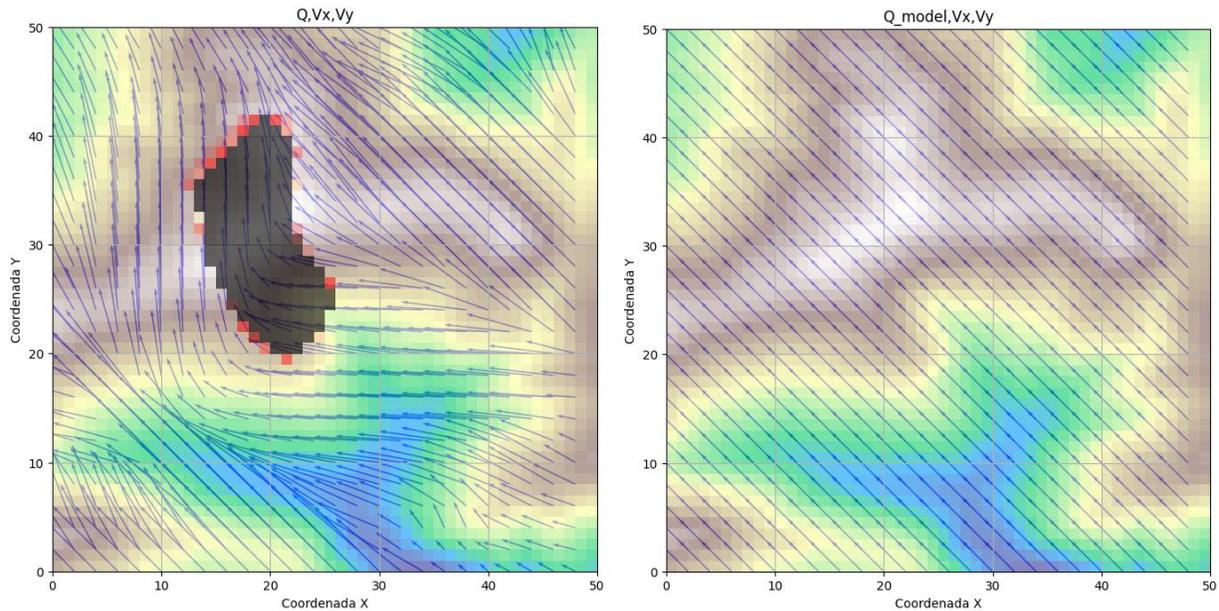


Ilustración 48\_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s\_n=50\_Model IV

El modelo no es capaz de hacer inicialización del fuego pese a tener las mejores métricas hasta ahora.

El modelo no responde bien a una única celda activada, parece que logra resultados a partir de fases posteriores o fuegos mas extendidos.

### 5.5 Modelo V - GlobalFlattenDense

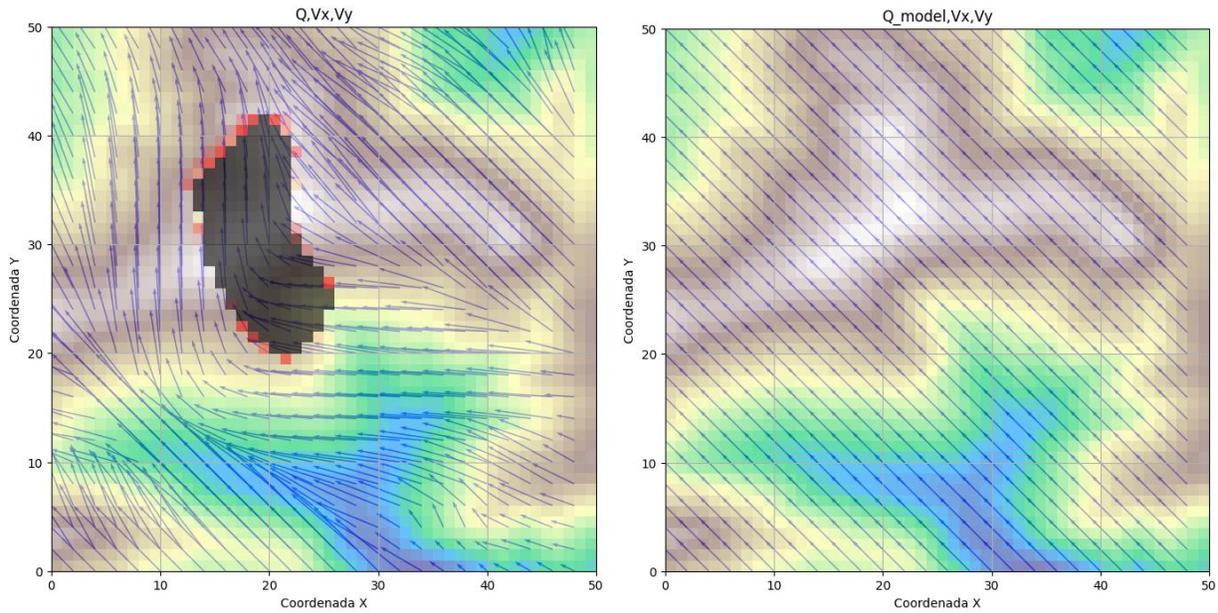


Ilustración 49\_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s\_n=50\_Model V

No es capaz de hacer inicialización del fuego

### 5.6 Modelo VI - Local\_Global\_Fusion

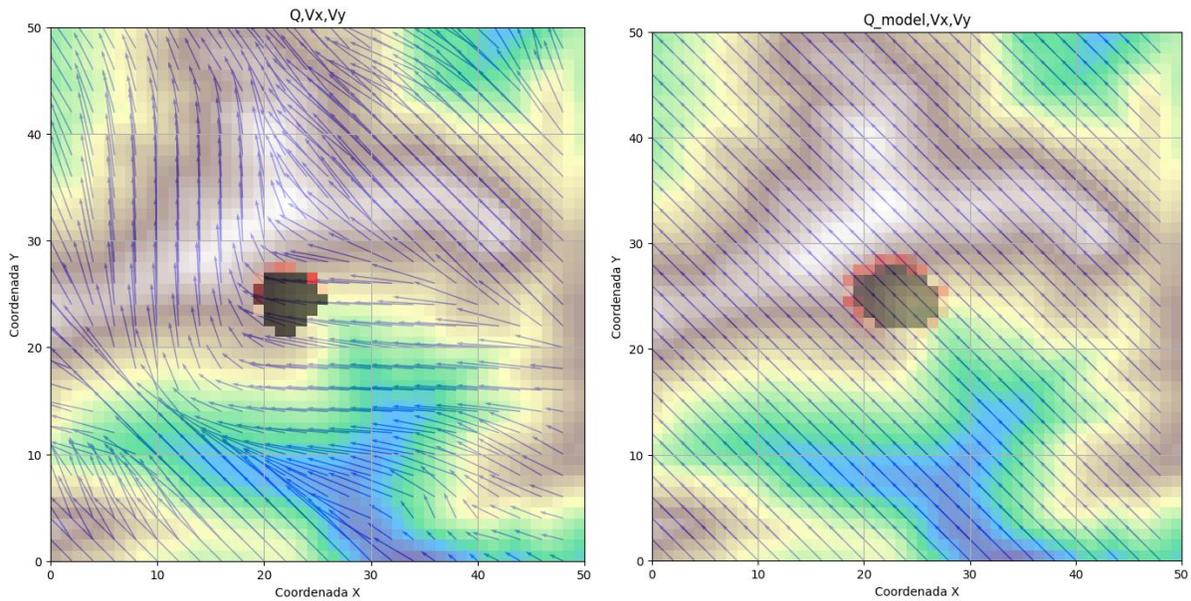
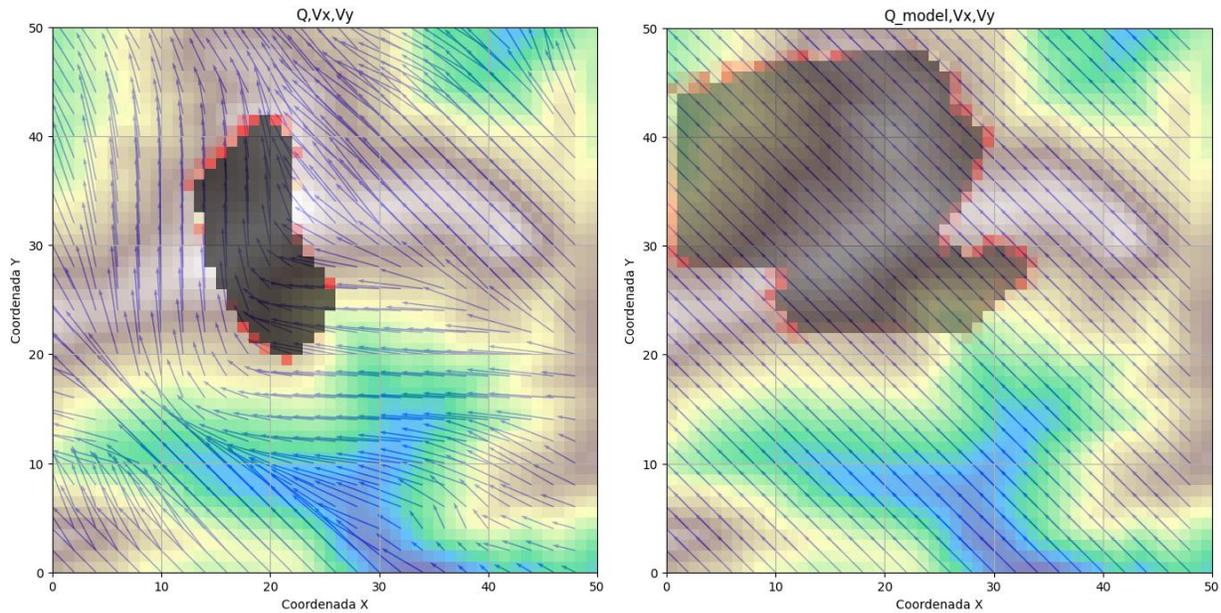


Ilustración 50\_Paso 25 Punta Carnero Angulo -45° vel 14.2m/s\_n=50\_Model VI

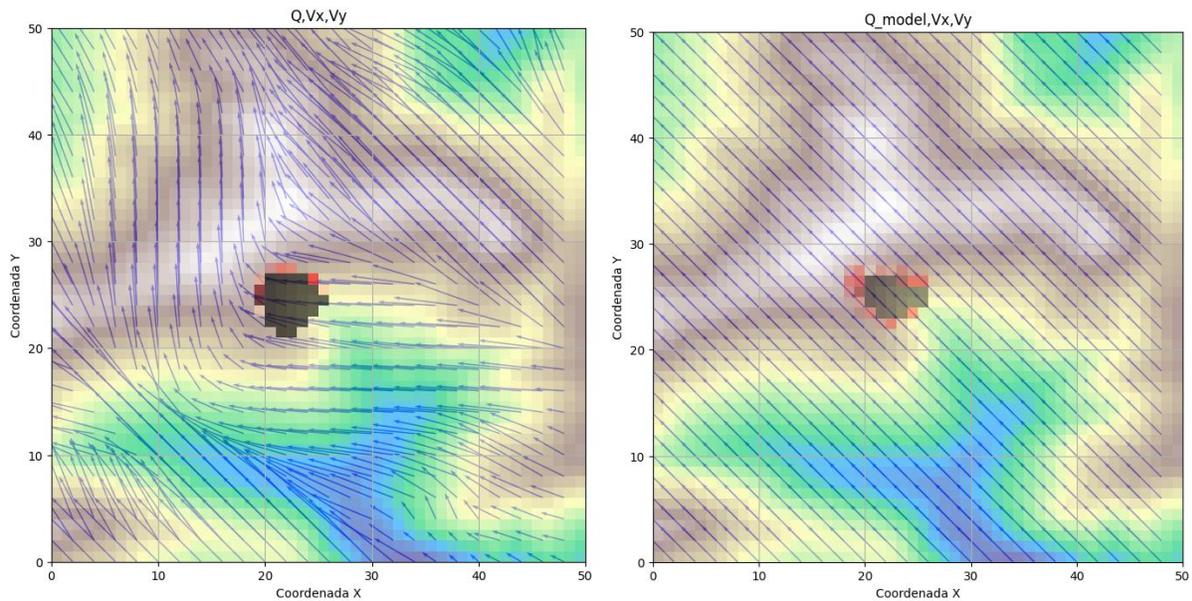


**Ilustración 51\_Paso 95 Punta Carnero Angulo  $-45^\circ$  vel  $14.2\text{m/s}$   $n=50$ \_Model VI**

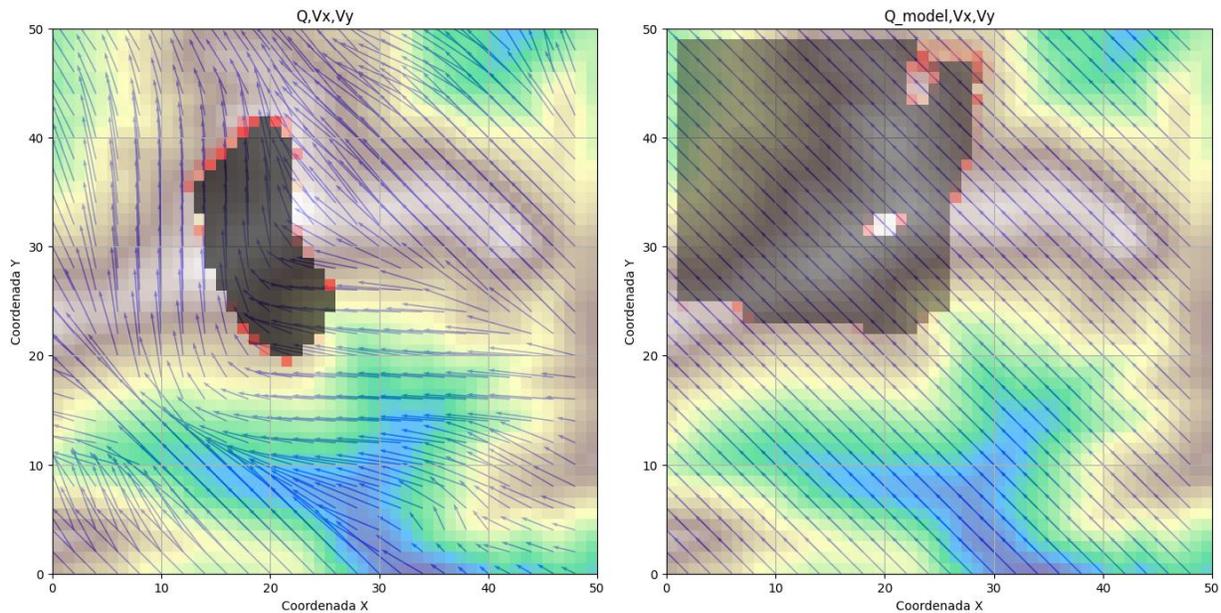
El modelo da unos resultados coherentes la propagación del incendio se dirige al Noroeste, coincidiendo la dirección del viento, además no genera activaciones espontaneas y mantiene el interior del área Quemada al mismo nivel.

No obstante el área quemada es bastante más amplia que el teórico. Parece ser que la pendiente negativa no logra ralentizar la propagación del fuego

## 5.7 Modelo VIII - UNet\_Segmentation\_Improved



**Ilustración 52\_Paso 25 Punta Carnero Angulo -45° vel 14.2m/s\_n=50\_Model VIII**



**Ilustración 53\_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s\_n=50\_Model VIII**

Existe inicialización de fuego y propagación en la dirección correcta, no obstante, no se observa desaceleración de la propagación debida a la pendiente negativa. También se observa una zona central que ha perdido su estado de “quemado”.

## 5.8 Modelo IX – Unet\_Transformer2D

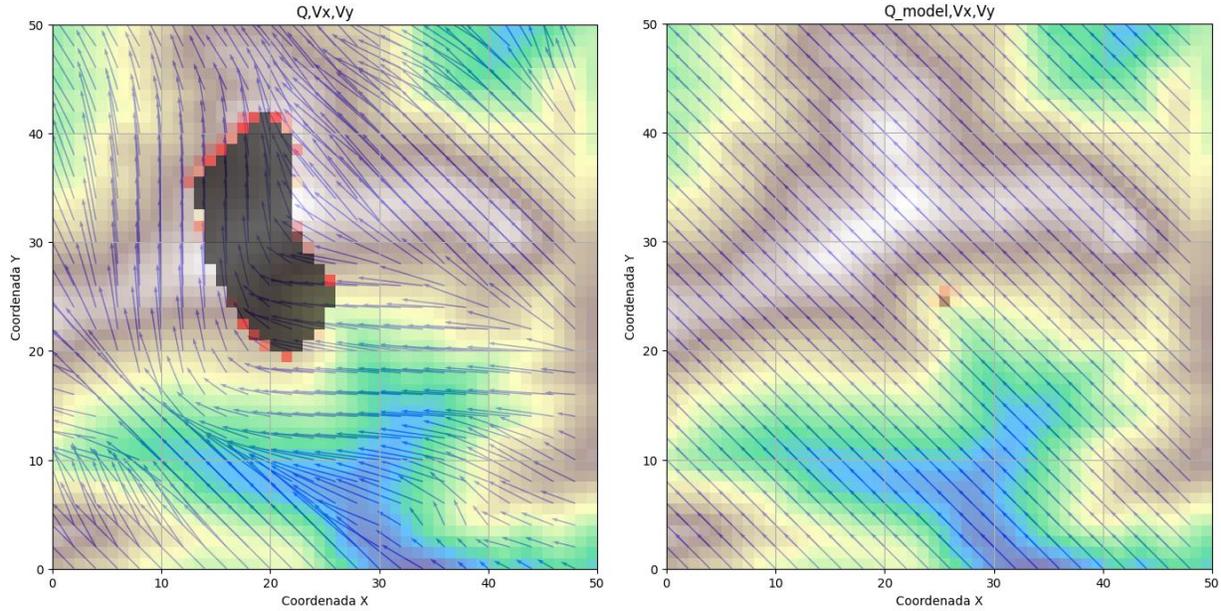


Ilustración 54\_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s\_n=50\_Model IX

El modelo logra iniciar el fuego, pero no logra propagación. Se comprueba con otros escenarios y parece funcionar de forma adecuada, no obstante, en el presente parece incapaz de avanzar.

## 5.9 Modelo X - UNet\_Attention

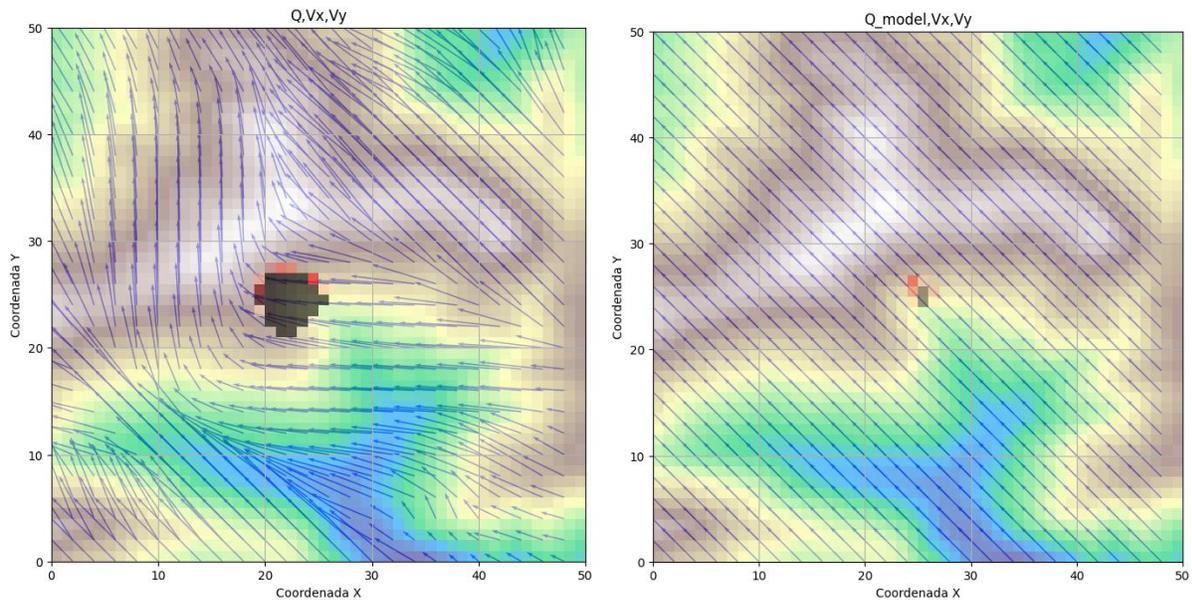


Ilustración 55\_Paso 25 Punta Carnero Angulo -45° vel 14.2m/s\_n=50\_Model X

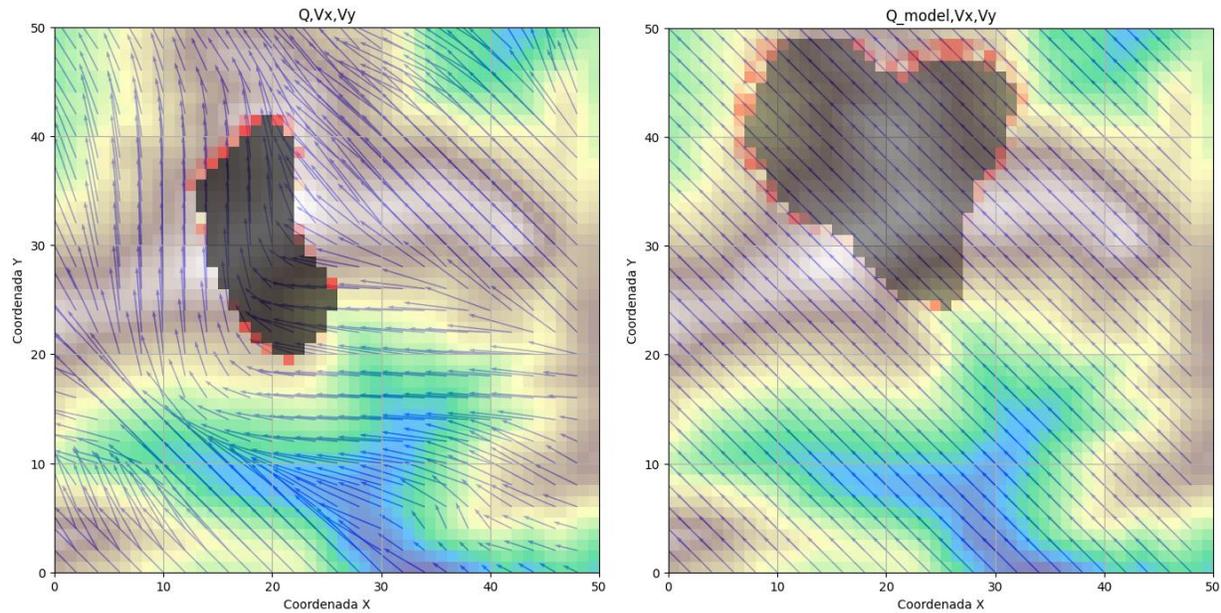


Ilustración 56\_Paso 95 Punta Carnero Angulo -45° vel 14.2m/s\_n=50\_Model X

El modelo da unos resultados coherentes:

- La propagación inicial del incendio se dirige al Noroeste, coincidiendo la dirección del viento y pendiente.
- Ralentización de la propagación del fuego con pendiente negativa.
- Expansión al norte coincidiendo con la dirección del viento en modelo teórico de viento.
- No genera activaciones expontaneas y mantiene el interior del área Quemada al mismo nivel.

No obstante el área quemada es bastante más amplia que el teórico. Además, en los primeros instantes parece que la propagación no es lo suficientemente rápida.

## 6 Discusión

### 6.1 Interpretación de los resultados

Los modelos que se han propuesto en el presente proyecto han dado resultados diversos, no obstante, se han visto mejoras significativas en algunas propuestas según han ido evolucionando, siendo uno de los objetivos principales del presente proyecto.

Las primeras etapas del proyecto partiendo de su sencillez ya daban resultados “buenos” de un momento temporal al siguiente (Modelo I  $R^2$  Score superior a 0.99), no obstante, cuando se ejecutaba una evolución autorregresiva los errores se incrementan exponencialmente.

La segunda etapa (Modelos Ramificados y capas densas) han supuesto una mejora considerable en varios casos, la capacidad de no seguir una estructura lineal nos ha permitido desde incrementar la atención de ciertos canales hasta poder tener en cuenta características globales y locales.

Finalmente, la Etapa 3 abordó la no-localidad espacial mediante arquitecturas avanzadas como la Attention U-Net, lo que permitió al modelo integrar dependencias espaciales más amplias, capturando efectos retardados o indirectos del entorno en la evolución del fuego. Los resultados mostraron mejoras visibles en coherencia espacial y en la predicción de regiones de ignición secundaria o expansión acelerada.

### 6.2 Limitaciones del modelo y de los datos

- **Datos:** La reducida disponibilidad de imágenes de evolución de fuegos forestales, y la ausencia de contexto (climatología, inicialización etc.) ha obligado a realizar los entrenamientos con datos sintéticos, controlados, pero sin reflejar fielmente toda la variabilidad y ruido de la realidad
- **Resolución:** Las simulaciones y predicciones se hicieron a una resolución de 50\*50 píxeles, con una longitud de celda de 30\*30 metros. Esto se ha debido, sobre todo, a la capacidad computacional disponible.
- **Simplificaciones físicas:** Para la obtención de los datos sintéticos se tuvieron que realizar varias simplificaciones: modelados de viento simples; humedad relativa, temperaturas etc. constantes en el mapa; los tiempos de quemado, agotamiento de combustible etc. son aproximados. Otros factores no se han tenido directamente en cuenta como la radiación y procesos convectivos más complejos.
- **Capacidad computacional:** Ha sido limitada durante todo el proyecto (la mayor parte de este se ha desarrollado usando la herramienta Colab). Esto ha provocado la inviabilidad de entrenamiento de modelos más profundos y complejos.

### 6.3 Posibles mejoras en la arquitectura y los datos

#### En cuanto a datos:

- Generar una base de datos con la evolución real de los fuegos forestales, incluyendo datos climatológicos.
- En caso de incapacidad de generar una base de datos reales, una mejora significativa sería la reducción de la simplificación, añadiendo simulaciones de viento más precisas, e incluyendo un mapeo local de todos los datos climatológicos con sus variaciones que influyan en la propagación de los fuegos forestales.
- El programa principal de simulación del incendio forestal es una versión generada para el presente proyecto. Actualmente existen otras herramientas más precisas que podrían servir como generadoras de datos.

#### En cuanto a los modelos:

- El programa para la generación de la base de datos permite hacer una evolución de los modelos para que tengan en cuenta la dimensión temporal, lo que permitiría una mejora en la capacidad predictiva. No obstante, habría que tener en cuenta que los cambios súbitos en las condiciones climatológicas podrían pasar desapercibidos.
- Modelos más complejos y profundos: se probaron modelos basados en SwinTransformer, no obstante, la capacidad computacional disponible ha sido insuficiente para terminar de entrenar el modelo. Los resultados estaban siendo prometedores en las primeras fases del estudio. También se han probado modelos simples CNN sin resultados relevantes, pero con una profundización en el estudio podrían llegar a dar buenos resultados.
- Se probaron también modelos neuronales con proxy (programa independiente) de viento, no obstante, no se ha profundizado lo suficiente en la investigación como para obtener resultados adecuados pero con una mayor aplicabilidad.
- En general para el entrenamiento de los modelos se han usado los mismos factores y métricas de entrenamiento, un estudio más exhaustivo puede desembocar en resultados mucho mejores

### 6.4 Aplicabilidad real en contextos de predicción operativa

El objetivo final de este tipo de modelos sería facilitar herramientas de predicción rápida y precisa del avance del fuego, de forma que puedan asistir a cuerpos de protección civil, DCI y gestores forestales. En ese contexto:

- Se ve un gran potencial en los modelos para la predicción de la propagación, sobre todo en fases con poca separación temporal, donde el error base todavía no es apenas perceptible.
- Los modelos son capaces de propagar y direccionar el incendio en base a velocidad, dirección de viento, topografía y tipo de combustible lo que puede ayudar significativamente a las labores de extinción.

- Una vez entrenados los modelos, la velocidad de inferencia permitiría realizar simulaciones rápidas, adelantándose a la propagación, y ser retroalimentado con datos reales, que sería esencial para escenarios operativos.

No obstante, se debe mencionar que para una aplicabilidad real se deben solucionar o al menos reducir las limitaciones del programa comentadas en el apartado 5.2. de forma consistente.

## 7 Conclusiones

### 7.1 Resumen de los logros del trabajo.

Este trabajo ha demostrado la viabilidad y efectividad del uso de modelos de inteligencia artificial para predecir fenómenos físicos complejos, en concreto la evolución de los fuegos forestales. Para ello se han combinado variables estáticas y dinámicas, siendo el modelo capaz de adaptarse en función de las mismas. Para ello:

- Se ha realizado un desarrollo progresivo de modelos, desde arquitecturas básicas hasta redes avanzadas con atención.
- Implementación de modelos que son capaces, no solo de predecir efectos locales si no también globales en la propagación del fuego. Esto es clave para la reproducción de fenómenos físicos complejos.
- Evaluaciones cuantitativas y cualitativas que demuestran que la evolución en los modelos y sus arquitecturas pueden mejora de forma significativa la precisión y la coherencia espacial.

### 7.2 Reflexión sobre la viabilidad del uso de IA en este campo

El uso de la Inteligencia Artificial para la predicción de la evolución de fenómenos físicos, en concreto de incendios forestales, representa una herramienta poderosa de la cual se pueden sacar beneficios y aplicabilidad inmediatos como la mejora en la gestión operativa del riesgo, y la planificación en tiempo real:

- Predicciones rápidas y detalladas. No requiere simulaciones físicas complejas
- Adaptabilidad y Flexibilidad. Aprendizaje de patrones a partir de los datos. Facilidad de introducir nuevos factores con el incremento de capas de forma rápida y sencilla.
- Posibilidad de integrar información heterogénea.

Se han encontrado desafíos importantes como la dependencia de datos realistas y la validación de estos.

Se ha de mencionar, que los modelos de IA, no podrán llegar a superar las evoluciones teóricas de procesos hipotéticos-Teóricos, sin interferencias y “perfectos” (imposibles en sistemas físicos

reales), no obstante dado el ruido que se presenta en fenómenos físicos reales, son de gran utilidad para simular procesos en entornos abiertos/no simulados.

### 7.3 Futuras líneas de investigación o mejoras.

- obtención de datos mediante simulaciones complejas
- Aplicar los modelos sobre incendios históricos reales, combinando sensores remotos, meteorología reanalizada y mapas de combustible de alta resolución
- Modelos híbridos fisico-datadriven: combinar conocimiento físico explícito con aprendizaje automático y deeplearning, ya sea incorporando restricciones físicas en la pérdida o estructurando el modelo para respetar principios físicos como la conservación de masa o energía.
- Integraciones temporales: adaptación del modelo de forma automática a las diferentes fases del incendio

## 8 Bibliografía

•Sun, X., Li, N., Chen, D., Chen, G., Sun, C., Shi, M., Gao, X., Wang, K., & Hezam, I. M. (2024). *A Forest Fire Prediction Model Based on Cellular Automata and Machine Learning*. IEEE Access, 12, 55389–55403. <https://doi.org/10.1109/ACCESS.2024.3389035>

•Reddy, G. C. K., Adithya, M., Mohan, P. N., Babu, T. N., & Mahesh, B. (2025). *Forest fire spread prediction based on improved Wang Zhengfei model combined with cellular automata*. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 11(3). <https://doi.org/10.32628/CSEIT2511311>

•Chen, Y., Liu, Q., & Zhang, H. (2023). *Prediction of fire spread based on geographical cellular automata*. Remote Sensing Applications: Society and Environment, 32, 101078.

•Martínez Millán, J., Vignote, S., Martos, J., & Caballero, D. (1991). *Utilización de «CARDIN» para la simulación de un incendio real*. Forest Systems, 1, 121–133. <https://doi.org/10.5424/1299>

•He, T., Wang, Y., & Liu, F. (2022). *Simulation of forest fire spread based on artificial intelligence*. Ecological Indicators, 136, 108653. <https://doi.org/10.1016/j.ecolind.2022.108653>

•Liu, D., Zhang, M., & Chen, X. (2024). *Desarrollo de un simulador de incendios forestales: simulación de viento sobre relieve con técnicas de CFD*. Ingeniería Ambiental y Ciencias del Fuego, 12(2), 45–58.

- Zhou, Y., Kong, R., Li, X., & Wang, H.** (2025). *Comparative and interpretative analysis of CNN and Transformer models in predicting wildfire spread using remote sensing data*. arXiv preprint. <https://arxiv.org/abs/2503.14150>
  
- Jain, P., Coogan, S. C. P., Subramanian, S. G., Crowley, M., Taylor, S., & Flannigan, M. D.** (2020). *A review of machine learning applications in wildfire science and management*. *Environmental Reviews*, 28(4), 478–505. <https://doi.org/10.1139/er-2020-0019>
  
- Üstek, K., Ji, X., et al.** (2024). *Advancements in artificial intelligence applications for forest fire prediction*. *Forests*, 15(4), 704. <https://doi.org/10.3390/f15040704>
  
- Jain, P., Coogan, S. C. P., Subramanian, S. G., Crowley, M., Taylor, S., & Flannigan, M. D.** (2020). *A review of machine learning applications in wildfire science and management*. *Environmental Reviews*, 28(4), 478–505. <https://doi.org/10.1139/er-2020-0019>
  
- Burge, J., Bonanni, M., Ihme, M., & Hu, L.** (2020). *Convolutional LSTM neural networks for modeling wildland fire dynamics*. arXiv preprint arXiv:2012.06679. <https://arxiv.org/abs/2012.06679>
  
- Khennou, F., & Akhloufi, M. A.** (2023). *Improving wildland fire spread prediction using deep U-Nets*. *Science of Remote Sensing*, 8, 100101. <https://doi.org/10.1016/j.srs.2023.100101>