



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Optimización de Riego  
Deficitario Controlado en  
cultivos Leñosos Mediante  
Algoritmos de Machine  
Learning**



Presentado por Rodrigo Castroviejo Ausucua  
en Universidad de Burgos — 13 de febrero  
de 2026

Tutor: Carlos Cambra Baseca  
Co-tutora: Antonia Maiara Marques Do  
Nascimento







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. Carlos Cambra Baseca y D.<sup>a</sup> Antonia Maiara Marques Do Nascimento, profesores del departamento de digitalización, área de ciencia de la computación e inteligencia artificial.

Exponen:

Que el alumno D. Rodrigo Castroviejo Ausucua, con DNI 71565680F, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Optimización de Riego Deficitario Controlado en cultivos Leñosos Mediante Algoritmos de Machine Learning.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección de los que suscriben, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 13 de febrero de 2026

Vº. Bº. del Tutor:

D. Carlos Cambra Baseca

Vº. Bº. de la co-tutora:

D.<sup>a</sup> Antonia Maiara Marques Do Nascimento





## Resumen

La agricultura moderna enfrenta el doble reto de aumentar la productividad mientras reduce su impacto ambiental, particularmente en el uso de recursos hídricos. En este contexto, la integración de nuevas tecnologías y técnicas de Inteligencia Artificial (IA) está revolucionando la gestión agronómica, permitiendo soluciones precisas y adaptativas a problemas tradicionalmente abordados mediante métodos empíricos. Este Trabajo de Fin de Grado explora cómo la IA puede optimizar el Riego Deficitario Controlado (RDC) en almendros, un cultivo estratégico en regiones con escasez hídrica, demostrando el potencial transformador de estas tecnologías en el sector agrícola.

El estudio parte de la premisa de que la agricultura de precisión, apoyada en herramientas de machine learning, puede superar las limitaciones de los sistemas de riego convencionales. La investigación se centra en desarrollar un modelo predictivo que analiza datos multi-sensor, incluyendo variables edáficas (humedad del suelo a diferentes profundidades), climáticas (temperatura, radiación, precipitaciones) y registros de riego. Estos datos, procesados mediante técnicas avanzadas de análisis temporal, permiten al sistema aprender patrones complejos y predecir las necesidades hídricas del cultivo con una precisión inalcanzable para métodos tradicionales.

La implementación de algoritmos como SARIMA (Seasonal Auto-regressive Integrated Moving Average), VAR (Vector Autoregression) y redes LSTM(Long Short-Term Memory) evidencia cómo la IA puede adaptarse a los desafíos agrícolas, desde la gestión eficiente del agua hasta la detección temprana de estrés vegetal. Estos modelos no solo proporcionan recomendaciones de riego ajustadas a las condiciones reales del cultivo, sino que también identifican relaciones no lineales entre variables que escapan al análisis humano. Los resultados preliminares sugieren reducciones significativas en el consumo hídrico sin merma productiva, validando el enfoque propuesto.

Más allá de su aplicación específica, este trabajo ilustra el amplio espectro de posibilidades que ofrece la IA en agricultura. La tecnología desarrollada sirve como caso de estudio para demostrar cómo la transformación digital del sector agrario puede abordar problemas globales como la sequía o el cambio climático, estableciendo un puente entre innovación tecnológica y sostenibilidad ambiental. Futuras investigaciones podrían expandir este marco metodológico a otros cultivos leñosos o integrar datos de teledetección para una gestión aún más precisa.

**Descriptores**

riego deficitario controlado, machine learning, agricultura de precisión, eficiencia hídrica, almendro, series temporales, optimización de riego, sostenibilidad agrícola, aprendizaje automático.

## Abstract

Modern agriculture faces the dual challenge of increasing productivity while reducing environmental impact, particularly in water resource management. In this context, the integration of new technologies and Artificial Intelligence (AI) techniques is revolutionizing agronomic management, enabling precise and adaptive solutions to problems traditionally addressed through empirical methods. This Bachelor's Thesis explores how AI can optimize Regulated Deficit Irrigation (RDI) in almond trees - a strategic crop in water-scarce regions - demonstrating the transformative potential of these technologies in the agricultural sector.

The study is based on the premise that precision agriculture, supported by machine learning tools, can overcome the limitations of conventional irrigation systems. The research focuses on developing a predictive model that analyzes multi-sensor data, including edaphic variables (soil moisture at different depths), climatic factors (temperature, solar radiation, precipitation), and irrigation records. This data, processed through advanced temporal analysis techniques, enables the system to learn complex patterns and predict crop water requirements with an accuracy unattainable by traditional methods.

The implementation of algorithms such as SARIMA, VAR and LSTM networks demonstrates how AI can adapt to agricultural challenges, from efficient water management to early detection of plant stress. These models not only provide irrigation recommendations tailored to actual crop conditions, but also identify non-linear relationships between variables that escape human analysis. Preliminary results suggest significant reductions in water consumption without compromising yield, validating the proposed approach.

Beyond its specific application, this work illustrates the broad spectrum of possibilities offered by AI in agriculture: from automated crop monitoring to input optimization and yield prediction. The developed technology serves as a case study demonstrating how digital transformation of the agricultural sector can address global problems such as drought or climate change, building a bridge between technological innovation and environmental sustainability. Future research could expand this methodological framework to other woody crops or integrate remote sensing data for even more precise management.

**Keywords**

regulated deficit irrigation, machine learning, precision agriculture, water use efficiency, almond tree, time series, irrigation optimization, agricultural sustainability, automated learning.

---

# Índice general

---

Índice general	v
Índice de figuras	vii
Índice de tablas	ix
<b>1. Introducción</b>	<b>1</b>
1.1. Recursos del proyecto . . . . .	2
<b>2. Objetivos del proyecto</b>	<b>3</b>
2.1. Objetivos específicos . . . . .	4
<b>3. Conceptos teóricos</b>	<b>5</b>
3.1. Riego Deficitario Controlado (RDC) . . . . .	5
3.2. Cálculo de la necesidad de riego del cultivo . . . . .	6
3.3. ARIMA y variantes . . . . .	13
3.4. VAR . . . . .	17
3.5. LSTM . . . . .	19
<b>4. Técnicas y herramientas</b>	<b>25</b>
4.1. Técnicas . . . . .	25
4.2. Herramientas . . . . .	28
4.3. Herramientas para la gestión del proyecto . . . . .	29
4.4. Herramientas para el desarrollo del jupyter notebook [34] . .	30
4.5. Herramientas para el desarrollo de la aplicación web . . . . .	31
4.6. Herramientas usadas para el desarrollo de la memoria . . . . .	33
4.7. Uso de la IA . . . . .	34

<b>5. Aspectos relevantes del desarrollo del proyecto</b>	<b>35</b>
5.1. Inicio del proyecto . . . . .	35
5.2. Desarrollo del jupyter notebook . . . . .	36
5.3. Desarrollo de la aplicación web . . . . .	43
<b>6. Trabajos relacionados</b>	<b>49</b>
6.1. Trabajos Relacionados . . . . .	49
<b>7. Conclusiones y Líneas de trabajo futuras</b>	<b>53</b>
7.1. Líneas de trabajo futuras . . . . .	54
<b>Bibliografía</b>	<b>57</b>

---

# Índice de figuras

---

3.1. “Valor promedio de $K_{c,ini}$ en relación con el nivel de $ET_0$ siendo el intervalo entre riegos y/o lluvias significativas durante la etapa inicial de crecimiento para todos los tipos de suelo ligeros a moderados (3-10 mm por evento) [2] . . . . .	9
3.2. “Valor promedio de $K_{c,ini}$ en relación con el nivel de $ET_0$ siendo el intervalo entre riegos mayor o igual a 40 mm por evento de mojado, durante la etapa inicial de crecimiento, para suelos de textura gruesa [2]. . . . .	9
3.3. “Valor promedio de $K_{c,ini}$ en relación con el nivel de $ET_0$ siendo el intervalo entre riegos mayor o igual a 40 mm por evento de mojado, durante la etapa inicial de crecimiento, para suelos de textura fina y media [2]. . . . .	10
3.4. Curva de coeficiente del cultivo [2]. . . . .	11
3.5. “Celda de una red RNN [29] . . . . .	19
3.6. “Una red RNN “desplegada”[29] . . . . .	20
3.7. Linea horizontal que representa el estado de la celda. [29] . . . . .	21
3.8. Compuerta LSTM [29] . . . . .	21
3.9. Forget gate [29] . . . . .	22
3.10. Input gate [29] . . . . .	22
3.11. Combina el estado de la celda con el input y actualiza el estado [29] . . . . .	23
3.12. Output gate [29] . . . . .	23
4.1. Retrospectiva llevado a cabo tras la finalización del Sprint 8 . . . . .	26
4.2. Backlog scrum del sprint 8 . . . . .	27
4.3. Tablero kanban durante el sprint 8 . . . . .	28
4.4. Configuración de recursos usada en Docker para el proyecto . . . . .	32
5.1. Posición grupo de sensores de suelo 'Mini1' (41.812959, -4.578752)	36

5.2. Posición grupo de sensores de suelo 'Mini2' (41.815940, -4.576148)	37
5.3. Imagen satelite de Google Maps [21] de ambos grupos de sensores	37
5.4. Temperatura predicha con menos de 2 ciclos anuales de datos .	38
5.5. Figura con los warnings de modelos . . . . .	39
5.6. Datos sintéticos generados para Velocidad del viento . . . . .	39
5.7. Temperatura predicha con datos sintéticos . . . . .	40
5.8. Curva de la $K_c$ ajustada para nuestro cultivo . . . . .	41
5.9. Necesidad de riego a lo largo del periodo predicho con y sin RDC	42
5.10. Comparación consumo total de agua en el periodo predicho con y sin RDC . . . . .	42
5.11. Monitorización de docker RAM . . . . .	44
5.12. Configuración recursos docker-compose.yml . . . . .	45
5.13. Estructuras de almacenamiento de progreso y resultados . . . . .	46
7.1. Socios empresariales colaboradores del proyecto . . . . .	54
7.2. Centros de investigación colaboradores del proyecto . . . . .	54

---

# Índice de tablas

---

3.1. Etapas de desarrollo del almendro variedad Penta [16] [2] [25] . . . . .	8
4.1. Principales librerías utilizadas en el desarrollo de los Jupyter Notebooks. . . . .	31
4.2. Principales librerías utilizadas en el desarrollo web y análisis de datos. . . . .	33
4.3. Modelos de Inteligencia Artificial utilizados en el proyecto. . . . .	34



---

# 1. Introducción

---

Este Trabajo de Fin de Grado desarrolla un sistema innovador para optimizar el Riego Deficitario Controlado en cultivos de almendro mediante la integración de tecnologías Internet of Things (IoT) y técnicas avanzadas de machine learning. Numerosos estudios recientes han demostrado cómo la combinación de sensores de suelo, datos climáticos y algoritmos predictivos permite superar las limitaciones de los métodos empíricos tradicionales, ofreciendo soluciones basadas en evidencia objetiva y adaptable a condiciones variables [36].

El proyecto parte de un conjunto de datos previamente recopilados mediante sensores IoT que incluyen mediciones de humedad del suelo a diferentes profundidades, variables climáticas como temperatura, radiación solar y precipitación, junto con registros históricos de riego. Estos datos, inicialmente procesados a nivel básico, son sometidos a un tratamiento avanzado mediante un pipeline de análisis implementado en Python. Para este fin, se emplean herramientas especializadas como Pandas para la gestión eficiente de series temporales y operaciones de preprocesamiento a gran escala, junto con Scikit-learn como framework principal para la implementación de algoritmos clásicos de Machine Learning. Las fases que requieren visualización interactiva y análisis exploratorio se desarrollan en Jupyter Notebooks, que permiten documentar detalladamente el proceso.

La metodología de desarrollo sigue un enfoque que combina scripts Python autónomos, optimizados para tareas computacionalmente intensivas como la carga masiva de datos y el entrenamiento de modelos, con notebooks interactivos destinados a etapas que demandan interpretación visual y análisis de patrones.

La memoria se estructura en siete capítulos que abarcan desde los fundamentos teóricos del Riego Deficitario Controlado y las técnicas de aprendizaje automático aplicadas, hasta el análisis detallado de los resultados obtenidos. Como complemento a la documentación principal, se incluyen seis anexos técnicos que cubren aspectos como el plan de proyecto con sus cronogramas y gestión de riesgos, la especificación de requisitos con casos de uso y objetivos técnicos, el diseño del sistema incluyendo su arquitectura técnica y modelos de datos, manuales de programación y usuario con guías de implementación y operación del sistema, así como un análisis de sostenibilidad que evalúa el impacto del proyecto en relación con los Objetivos de Desarrollo Sostenible.

Este enfoque no solo valida la viabilidad técnica de aplicar Inteligencia Artificial en la gestión del riego, con reducciones demostradas en el consumo hídrico según pruebas preliminares, sino que también establece un marco metodológico replicable para otros cultivos leñosos en contextos de escasez hídrica. Los resultados contribuyen significativamente al avance de la agricultura de precisión, ofreciendo herramientas escalables que combinan rigor científico con practicidad operativa, marcando un hito en la aplicación de tecnologías emergentes a los desafíos agronómicos actuales.

## 1.1. Recursos del proyecto

Todo el trabajo realizado está disponible en los siguientes recursos en línea:

- **Repositorio en GitHub:** [Visitar repositorio GitHub](#)
- **Enlace a jupyter notebook web:** [Enlace a entorno jupyter desplegado.](#)
  - **Token acceso:** tribunalUBU
- **Enlace a aplicación web:** [Enlace a aplicación web](#)

---

## 2. Objetivos del proyecto

---

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

El objetivo general del proyecto es construir un modelo con el que a partir de los datos de la finca de 'Canduela' [5.2](#), en la que se esta llevando a cabo una plantacion intensiva de almendros con una recogida de datos con dispositivos IoT (Internet of Things), se pueda predecir semanalmente la cantidad de riego a suministrar siguiendo una politica de riego RDC (riego dificitario controlado). De forma que optimicemos la cantidad de agua, manteniendo la produccion de la planta. Hazaña de gran importancia en zonas donde no abunda el agua (sequias) y hay que administrarla de la manera más eficiente posible.

Los principales objetivos del proyecto son:

- Implementar un modelo en python que procese, entrene y predizca el riego semanal con los datos que se le proporcionan.
- Construir una pagina web que permita al usuario final entrenar,predecir y visualizar resultados.

## **2.1. Objetivos específicos**

### **Desarrollo sistema de entrenamiento, predicción y evaluación de los modelos**

El propósito de este aportado es describir la estructura necesaria para llevar a cabo la predicción de la necesidad de riego por cada instancia del dataset, para conseguir ello, nos marcamos los siguientes objetivos:

- Importar los datos a la estructura y en el formato adecuado para tratarlos posteriormente.
- Hacer un análisis inicial de los datos y preprocesarlos.
- Construir la estructura para serializar el proceso de entrenamiento, predicción y evaluación.
- Implementar los modelos de acuerdo a la estructura establecida.
- Guardar los modelos y preparar los medios para realizar una carga y predicción desde estos.
- Aplicar el cálculo de la necesidad de riego sobre el dataset predicho.
- Aplicar el cálculo de sistema de riego deficitario controlado sobre el dataset predicho.

### **Desarrollo de la aplicación web**

El propósito de este apartado es describir el desarrollo de una aplicación web para la gestión y visualización de los modelos y resultados. Los objetivos específicos son:

- Implementar una interfaz para la visualización de los resultados obtenidos por los diferentes modelos.
- Proporcionar una ruta para la subida de datos que posteriormente pueda usar en el entrenamiento el usuario.
- Proporcionar una interfaz clara y sencilla que permita a los usuarios de la aplicación entrenar y predecir con dichos modelos entrenados sobre un archivo de datos proporcionado por el usuario.
- Desplegar la aplicación para que sea accesible sin necesidad de la instalación de ningun software.

---

### 3. Conceptos teóricos

---

Este capítulo presenta los fundamentos teóricos y metodológicos esenciales para comprender el desarrollo del sistema de predicción de riego deficitario controlado (RDC) en el cultivo del almendro. Se abordan conceptos clave de agronomía, series temporales, aprendizaje automático y desarrollo de aplicaciones web que sustentan el proyecto.

#### 3.1. Riego Deficitario Controlado (RDC)

El Riego Deficitario Controlado (RDC) es una estrategia de gestión hídrica que aplica estrés hídrico controlado durante fases fenológicas específicas del cultivo, con el objetivo de reducir significativamente el consumo de agua mientras se mantiene la productividad. Esta técnica se basa en el principio de que no todas las fases del desarrollo vegetativo son igualmente sensibles al déficit hídrico [36].

En el cultivo del almendro, estudios experimentales han demostrado que aplicar reducciones controladas del riego (especialmente durante la fase de llenado de grano) puede generar resultados económicos positivos [36]. Los datos indican que se puede lograr:

- Reducción del 28 % en el agua aplicada
- Disminución de solo el 7 % en la producción
- Incremento del 11,27 % en el beneficio económico
- Mejora del 33 % en la eficiencia del uso del agua (de 0,18 a 0,24 kg/m<sup>3</sup>)

Estos resultados convierten al RDC en una técnica especialmente valiosa para regiones con limitaciones hídricas, donde la optimización del recurso agua es crítica tanto para la sostenibilidad ambiental como para la viabilidad económica de las explotaciones agrícolas.

La implementación exitosa del RDC requiere un conocimiento preciso de las fases fenológicas del cultivo y un sistema de monitorización que permita aplicar los déficits hídricos en los momentos óptimos, maximizando así el ahorro de agua mientras se minimiza el impacto en el rendimiento final.

### 3.2. Cálculo de la necesidad de riego del cultivo

Las necesidades de riego de un cultivo representan la cantidad de agua que debe aplicarse para compensar las pérdidas por evapotranspiración, manteniendo un balance hídrico que optimice la producción. El cálculo preciso de estas necesidades es fundamental para implementar estrategias como el Riego Deficitario Controlado (RDC), donde se busca aplicar déficits hídricos específicos que maximicen la eficiencia en el uso del agua [2].

#### Cálculo de la evotranspiración específica del cultivo ( $ET_c$ )

Para calcular la  $ET_c$ , necesitaremos previamente haber calculado la  $ET_0$  y la  $K_c$  específica para cada momento [2].

$$ET_c = K_c \times ET_0 \quad [2] \quad (3.1)$$

#### Cálculo de la evotranspiración de referencia ( $ET_0$ )

La metodología estándar para calcular la evapotranspiración de referencia ( $ET_0$ ) es la ecuación FAO Penman–Monteith, definida en la publicación FAO-56 [2]. Esta ecuación integra variables climáticas para estimar la demanda evaporativa de una superficie de referencia (césped bien regado):

$$ET_0 = \frac{0,408 \Delta (R_n - G) + \gamma \frac{900}{T+273} u_2 (e_s - e_a)}{\Delta + \gamma (1 + 0,34 u_2)} \quad [2] \quad (3.2)$$

donde:

- $R_n$ : radiación neta en la superficie del cultivo ( $\text{MJ m}^{-2} \text{día}^{-1}$ )
- $G$ : densidad del flujo de calor del suelo ( $\text{MJ m}^{-2} \text{día}^{-1}$ )
- $T$ : temperatura media del aire a 2 m de altura ( $^{\circ}\text{C}$ )
- $u_2$ : velocidad del viento a 2 m de altura ( $\text{m s}^{-1}$ )
- $e_s$ : presión de vapor de saturación (kPa)
- $e_a$ : presión de vapor real (kPa)
- $\Delta$ : pendiente de la curva de presión de vapor (kPa  $^{\circ}\text{C}^{-1}$ )
- $\gamma$ : constante psicrométrica (kPa  $^{\circ}\text{C}^{-1}$ )

### Variables necesarias para el cálculo

Para aplicar esta ecuación se requieren mediciones diarias de las siguientes variables meteorológicas:

- Temperatura del aire: valores máximo y mínimo diarios
- Humedad relativa: o datos necesarios para calcular la presión de vapor
- Radiación solar: o, en su defecto, horas de sol para su estimación
- Velocidad del viento: medida a 2 m sobre la superficie
- Presión atmosférica: utilizada para el ajuste de la constante psicrométrica

### Cálculo de la $K_c$

#### Coeficiente de cultivo ( $K_c$ )

El coeficiente de cultivo ( $K_c$ ) relaciona la evapotranspiración de referencia ( $ET_0$ ) con la evapotranspiración real del cultivo ( $ET_c$ ) mediante la siguiente expresión [2]:

$$ET_c = K_c \times ET_0 \quad [2] \quad (3.3)$$

#### Etapas de desarrollo del cultivo

Para el almendro (*Prunus dulcis*, variedad Penta), el ciclo fenológico se divide en cuatro etapas con duraciones específicas, mostradas en la Tabla 3.1.

Tabla 3.1: Etapas de desarrollo del almendro variedad Penta [16] [2] [25]

Etapa	Duración (días)	Descripción
Inicial	15	Desde brotación (día 59) hasta floración (día 74)
Desarrollo	50	Desde floración hasta cuajado completo
Mediados	98	Máxima actividad fisiológica y llenado de frutos
Final	31	Desde inicio de maduración hasta cosecha

### Valores base de $K_c$ para almendro

Según FAO-56 (Tabla 12, almendros sin cubierta vegetal) [2], los valores base del coeficiente de cultivo son:

$$K_{c,ini} = 0,40 \quad (3.4)$$

$$K_{c,mid} = 0,90 \quad (3.5)$$

$$K_{c,end} = 0,65 \quad [2] \quad (3.6)$$

### Ajuste de $K_c$ a condiciones locales

**Ajuste de  $K_{c,ini}$**  Para sistemas de riego por goteo con mojado parcial del suelo, el coeficiente inicial se ajusta mediante:

$$K_{c,ini} = f_w \times K_{c,ini}^{(Fig)} \quad [2] \quad (3.7)$$

donde  $f_w$  es la fracción de superficie mojada (típicamente entre 0.3 y 0.4 para riego por goteo).

El valor  $K_{c,ini}^{(Fig)}$  se obtiene a partir de las Figuras 29 o 30 de FAO-56 [2], considerando:

- Intervalo entre eventos de mojado
- Profundidad de infiltración ajustada ( $I_w$ ):

$$I_w = \frac{I}{f_w} \quad [2] \quad (3.8)$$

donde:

- $I$ : lámina de riego aplicada sobre el área total del campo (mm/evento)
- $f_w$ : fracción de superficie mojada (típicamente entre 0.3 y 0.4 para riego por goteo)

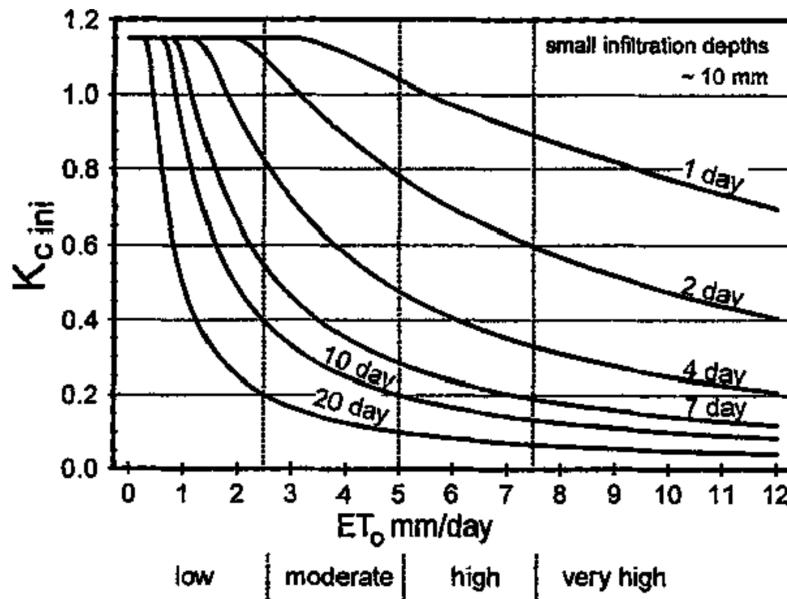


Figura 3.1: “Valor promedio de  $K_{c,ini}$  en relación con el nivel de  $ET_0$  siendo el intervalo entre riegos y/o lluvias significativas durante la etapa inicial de crecimiento para todos los tipos de suelo ligeros a moderados (3-10 mm por evento) [2] .

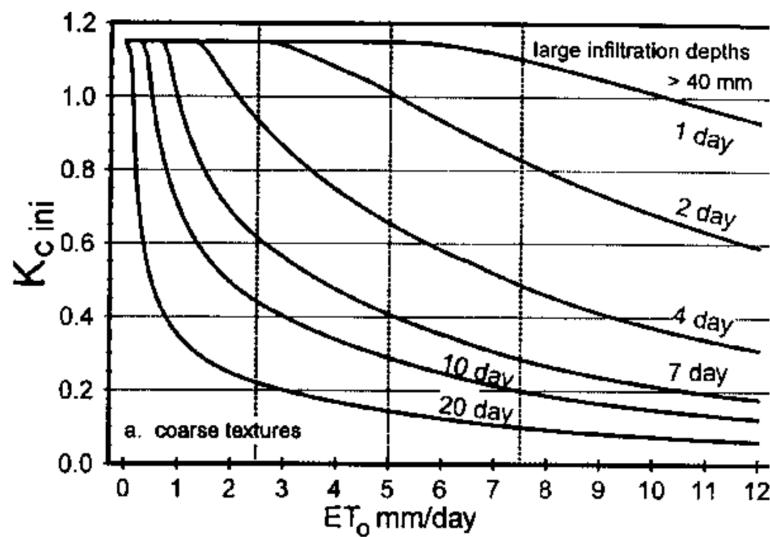


Figura 3.2: “Valor promedio de  $K_{c,ini}$  en relación con el nivel de  $ET_0$  siendo el intervalo entre riegos mayor o igual a 40 mm por evento de mojado, durante la etapa inicial de crecimiento, para suelos de textura gruesa [2].

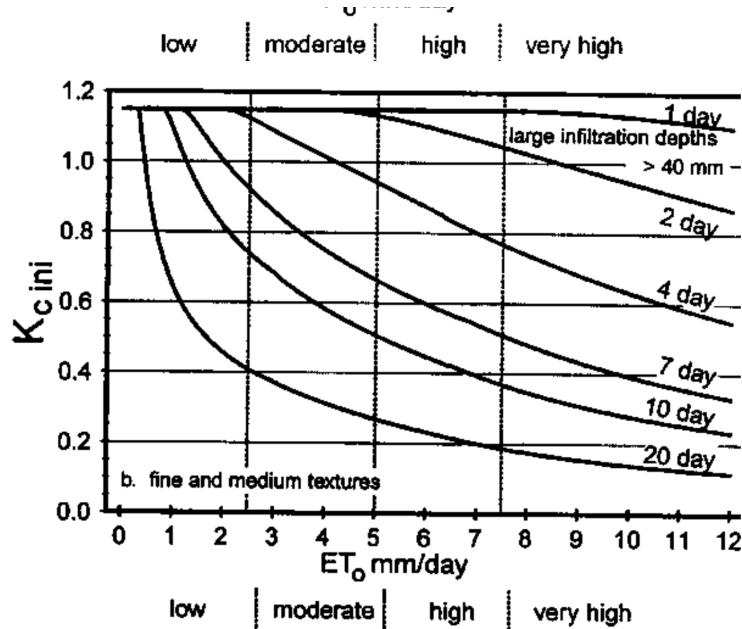


Figura 3.3: “Valor promedio de  $K_{c,ini}$  en relación con el nivel de  $ET_0$  siendo el intervalo entre riegos mayor o igual a 40 mm por evento de mojado, durante la etapa inicial de crecimiento, para suelos de textura fina y media [2].

**Ajuste de  $K_{c,mid}$  y  $K_{c,end}$**  Para considerar condiciones climáticas locales, se aplica el siguiente ajuste:

$$K_{c,aj} = K_c^{(Tab)} + [0,04(u_2 - 2) - 0,004(RH_{\min} - 45)] \left(\frac{h}{3}\right)^{0,3} \quad [2] \quad (3.9)$$

donde:

- $u_2$ : velocidad media del viento a 2 m de altura ( $\text{m s}^{-1}$ )
- $RH_{\min}$ : humedad relativa mínima media (%)
- $h$ : altura del cultivo (m).

**Nota:** Para el coeficiente  $K_{c,end}$ , este ajuste solo se aplica si:

$$K_{c,end}^{(Tab)} > 0,45 \quad [2] \quad (3.10)$$

### Construcción de la curva temporal de $K_c$

La evolución temporal del coeficiente de cultivo se representa mediante una función lineal por tramos [2]:

- **Etapa inicial:** valor constante igual a  $K_{c,ini}$
- **Etapa de desarrollo:** interpolación lineal desde  $K_{c,ini}$  hasta  $K_{c,mid}$
- **Etapa de mediados:** valor constante igual a  $K_{c,mid}$
- **Etapa final:** interpolación lineal desde  $K_{c,mid}$  hasta  $K_{c,end}$

FIGURE 34. Crop coefficient curve

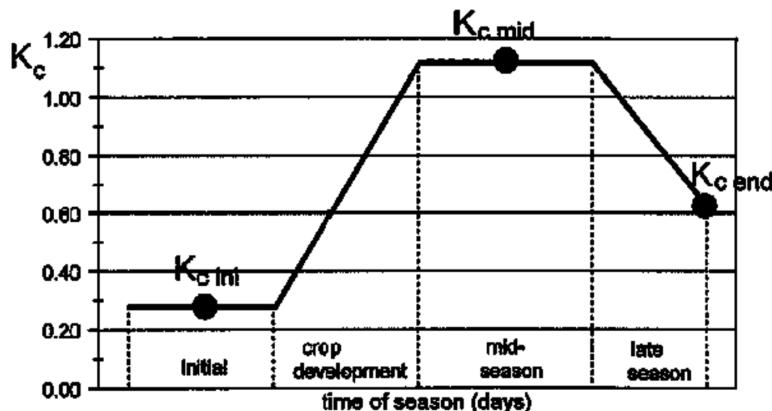


Figura 3.4: Curva de coeficiente del cultivo [2].

### Obtención del valor de $K_c$ para un día concreto

El coeficiente de cultivo ( $K_c$ ) para cualquier período de la estación de crecimiento puede determinarse considerando que, durante las etapas inicial y de mediados de temporada,  $K_c$  permanece constante e igual al valor correspondiente a la etapa fenológica considerada [2].

Durante la etapa de desarrollo del cultivo y la etapa final, el coeficiente de cultivo varía linealmente entre el valor de  $K_c$  al final de la etapa previa ( $K_{c,prev}$ ) y el valor de  $K_c$  al inicio de la etapa siguiente ( $K_{c,next}$ ), que en el caso de la etapa final corresponde a  $K_{c,end}$ :

$$K_{c,i} = K_{c,prev} + \left( \frac{i - \sum L_{prev}}{L_{stage}} \right) (K_{c,next} - K_{c,prev}) \quad [2] \quad (3.11)$$

donde:

- $i$ : número de día dentro de la estación de crecimiento ( $1 \leq i \leq$  duración total del ciclo)
- $K_{c,i}$ : coeficiente de cultivo en el día  $i$
- $L_{\text{stage}}$ : duración de la etapa fenológica considerada (días)
- $\sum L_{\text{prev}}$ : suma de las duraciones de todas las etapas previas (días)

## Cálculo de las necesidades brutas de riego

### Necesidades netas

Las necesidades netas de riego ( $R_N$ ) consideran la contribución de las precipitaciones efectivas ( $P_{ef}$ ) [2]:

$$R_N = ET_c - P_{ef} \quad [2] \quad (3.12)$$

donde  $P_{ef}$  es la porción de precipitación que realmente está disponible para el cultivo.

### Eficiencia del sistema de riego

Las necesidades brutas de riego ( $R_B$ ) incorporan la eficiencia del sistema ( $E_a$ ) [2]:

$$R_B = \frac{R_N}{E_a} \quad [2] \quad (3.13)$$

Para riego por goteo,  $E_a$  típicamente varía entre 0.85 y 0.95 [2].

### Aplicación en Riego Deficitario Controlado (RDC)

En estrategias de Riego Deficitario Controlado, se aplica un coeficiente de estrés ( $K_s$ ) que reduce las necesidades [36]:

$$R_B^{\text{RDC}} = \frac{ET_c \times K_s - P_{ef}}{E_a} \quad [36] \quad (3.14)$$

donde  $K_s$  varía según la fase fenológica y el nivel de déficit deseado (por ejemplo,  $K_s = 0.8$  para reducir un 20 % el riego durante la fase de llenado de grano) [36].

### 3.3. ARIMA y variantes

#### Estacionalidad

Una serie temporal estacionaria es aquella cuyas propiedades estadísticas no dependen del instante de tiempo en el que se observa la serie [23]. Por lo tanto, las series temporales con tendencia o con estacionalidad no son estacionarias, ya que la tendencia y la estacionalidad afectan al valor de la serie en distintos momentos del tiempo. Por el contrario, una serie de ruido blanco sí es estacionaria: no importa cuándo se observe, su comportamiento será prácticamente el mismo en cualquier punto del tiempo.

Algunos casos pueden resultar confusos: una serie temporal con comportamiento cíclico (pero sin tendencia ni estacionalidad) es estacionaria. Esto se debe a que los ciclos no tienen una longitud fija, por lo que antes de observar la serie no se puede determinar con certeza dónde se situarán los picos y los valles de dichos ciclos [23].

En general, una serie temporal estacionaria no presenta patrones predecibles a largo plazo. Las representaciones gráficas en el tiempo muestran la serie aproximadamente horizontal (aunque puede existir cierto comportamiento cíclico), con varianza constante [23].

#### Modelo AR

En un modelo autorregresivo, la variable de interés se predice utilizando una combinación lineal de valores pasados de la propia variable. El término *autorregresión* indica que se trata de una regresión de la variable sobre sí misma [23].

Así, un modelo autorregresivo de orden  $p$  puede escribirse como

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t \quad [23] \quad (3.15)$$

donde  $\varepsilon_t$  es un proceso de ruido blanco. Este modelo es similar a una regresión múltiple, pero utilizando valores retardados de  $y_t$  como variables explicativas. Nos referimos a este modelo como un modelo AR( $p$ ), es decir, un modelo autorregresivo de orden  $p$  [23].

Los modelos autorregresivos son notablemente flexibles para manejar una amplia variedad de patrones en series temporales. [23]

## Modelo MA y su relación con AR

En lugar de utilizar valores pasados de la variable a predecir en una regresión, un modelo de media móvil utiliza errores de predicción pasados en un modelo de tipo regresión [23]:

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} \quad [23] \quad (3.16)$$

donde  $\varepsilon_t$  es un proceso de ruido blanco. Nos referimos a este modelo como un modelo MA( $q$ ), es decir, un modelo de media móvil de orden  $q$ . Evidentemente, los valores de  $\varepsilon_t$  no se observan, por lo que no se trata de una regresión en el sentido habitual [23].

Obsérvese que cada valor de  $y_t$  puede interpretarse como una media móvil ponderada de los errores de predicción pasados (aunque los coeficientes no suelen sumar uno) [23].

Es posible expresar cualquier modelo AR( $p$ ) estacionario como un modelo MA( $\infty$ ) [23]. Por ejemplo, utilizando sustituciones repetidas, podemos demostrarlo para un modelo AR(1):

$$\begin{aligned} y_t &= \phi_1 y_{t-1} + \varepsilon_t \\ &= \phi_1(\phi_1 y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \\ &= \phi_1^2 y_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\ &= \phi_1^3 y_{t-3} + \phi_1^2 \varepsilon_{t-2} + \phi_1 \varepsilon_{t-1} + \varepsilon_t \\ &\vdots \end{aligned} \quad [23] \quad (3.17)$$

Siempre que  $-1 < \phi_1 < 1$ , el valor de  $\phi_1^k$  disminuye a medida que  $k$  aumenta. Por tanto, finalmente se obtiene

$$y_t = \varepsilon_t + \phi_1 \varepsilon_{t-1} + \phi_1^2 \varepsilon_{t-2} + \phi_1^3 \varepsilon_{t-3} + \cdots \quad [23] \quad (3.18)$$

lo que constituye un proceso MA( $\infty$ ).

El resultado inverso también se cumple si imponemos ciertas restricciones sobre los parámetros del modelo MA. En ese caso, el modelo MA se denomina *invertible* [23]. Es decir, cualquier proceso MA( $q$ ) invertible puede expresarse como un proceso AR( $\infty$ ) [23]. Los modelos invertibles no se introducen únicamente para permitir la conversión de modelos MA en modelos AR, sino que también presentan propiedades matemáticas deseables [23].

Por ejemplo, considérese el proceso MA(1):

$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1}. \quad [23] \quad (3.19)$$

En su representación AR( $\infty$ ), el error más reciente puede expresarse como una combinación lineal de las observaciones actuales y pasadas:

$$\varepsilon_t = \sum_{j=0}^{\infty} (-\theta_1)^j y_{t-j} \quad [23] \quad (3.20)$$

Cuando  $|\theta_1| > 1$ , los pesos aumentan a medida que crecen los retardos, de modo que las observaciones más alejadas en el tiempo tienen mayor influencia sobre el error actual [23]. Cuando  $|\theta_1| = 1$ , los pesos son constantes y las observaciones lejanas tienen la misma influencia que las más recientes [23]. Dado que ninguna de estas situaciones resulta razonable, se requiere que

$$|\theta_1| < 1 \quad [23] \quad (3.21)$$

de forma que las observaciones más recientes tengan mayor peso que las del pasado lejano. Por tanto, el proceso es invertible cuando  $|\theta_1| < 1$  [23].

Esta capacidad de representar un modelo MA(1) como un AR( $\infty$ ), nos permite usar el modelo más simple que capture los patrones [23].

Supóngase una serie temporal para la cual sería necesario ajustar un modelo AR(20) para capturar su estructura de dependencia. En este caso, puede resultar preferible ajustar un modelo MA(1) o MA(2), ya que estos modelos pueden representar patrones similares con un menor número de parámetros, facilitando la estimación y la interpretación del modelo [23].

## SARIMA

El proceso de construcción del modelo SARIMA está diseñado para aprovechar la asociación en las relaciones secuenciales con retardos que normalmente existen en los datos recogidos de forma periódica [7]. Una serie temporal es generada por un modelo SARIMA si:

$$\Phi_P(L^s) \phi_p(L) (1 - L)^d (1 - L^s)^D y_t = \Theta_Q(L^s) \theta_q(L) \varepsilon_t \quad [23] \quad (3.22)$$

donde  $n$  es el número de observaciones;  $p, d, q, P, D, Q$  son enteros;  $L$  es el operador retardo (lag);  $s$  es la longitud del período estacional;  $d$  es el número de diferenciaciones regulares  $(1 - L)^d$ ;  $D$  es el número de diferenciaciones estacionales  $(1 - L^s)^D$ ; y  $\varepsilon_t$  es el residuo estimado en el instante  $t$ , que sigue un proceso de ruido blanco gaussiano (WN).

Además:

$$\phi_p(L) = 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p \quad [7]$$

es el operador autorregresivo regular (AR) de orden  $p$ ;

$$\theta_q(L) = 1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q \quad [7]$$

es el operador de media móvil regular (MA) de orden  $q$ ;

$$\Phi_P(L^s) = 1 - \Phi_1 L^s - \Phi_2 L^{2s} - \cdots - \Phi_P L^{Ps} \quad [7]$$

es el operador autorregresivo estacional (SAR) de orden  $P$ ;

$$\Theta_Q(L^s) = 1 + \Theta_1 L^s + \Theta_2 L^{2s} + \cdots + \Theta_Q L^{Qs} \quad [7]$$

es el operador de media móvil estacional (SMA) de orden  $Q$ .

## SARIMAX

Al extender el modelo SARIMA mediante la inclusión de variables externas, se obtiene el modelo SARIMAX, el cual presenta una mejor capacidad de predicción a largo plazo. No obstante, la incorporación de factores externos hace que el modelo SARIMAX sea más vulnerable a las fluctuaciones de los datos. Este modelo es muy similar al SARIMA, con la diferencia de que emplea análisis de correlación para predecir de forma más precisa el impacto de influencias externas [14].

La Ecuación (12) describe el modelo SARIMAX, donde  $r$  representa la diferenciación de la variable externa:

$$\text{SARIMAX}(p, d, q, s, r) = \text{SARIMA}(p, d, q, s) + \sum_{i=1}^r \gamma_i x_i \quad [14] \quad (3.23)$$

El modelo SARIMAX ofrece un mejor rendimiento cuando existen fuertes correlaciones entre las variables independientes y la variable dependiente. [14]

### 3.4. VAR

Una limitación de los modelos considerados hasta ahora es que imponen una relación unidireccional: la variable a predecir está influida por las variables explicativas, pero no ocurre lo contrario [23]. Sin embargo, existen muchos casos en los que también debería permitirse la relación inversa, es decir, situaciones en las que todas las variables se influyen mutuamente [23].

Un modelo VAR es una generalización del modelo autorregresivo univariante para la predicción de un vector de series temporales. Consta de una ecuación por cada variable del sistema [23]. En el lado derecho de cada ecuación se incluye una constante y retardos de todas las variables del sistema [23]. Para simplificar, se considera un modelo VAR de dos variables con un único retardo. Un modelo VAR(1) bidimensional puede escribirse como:

$$y_{1,t} = c_1 + \phi_{11,1}y_{1,t-1} + \phi_{12,1}y_{2,t-1} + \varepsilon_{1,t}, \quad [23] \quad (3.24)$$

$$y_{2,t} = c_2 + \phi_{21,1}y_{1,t-1} + \phi_{22,1}y_{2,t-1} + \varepsilon_{2,t}, \quad [23] \quad (3.25)$$

donde  $\varepsilon_{1,t}$  y  $\varepsilon_{2,t}$  son procesos de ruido blanco que pueden estar correlacionados de forma contemporánea. El coeficiente  $\phi_{ii,\ell}$  captura la influencia del retardo  $\ell$  de la variable  $y_i$  sobre sí misma, mientras que el coeficiente  $\phi_{ij,\ell}$  representa la influencia del retardo  $\ell$  de la variable  $y_j$  sobre la variable  $y_i$ .

Si las series son estacionarias, la predicción se realiza ajustando directamente un modelo VAR a los datos, lo que se conoce como un *VAR en niveles* [23]. Si las series son no estacionarias, se diferencian previamente para hacerlas estacionarias y posteriormente se ajusta un modelo VAR, conocido como *VAR en diferencias* [23]. En ambos casos, los modelos se estiman ecuación por ecuación utilizando el método de mínimos cuadrados. Para cada ecuación, los parámetros se estiman minimizando la suma de los cuadrados de los errores  $\varepsilon_{i,t}$ . [23]

## Predecir con VAR

Las predicciones generadas a partir de un VAR se obtienen de forma recursiva [23]. El modelo VAR produce predicciones para cada variable incluida en el sistema [23]. Para ilustrar el procedimiento, supóngase que se ha ajustado el modelo VAR(1) bidimensional definido en las Ecuaciones (3.24)–(3.25) utilizando observaciones hasta el instante  $T$  [23]. Entonces, las predicciones a un paso vista se obtienen como:

$$\hat{y}_{1,T+1|T} = \hat{c}_1 + \hat{\phi}_{11,1}y_{1,T} + \hat{\phi}_{12,1}y_{2,T}, \quad [23] \quad (3.26)$$

$$\hat{y}_{2,T+1|T} = \hat{c}_2 + \hat{\phi}_{21,1}y_{1,T} + \hat{\phi}_{22,1}y_{2,T} \quad [23] \quad (3.27)$$

Estas expresiones tienen la misma forma que las Ecuaciones (3.24)–(3.25), excepto que los términos de error se han fijado a cero y los parámetros han sido sustituidos por sus estimaciones. Para un horizonte de predicción  $h = 2$ , las predicciones vienen dadas por:

$$\hat{y}_{1,T+2|T} = \hat{c}_1 + \hat{\phi}_{11,1}\hat{y}_{1,T+1|T} + \hat{\phi}_{12,1}\hat{y}_{2,T+1|T}, \quad [23] \quad (3.28)$$

$$\hat{y}_{2,T+2|T} = \hat{c}_2 + \hat{\phi}_{21,1}\hat{y}_{1,T+1|T} + \hat{\phi}_{22,1}\hat{y}_{2,T+1|T} \quad [23]. \quad (3.29)$$

De nuevo, la forma del modelo es la misma, pero los valores desconocidos de  $y_1$  y  $y_2$  se sustituyen por sus predicciones. Este procedimiento puede iterarse para cualquier horizonte temporal futuro [23].

## Uso modelo VAR

Al utilizar un modelo VAR para la predicción, deben tomarse dos decisiones fundamentales: el número de variables incluidas en el sistema, denotado por  $K$ , y el número de retardos, denotado por  $p$  [23]. El número total de coeficientes a estimar en un VAR es  $K + pK^2$ , o bien  $1 + pK$  por ecuación [23]. Por ejemplo, en un VAR con  $K = 5$  variables y  $p = 3$  retardos, se estiman 16 coeficientes por ecuación, lo que da un total de 80 coeficientes. Cuantos más coeficientes deban estimarse, mayor será el error de estimación que se introduce en la predicción [23].

En la práctica, se suele mantener un valor reducido de  $K$  [23], incluyendo únicamente variables que estén correlacionadas entre sí y que, por tanto, sean útiles para predecirse mutuamente. Los criterios de información se emplean

habitualmente para seleccionar el número de retardos a incluir. No obstante, debe tenerse cuidado al utilizar el AICc, ya que tiende a seleccionar un número elevado de retardos; en su lugar, para modelos VAR suele preferirse el uso del BIC [23]. Existen versiones más sofisticadas del modelo, como los *VAR dispersos* (en los que muchos coeficientes se fijan a cero) o enfoques basados en *shrinkage*, donde los coeficientes se estiman con valores más pequeños [23].

## 3.5. LSTM

### Recurrent Neural Networks

Los seres humanos no comenzamos nuestro proceso de pensamiento desde cero en cada instante. Al leer este texto, comprendemos cada palabra en función del entendimiento de las palabras anteriores; no descartamos toda la información previa para volver a empezar a pensar desde cero. El pensamiento humano presenta, por tanto, persistencia.

Las redes neuronales tradicionales no son capaces de reproducir este comportamiento, lo que constituye una limitación importante [29]. Por ejemplo, si se desea clasificar el tipo de evento que ocurre en cada instante de una película, no resulta evidente cómo una red neuronal tradicional podría utilizar la información sobre eventos anteriores para influir en la interpretación de los posteriores.

Las redes neuronales recurrentes abordan este problema. Se trata de redes que incorporan bucles en su arquitectura, lo que permite que la información persista a lo largo del tiempo [29].

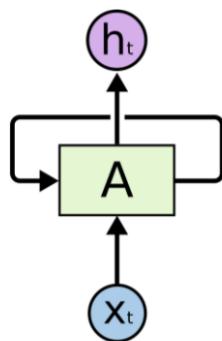


Figura 3.5: “Celda de una red RNN [29]

Podemos visualizar mejor esta retroalimentación de las celdas que componen una RNN "desplegando" el bucle que la comprende.

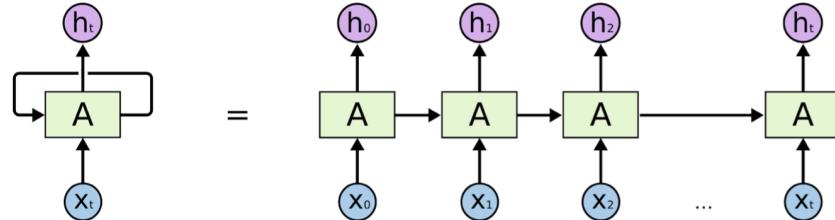


Figura 3.6: "Una red RNN "desplegada" [29]

Desafortunadamente, las RNNs no son capaces de en la práctica, de recordar contexto que se aleja mucho en el espacio de la celda actual [29]. No son capaces de mantener esas dependencias de largo plazo. Sin embargo, las LSTMs, una evolución de estas RNNs, solucionan esta problemática [29].

## LSTM Networks

Las redes *Long Short-Term Memory*, conocidas habitualmente como *LSTM*, son un tipo especial de redes neuronales recurrentes capaces de aprender dependencias a largo plazo. Fueron introducidas por Hochreiter y Schmidhuber en 1997 [22], y posteriormente refinadas y popularizadas por numerosos trabajos posteriores. En la actualidad, ofrecen un rendimiento sobresaliente en una amplia variedad de problemas y se utilizan de forma generalizada [29].

Las LSTM están diseñadas explícitamente para evitar el problema de las dependencias a largo plazo [29]. Recordar información durante períodos prolongados de tiempo forma parte de su funcionamiento natural, y no supone una dificultad que el modelo deba aprender a superar.

### El nucleo de las celdas LSTM

La clave del funcionamiento de las redes LSTM es el *estado de la celda*, representado como la línea horizontal que recorre la parte superior del diagrama [29].

El estado de la celda puede entenderse como una especie de cinta transportadora. Atraviesa toda la cadena prácticamente en línea recta, con tan solo algunas interacciones lineales menores, lo que permite que la información fluya a lo largo de ella sin apenas modificaciones.

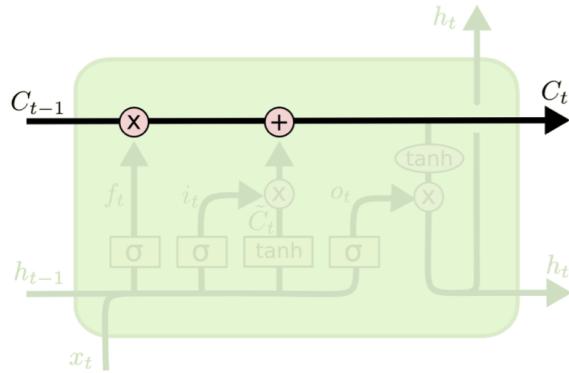


Figura 3.7: Linea horizontal que representa el estado de la celda. [29]

No obstante, la LSTM tiene la capacidad de eliminar o añadir información al estado de la celda. Este proceso está cuidadosamente controlado por unas estructuras denominadas *compuertas* [22].

Las compuertas permiten decidir de forma selectiva qué información se deja pasar. Están formadas por una capa neuronal con función sigmoide y una operación de multiplicación punto a punto [29].

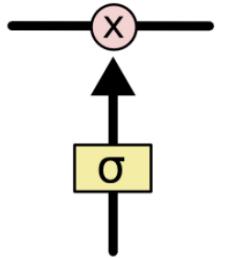


Figura 3.8: Compuerta LSTM [29]

La capa sigmoide produce valores comprendidos entre cero y uno, que indican qué proporción de cada componente debe transmitirse. Un cero significa que no se deja pasar información, mientras que un uno implica que se transmite toda [29].

Una red LSTM dispone de tres compuertas de este tipo, cuya función es proteger y regular el estado de la celda [29].

El primer paso en una red LSTM consiste en decidir qué información se va a descartar del estado de la celda. Esta decisión la toma una capa sigmoide

denominada *compuerta de olvido (forget gate)* [29]. Dicha compuerta analiza el estado oculto anterior  $h_{t-1}$  y la entrada actual  $x_t$ , y produce un valor entre 0 y 1 para cada componente del estado de la celda anterior  $C_{t-1}$  [29]. Un valor de 1 indica que la información se conserva por completo, mientras que un valor de 0 implica que se elimina totalmente [29].

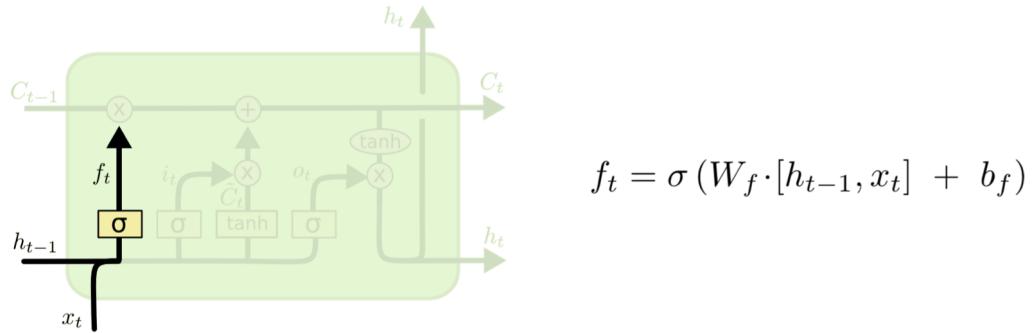


Figura 3.9: Forget gate [29]

El siguiente paso consiste en decidir qué nueva información se va a almacenar en el estado de la celda [29]. Este proceso tiene dos partes. En primer lugar, una capa sigmoide llamada *compuerta de entrada (input gate)* determina qué valores se van a actualizar [29]. A continuación, una capa *tanh* genera un vector de nuevos valores candidatos, denotado por  $\tilde{C}_t$ , que podrían incorporarse al estado [29]. En el siguiente paso, ambas salidas se combinan para actualizar el estado de la celda [29].

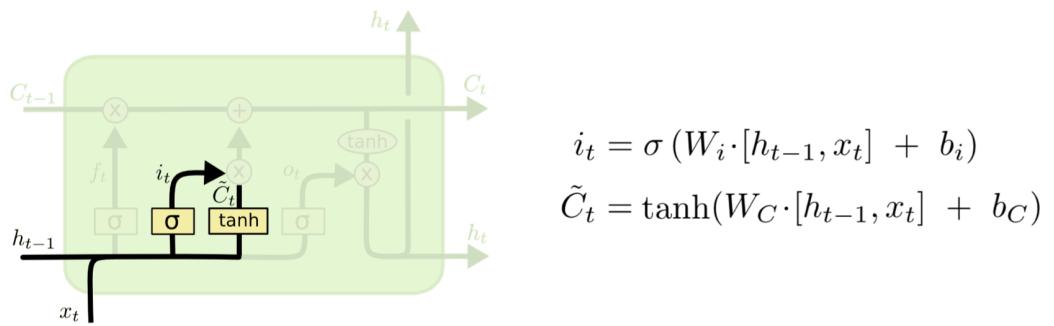


Figura 3.10: Input gate [29]

Una vez tomadas estas decisiones, se actualiza el estado de la celda anterior  $C_{t-1}$  para obtener el nuevo estado  $C_t$  [29]. Para ello, se multiplica el estado antiguo por  $f_t$ , eliminando así la información que se decidió descartar

[29]. A continuación, se suma el término  $i_t \cdot \tilde{C}_t$ , que representa los nuevos valores candidatos ponderados según el grado en que se ha decidido actualizar cada componente del estado [29].

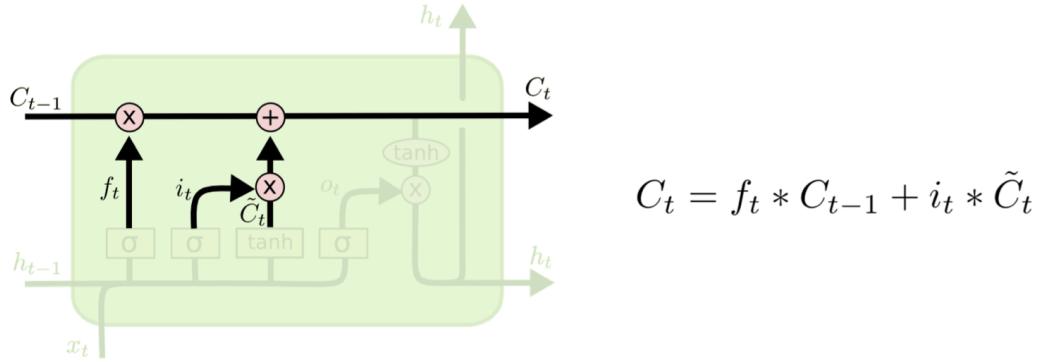


Figura 3.11: Combina el estado de la celda con el input y actualiza el estado [29]

Finalmente, se decide cuál será la salida de la red en este instante [29]. Esta salida se basa en el estado de la celda, pero en una versión filtrada del mismo [29]. En primer lugar, se aplica una capa sigmoide que determina qué partes del estado de la celda se van a emitir [29]. Después, el estado de la celda se pasa por una función  $\tanh$ , que restringe sus valores al intervalo  $[-1, 1]$ , y se multiplica por la salida de la compuerta sigmoide, de modo que solo se transmiten las componentes seleccionadas [29].

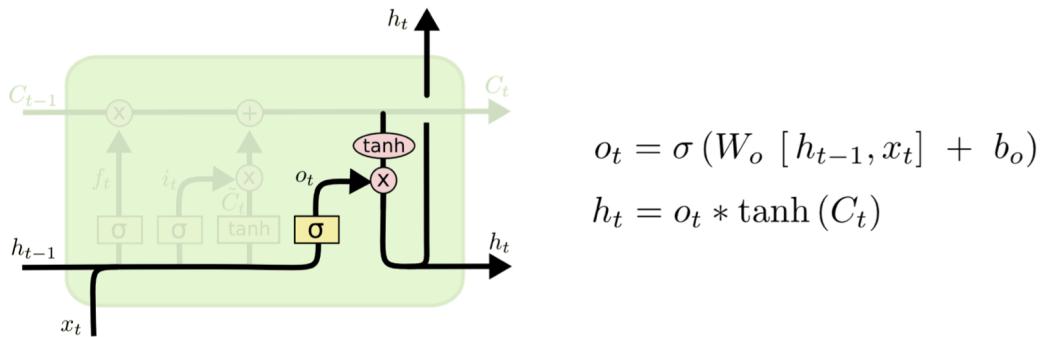


Figura 3.12: Output gate [29]



---

## 4. Técnicas y herramientas

---

Este capítulo presenta las técnicas, herramientas y bibliotecas empleadas durante el desarrollo del proyecto. Se detallan las decisiones técnicas tomadas, junto con una breve comparativa de alternativas consideradas, justificando las elecciones realizadas en cada caso.

### 4.1. Técnicas

#### SCRUM

Scrum [37] es una metodología ágil muy extendida en el ámbito del desarrollo de software, especialmente adecuada para proyectos en los que los requisitos evolucionan con rapidez y se requiere una entrega progresiva de funcionalidades. Su propósito principal es promover el trabajo colaborativo, aumentar la transparencia en el desarrollo y facilitar la adaptación continua a los cambios.

Este marco de trabajo se estructura en ciclos iterativos denominados *sprints*, cuya duración habitual oscila entre una y cuatro semanas. En cada sprint se definen los objetivos y tareas a realizar, se llevan a cabo reuniones diarias de seguimiento (*daily stand-ups*) y, al finalizar el ciclo, se realiza una revisión del trabajo completado (*sprint review*), junto con una retrospectiva orientada a detectar posibles mejoras en el proceso.

confluence-article-iframe

Editar Compartir ...

## Resumen Sprint 8 - 9/10/2025

De Rodrigo Castroviejo Ausucua Ver visualizaciones Añade una reacción

**Fecha**  
9 dic 2025

**Participantes**  
@Rodrigo Castroviejo Ausucua

**Como ha finalizado el sprint...**

Actividad	October	November	Decem
Sprints	Tablero Sp...	Tablero Sp...	Tablero Sp...
Publicaciones	Tablero Sp...	Tablero Sp...	Tablero Sp...
RIEGOIA-1 Importar datos	FINALIZ		
RIEGOIA-6 EDA inicial	FINALIZ		
RIEGOIA-17 EDA selección de features	FINALIZ		
RIEGOIA-18 Implementar matriz de correlación de atributos respecto a la variable objetivo	FINALIZ		
RIEGOIA-19 Analizar matriz correlación y sacar primeras conclusiones	FINALIZ		
RIEGOIA-20 Finalizar sprint	FINALIZ		

Figura 4.1: Retrospectiva llevado a cabo tras la finalización del Sprint 8

En el desarrollo de este proyecto se aplicaron diversas prácticas propias de Scrum adaptadas a el equipo conformado por solo 1 persona, como la planificación de sprints con una duración de dos semanas y la evaluación de las tareas finalizadas al término de cada iteración. Este enfoque permitió mantener la atención en los objetivos prioritarios, incrementar la eficiencia del equipo y garantizar una evolución continua del proyecto.

The screenshot shows the Atlassian Jira interface. On the left is a sidebar with navigation links: 'Para ti', 'Recientes', 'Marcados como favorito', 'Aplicaciones', 'Planes', 'Espacio', 'Recientes', 'IA en Riego deficitario del cultivo del almendro' (which is highlighted in blue), 'Más espacios', 'Explorar plantillas', 'Filtros', 'Paneles', 'Confluence', 'Equipos', and 'Personalizar barra lateral'. The main content area is titled 'IA en Riego deficitario del cultivo del almendro' and shows the 'Backlog' tab selected. Below the title, there is a search bar and a 'Filtro' button. The main content area displays a table for 'Tablero Sprint 9' (10 dic – 24 dic (2 actividades)). The table has two rows:

Issue	Nombre	Tareas	Estado
RIEGOJA-45	Calculo necesidad de riego deficitario final diaria cultivo sobre i...	CALCULO NECESIDA...	TAREAS POR HACER
RIEGOJA-55	Crear página web muestra de resultados de modelos	DESARROLLO Y UNIR...	EN CURSO

Below this, there is a section for 'Backlog' with the message 'Tu backlog está vacío.' and a 'Crear sprint' button.

Figura 4.2: Backlog scrum del sprint 8

## Kanban

Kanban [3] [28] es una técnica de gestión visual del trabajo que facilita el control del flujo de tareas mediante un tablero organizado en columnas que representan las distintas fases del proceso, como *Pendiente*, *En progreso* y *Completado*. Su finalidad es mejorar la eficiencia, minimizar los tiempos de entrega y evitar la acumulación excesiva de trabajo.

Como complemento a la planificación basada en Scrum, se empleó un tablero Kanban a través de *Atlassian Jira*. Las tareas se organizaron en las columnas *Backlog*, *In Progress* y *Done*, lo que permitió disponer en todo momento de una visión clara y actualizada del estado del proyecto. Para la creación de tareas se utilizaron *stories* e *issues* en el repositorio, en los que se especificaban el título, una descripción detallada, un epic al que pertenecían y una aproximación de su duración en forma de puntos de historia siguiendo la serie de Fibonacci para asignar estos. El tablero utilizado durante el desarrollo del proyecto se muestra en la Figura 4.3.

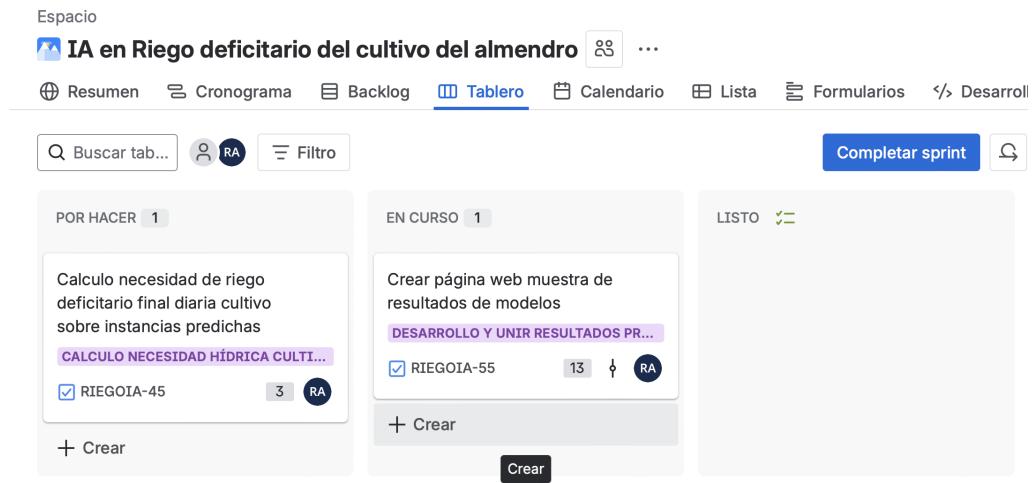


Figura 4.3: Tablero kanban durante el sprint 8

## 4.2. Herramientas

### Visual Studio Code [27]

Ha sido tanto para la ejecución del jupyter notebook, como para el desarrollo de la aplicación web y la redacción de memoria el IDE seleccionado.

He usado este entorno para el desarrollo de mi proyecto por mi familiaridad con el entorno y la fácil customización para los diferentes tipos de proyectos. Manteniendo una única aplicación para el desarrollo de los diferentes apartados de este proyecto.

### Python [35]

Se eligió Python (versión 3.10 o superior) como lenguaje de programación principal por la familiaridad con el mismo y por la amplia disponibilidad de librerías y recursos relacionados con la minería de datos. Como posibles alternativas podrían considerarse MATLAB o R, sin embargo, aprender un nuevo entorno habría aumentado innecesariamente la complejidad del proyecto, además de que estas opciones resultan menos adecuadas para los objetivos planteados: R presenta ciertas limitaciones por su antigüedad y su uso puede resultar engoroso, mientras que MATLAB cuenta con menor

soporte de la comunidad en el ámbito de la minería de datos en comparación con Python.

Python se utiliza en varias partes del proyecto. Por un lado, en los Jupyter Notebooks, que permiten combinar código Python con celdas Markdown para explicar conceptos y ofrecen una visualización más clara de los resultados que un archivo Python tradicional. Por otro lado, Python se emplea en el desarrollo de la aplicación web mediante el uso del framework ligero Flask, así como en los distintos componentes asociados a esta.

### 4.3. Herramientas para la gestión del proyecto

#### Github

Github [18] junto con la herramienta git [17] se usó como plataforma para alojar el repositorio y se empleó el sistema de issues para organizar tareas y seguimiento del desarrollo.

He elegido github frente a otras páginas similares como gitlab. Por familiaridad y mayor extensión de github.

#### Atlassian Jira [6]

En este proyecto, Jira se utiliza como herramienta de gestión de tareas y planificación, facilitando el seguimiento de los avances y la organización de los sprints. Se eligió Jira frente a otras alternativas, como GitHub Projects, que podría haber sido más accesible al estar ya trabajando con GitHub. Jira ofrece un seguimiento de proyectos más completo, permitiendo, además de un tablero Kanban, una gestión más eficiente de los sprints y la integración de múltiples funcionalidades, como la posibilidad de documentar retrospectivas de sprint o crear ramas vinculadas a issues e histories directamente desde Jira en GitHub. Adicionalmente, el entorno resultaba familiar, ya que se había trabajado previamente con él en la asignatura de Gestión de Proyectos.

## 4.4. Herramientas para el desarrollo del jupyter notebook [34]

### Set up para desarrollo de jupyter notebooks en VS Code

He instalado el entorno jupyter labs/notebook [34] para poder ejecutar archivos .ipynb.

Además tengo instalado las siguientes extensiones de la store de VS code:

- **jupyter**: Extensión de Visual Studio Code que añade soporte básico para Jupyter Notebooks y permite usar cualquier entorno Python como kernel, siempre que tenga Jupyter instalado.
- **Jupyter keymap**: Esta extensión proporciona atajos de teclado para notebooks en Visual Studio Code que replican los de Jupyter Notebook. Viene incluida con la extensión Jupyter para VS Code y puede deshabilitarse o desinstalarse.
- **Jupyter cell tags**: Viene incluida con la extensión Jupyter para VS Code y puede deshabilitarse o desinstalarse. No la uso para nada en el proyecto
- **jupyter slide show**: Viene incluida con la extensión Jupyter para VS Code y puede deshabilitarse o desinstalarse. No la uso para nada en el proyecto

## Librerías usadas en el desarrollo del Jupyter Notebook

Librería	Descripción breve
Matplotlib.pyplot	Gráficos 2D
NumPy	Cálculos numéricos y matrices
Seaborn	Visualización estadística
Pandas	Manipulación de datos en DataFrames
Statsmodels	Modelos SARIMA, SARIMAX, VAR
Scikit-learn [33]	Aprendizaje automático y preprocesado
TensorFlow/Keras	Redes neuronales LSTM
Joblib	Guardado de modelos y objetos
Scipy.stats	Pruebas estadísticas
Datetime	Manejo de fechas y tiempos
Math	Funciones matemáticas
OS	Interacción con el sistema operativo
Warnings	Gestión de avisos y alertas
Typing	Anotaciones de tipos (Dict, List, Tuple)

Tabla 4.1: Principales librerías utilizadas en el desarrollo de los Jupyter Notebooks.

## 4.5. Herramientas para el desarrollo de la aplicación web

### Docker [13]

Se ha usado para el desarrollo de la aplicación web y el despliegue de el jupyter notebook/aplicación web para su visualización a traves de un enlace. Docker nos permite tener en un contenedor la aplicación y no tener que instalar en nuestro ordenador las dependencias necesarias para la aplicación. Docker hace de capa de isolación y contiene en un contenedor el proyecto. Además, se ha escogido para su posterior facilidad para ser desplegada la aplicación.

#### Set up de docker

Se necesita configurar docker para que permita crear contenedores con mínimo 10-12GB, ya que los modelos consumen gran cantidad de esta, al almacenarse en ella durante la predicción.

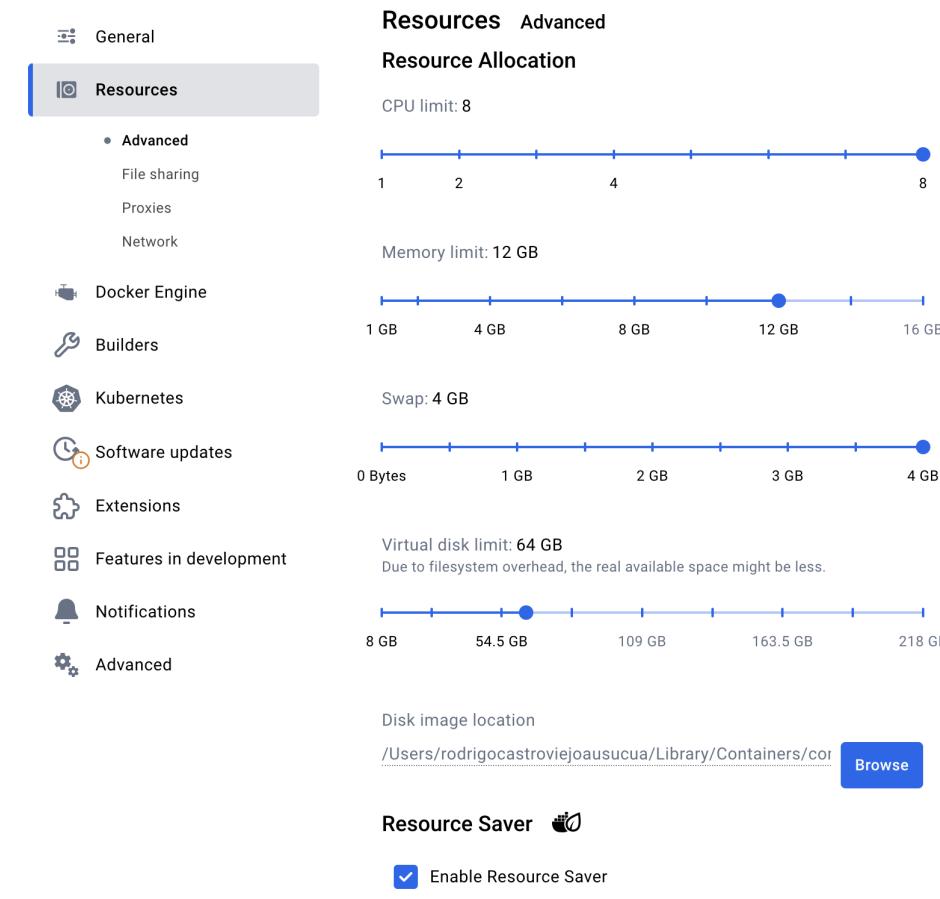


Figura 4.4: Configuración de recursos usada en Docker para el proyecto

La estructura de archivos de configuración de Docker utilizada en el proyecto es la siguiente:

- **Dockerfile:** define la imagen Docker de la aplicación, especificando la imagen base, las dependencias necesarias, la configuración del entorno y los comandos que se ejecutan para construir el contenedor.
- **docker-compose.yml:** permite definir y gestionar servicios Docker de forma sencilla, facilitando la ejecución de operaciones recurrentes como construir, levantar y detener contenedores mediante un único comando.

## Flask [5]

Se eligió Flask frente a otras alternativas principalmente por dos motivos. En primer lugar, ya existía un conocimiento básico del entorno, lo que facilitaba el desarrollo inicial. En segundo lugar, otros frameworks populares, como Spring Boot en Java o React.js y Vue.js en JavaScript, están orientados a proyectos más complejos, ofreciendo numerosas herramientas y funcionalidades diseñadas para aplicaciones de mayor envergadura. Por el contrario, Flask destaca por su simplicidad y ligereza, características que se adaptan mejor a este proyecto, dado que el objetivo principal no es la creación de la aplicación web en sí, sino la investigación y la implementación de modelos sobre los datos.

## Dependencias complementarias usadas en el desarrollo Web

Librería	Descripción breve
Flask_SQLAlchemy	ORM para bases de datos con Flask
Flask_Migrate	Gestión de migraciones de DB
Flask-Login	Autenticación y sesiones de usuarios
NumPy	Cálculos numéricos y matrices
Pandas	Manipulación de datos en DataFrames
Scikit-learn	Aprendizaje automático y minería de datos
Joblib	Serialización de objetos y modelos
Statsmodels	Modelos estadísticos y series temporales
TensorFlow	Redes neuronales y deep learning
Seaborn	Visualización estadística de datos
Matplotlib	Creación de gráficos 2D
Openpyxl	Lectura y escritura de archivos Excel

Tabla 4.2: Principales librerías utilizadas en el desarrollo web y análisis de datos.

## 4.6. Herramientas usadas para el desarrollo de la memoria

### Set up de VsCode para desarrollo de proyecto LateX

¿Porque he escogido esto en vez de la popular opción de Overleaf [31]?

Porque para documentos grandes, la carga con el plan basico en overleaf es algo lenta.

En mi entorno, he instalado la versión minima para ejecutar el entorno LateX, en mi caso (ya que trabajo en Mac) MacTeX [26], la versión más reciente.

Dispongo de la extensión LateX Workshop, disponible en la sección de extensiones. Esta me aporta los siguientes beneficios que facilitan el desarrollo de la memoria:

- Atajos de teclado: por ejemplo, cuando una línea comienza con `\item` o `\item[]`, al pulsar Enter se añade automáticamente una nueva línea con la misma estructura.
- Fragmentos de código (\*snippets\*): muchas órdenes de LaTeX pueden escribirse mediante fragmentos que comienzan con `\`; basta con escribir parte del comando para filtrar las opciones disponibles.
- Intellisense: incluye autocompletado para claves bibliográficas y etiquetas.
- Visualización del PDF en tiempo real, tanto dentro de Visual Studio Code como en el navegador.
- Compilación automática de documentos LaTeX (incluyendo BibTeX) a PDF al guardar el archivo.

## 4.7. Uso de la IA

Se han usado a lo largo del desarrollo técnico, como de la documentación, herramientas de IA para la consulta de dudas y resolución de problemas de código. A continuación en la [Tabla 4.3](#), se especifican las páginas y modelos usados a lo largo del transcurso del proyecto:

Tabla 4.3: Modelos de Inteligencia Artificial utilizados en el proyecto.

Plataforma	Modelos Específicos	Referencia
Gemini (Google)	Gemini 3	[20]
ChatGPT (OpenAI)	GPT-4.1/mini, GPT-4o, GPT-5 mini/Instant	[30]
DeepSeek (DeepSeek-AI)	DeepSeek-R1, DeepSeek-V3	[11]

---

## 5. Aspectos relevantes del desarrollo del proyecto

---

### 5.1. Inicio del proyecto

El proyecto se enmarca dentro del paradigma de la agricultura 5.0 [1], cuyo objetivo principal es la integración de tecnologías avanzadas en el sector agrícola para optimizar los procesos productivos. En particular, la creciente escasez de recursos hídricos durante los períodos de sequía en zonas del centro-sur de la península ibérica pone de manifiesto la necesidad de llevar a cabo una gestión del agua más eficiente y sostenible.

En este contexto surge este trabajo, cuyo propósito es mejorar el rendimiento económico del cultivo del almendro mediante la aplicación de un riego deficitario controlado. Para ello, se emplean datos obtenidos a partir de un conjunto de sensores instalados en la plantación, con el objetivo de reducir el consumo de agua sin afectar de manera significativa ni a la calidad ni a la cantidad de la producción, logrando así una mayor rentabilidad del cultivo.

Con este fin, se diseña un sistema de predicción basado en algoritmos de minería de datos, desarrollados en Python [35] y ejecutados en entornos Jupyter Notebook [34], que permite estimar de forma precisa las necesidades hídricas específicas de la plantación bajo un esquema de riego deficitario controlado.

Finalmente, y con el objetivo de facilitar la utilización y la visualización de los resultados obtenidos, se desarrolla una aplicación web complementaria. Esta aplicación integra las tecnologías implementadas y permite al usuario

entrenar modelos, realizar predicciones y visualizar los resultados de manera accesible, intuitiva y dinámica.

## 5.2. Desarrollo del jupyter notebook

### Importación de datos

Una vez definidas las primeras líneas del proyecto y el stack tecnológico 3.5, se procedió a la importación de los datos proporcionados por la co-tutora, obtenidos a través de una página web[8] que permitía acceder a los sensores de la finca 'Canduela', situada en Cubillas de Santa Marta, Valladolid, y descargar los datos en formato .xlsx para los períodos deseados. En una primera aproximación se seleccionó un amplio conjunto de sensores de suelo, con el objetivo de disponer de la mayor cantidad de información posible, dejando para fases posteriores la selección definitiva de variables.

Debido a las limitaciones de la plataforma, los datos se recibieron en varios ficheros separados por franjas temporales anuales y diferenciando entre sensores de suelo y sensores climáticos. Los sensores de suelo, a su vez, se dividían en dos grupos (mini1 y mini2), correspondientes a sensores ubicados en diferentes zonas de la finca (véanse Figuras 5.1 y 5.2).



Figura 5.1: Posición grupo de sensores de suelo 'Mini1' (41.812959, -4.578752)



Figura 5.2: Posición grupo de sensores de suelo 'Mini2' (41.815940, -4.576148)

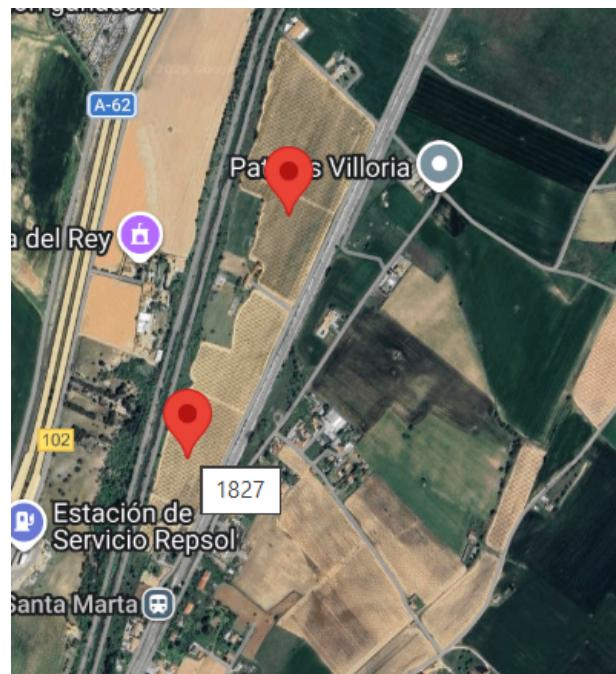


Figura 5.3: Imagen satelite de Google Maps [21] de ambos grupos de sensores

Como estrategia de importación, se unificaron los ficheros correspondientes a distintos años para cada grupo de sensores, obteniendo una estructura

de datos consolidada que sirvió como base para el resto del trabajo de investigación. La carga de los datos se realizó desde un archivo .py independiente, con el fin de optimizar el rendimiento debido al elevado volumen de información manejada.

## Creación de datos sintéticos

Tras una fase inicial de investigación, se optó por implementar modelos de predicción para series temporales multivariantes con el objetivo de estimar valores futuros del conjunto de datos y, a partir de ellos, calcular la necesidad diaria de riego de la plantación.

La estrategia seguida consistió en comenzar con modelos sencillos y, en caso de obtener buenos resultados, avanzar hacia modelos de mayor complejidad. Siguiendo este planteamiento, se implementaron los modelos SARIMA, SARIMAX y VAR [2.1](#).

Sin embargo, una vez evaluados, los resultados obtenidos fueron claramente [Figura 5.4](#), con métricas de rendimiento muy bajas. El análisis posterior permitió identificar como principal causa la escasez de datos disponibles, especialmente la falta de suficientes ciclos estacionales completos. Al contar únicamente con dos ciclos anuales, los modelos no disponen de información suficiente para aprender patrones estacionales de forma fiable.

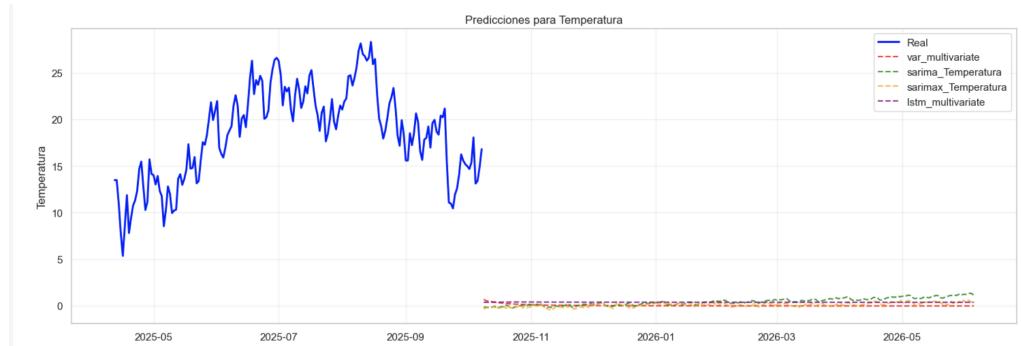


Figura 5.4: Temperatura predicha con menos de 2 ciclos anuales de datos

Esta limitación se reflejó también en los avisos generados por los modelos SARIMA y SARIMAX [Figura 5.5](#), que mostraban problemas de convergencia y reajustes automáticos de parámetros, lo que reforzó la hipótesis de una falta de datos históricos.

```
ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check
mle_retrials"Check mle_retrials", ConvergenceWarning)
```

Figura 5.5: Figura con los warnings de modelos

Ante esta situación, se decidió adoptar una estrategia alternativa basada en la generación de datos sintéticos, con el objetivo de ampliar el conjunto de datos y mejorar el aprendizaje de los modelos. Para ello, se implementaron dos técnicas: la simulación basada en física agrícola y el bootstrapping estacional con perturbación.

Tras comparar los datos sintéticos generados por ambos métodos con los datos reales, se seleccionó la simulación basada en física agrícola, al ofrecer resultados más coherentes y realistas. En azul podemos observar los datos reales y en naranja los datos sintéticos generados [Figura 5.6](#).

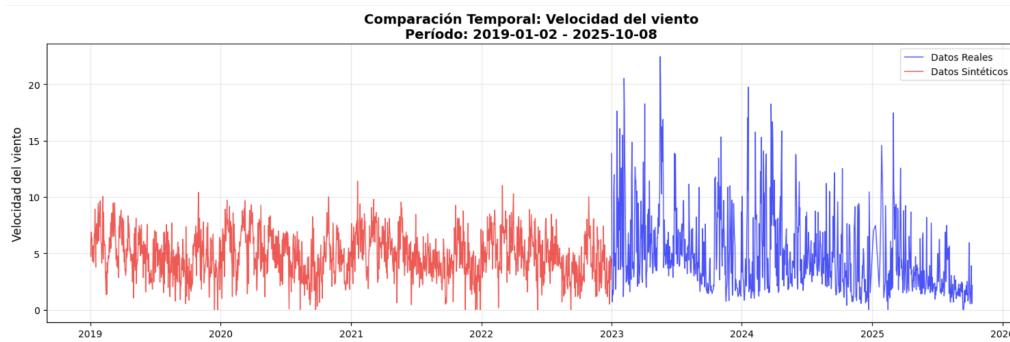


Figura 5.6: Datos sintéticos generados para Velocidad del viento

Tras entrenar los modelos con estos nuevos datos, conseguimos una mejora notable en las métricas de evaluación [Figura 5.6](#), sin embargo, esto no terminara de solucionar nuestros problemas como trataremos en posteriores apartados.

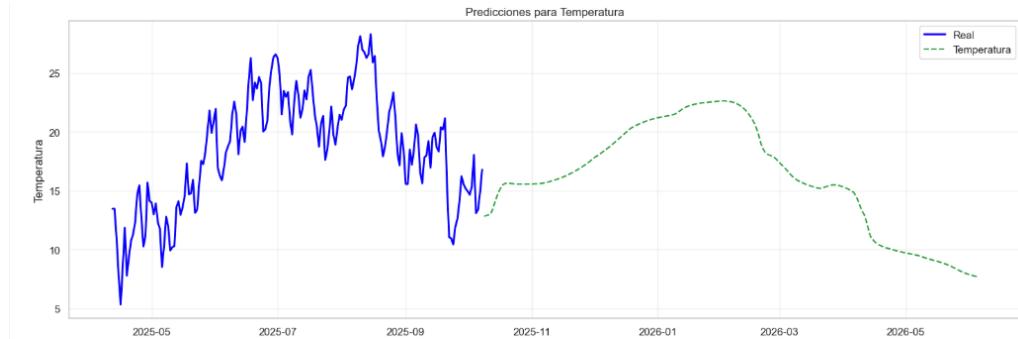


Figura 5.7: Temperatura predicha con datos sintéticos

## Reestructuración arquitectura entrenamiento y evaluación

Un aspecto crítico durante la fase de desarrollo fue la transición de un flujo de trabajo manual y propenso a errores hacia una arquitectura robusta basada en el concepto de *Pipelines* [33]. Inicialmente, el entrenamiento de modelos como LSTM presentaba riesgos de **fuga de datos (data leakage)** y una alta carga de error humano al realizar el escalado y la división de conjuntos de forma independiente.

Para mitigar estos problemas, se rediseñó el proceso de entrenamiento y evaluación bajo los siguientes pilares:

- **Estandarización mediante Wrappers:** Se desarrollaron clases personalizadas para modelos externos (*SARIMA*, *SARIMAX*, *VAR* y *LSTM* Capítulo 2.1), permitiendo su integración nativa en el ecosistema de *scikit-learn* [33].
- **Automatización del flujo:** La implementación de *Pipelines* [33] garantiza que todas las transformaciones de datos se apliquen de manera secuencial y aislada, asegurando la integridad de los resultados.
- **Persistencia de modelos:** Se integró un sistema de serialización para guardar y cargar modelos entrenados, optimizando los ciclos de prueba y evitando reentrenamientos redundantes tras el reinicio del entorno de ejecución.

Los detalles técnicos sobre la implementación de los *wrappers*, el uso de las clases *BaseEstimator* y *TransformerMixin* [33], así como el esquema

detallado de esta arquitectura, se exponen en profundidad en la Sección C.3 (*Reestructuración de arquitectura entrenamiento y evaluación*).

## Calculo de la necesidad de riego

Sobre los atributos predichos, prodecemos a realizar una investigación a cerca del cálculo de la  $K_c$ , necesaria para el cálculo de la fórmula de penman-monteith [2] sobre la que gira nuestra estrategia para calcular la necesidad de riego.

Recopilamos información acerca de las etapas fenológicas de nuestro variedad de almendro [25] [16] y siguiendo el artículo de la FAO 56 [2], establecemos los cálculos necesarios para realizar el cálculo de la  $K_c$  en cada una de las etapas ( $K_{c,ni}$ ,  $K_{c,id}$  y  $K_{c,end}$ ) Capítulo 2.1 y proceder a la determinación numérica de  $K_c$  mediante la formula (3.2) para cada una de las instancias para las que queremos predecir el riego.

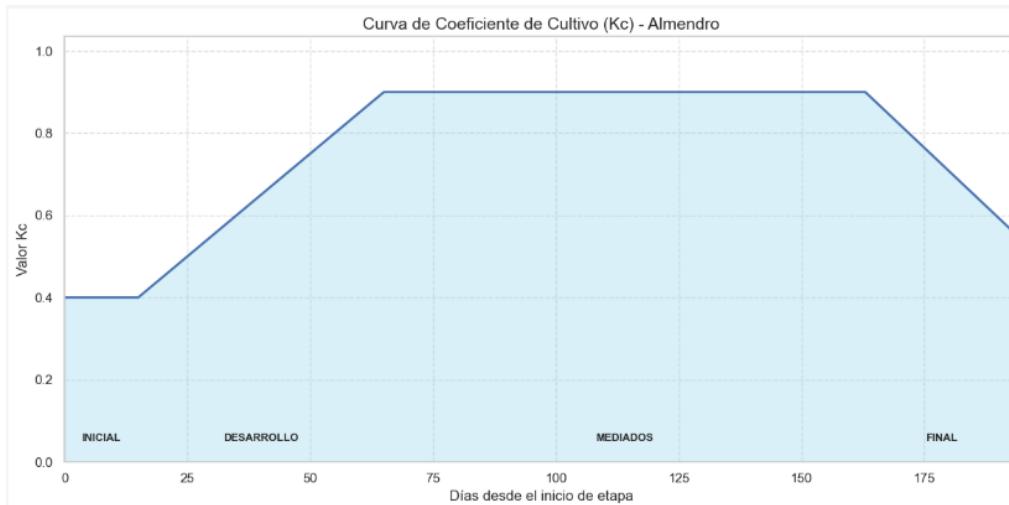


Figura 5.8: Curva de la  $K_c$  ajustada para nuestro cultivo

Finalmente, calculamos la evotranspiración de referencia ( $Et_0$ ) para cada instancia y usamos la formula de la figura (3.2) para determinar la evostranspiración específica ( $Et_c$ ).

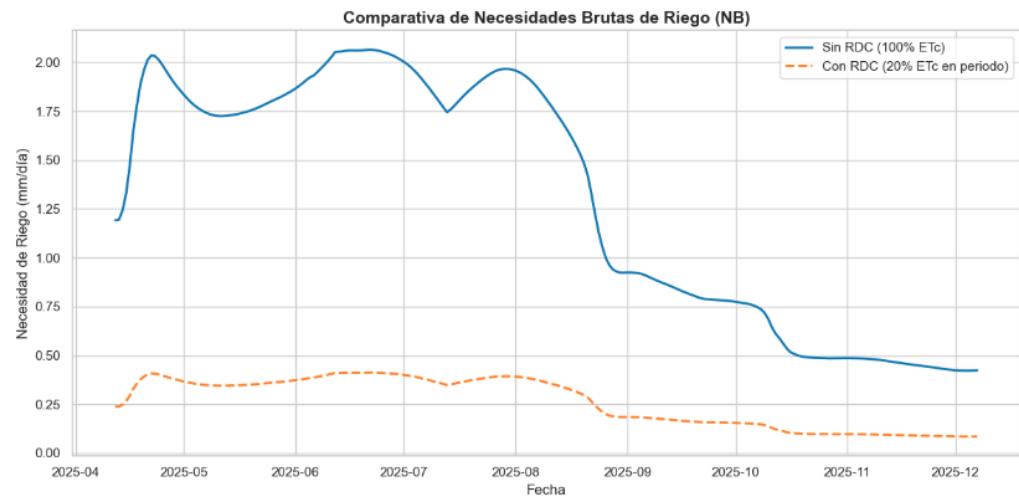


Figura 5.9: Necesidad de riego a lo largo del periodo predicho con y sin RDC

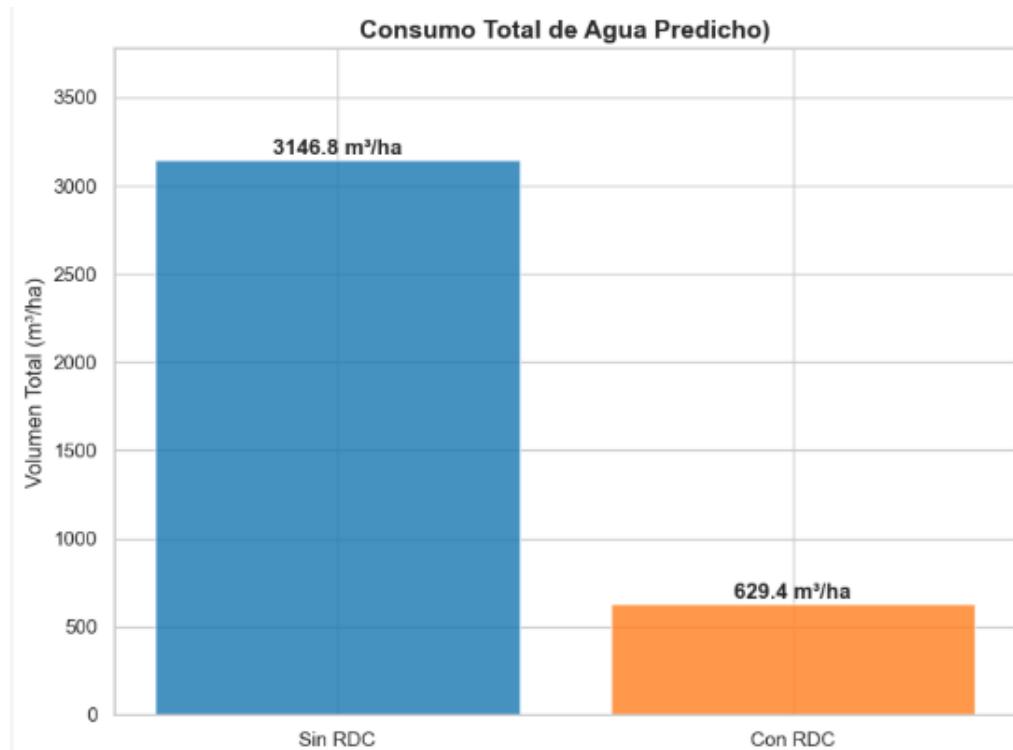


Figura 5.10: Comparación consumo total de agua en el periodo predicho con y sin RDC

## Aplicación riego deficitario controlado

### 5.3. Desarrollo de la aplicación web

Originalmente, la aplicación web se concibió como una plataforma sencilla para la visualización y descarga de los resultados obtenidos mediante los modelos de predicción. No obstante, durante el desarrollo de los fundamentos técnicos, se identificó la oportunidad de ampliar su funcionalidad, permitiendo el entrenamiento y la ejecución de predicciones directamente desde la interfaz. Este enfoque incrementa la accesibilidad y utilidad práctica del sistema, convirtiendo a la aplicación web en un componente central del proyecto.

La implementación de esta versión avanzada supuso un incremento en la complejidad técnica y una gestión de tiempos más exigente, lo que derivó en diversos desafíos que debieron resolverse durante la fase de desarrollo.

#### Desarrollo de la ruta de predicción

Se priorizó el desarrollo de la funcionalidad de predicción sobre la de entrenamiento. Esta decisión se fundamentó en la limitación temporal del proyecto, con el objetivo inicial de cargar modelos previamente entrenados en el entorno de desarrollo. De este modo, la implementación de la ruta de entrenamiento se planteó como una fase posterior supeditada a la disponibilidad de tiempo.

#### Fallo en la serialización de los modelos

El primer obstáculo técnico surgió al intentar cargar los modelos serializados mediante la herramienta `joblib` [24], los cuales habían sido generados en un entorno de desarrollo distinto. El sistema no lograba reconstruir los objetos debido a la falta de referencias a clases y funciones específicas en el nuevo contexto del servidor.

Como solución, se implementó en el *backend* una ruta de entrenamiento que permitió adaptar e integrar el código desarrollado en los Jupyter notebooks a la aplicación Flask. Durante esta etapa, se priorizó la lógica del servidor, posponiendo el desarrollo de la interfaz de usuario correspondiente.

#### Insuficiencia de memoria en el contenedor Docker

Al retomar la funcionalidad de predicción, se detectaron errores de ejecución relacionados con la gestión de recursos. Mediante la monitorización

del contenedor Docker [13], se confirmó que la carga sucesiva de modelos en la memoria RAM saturaba la capacidad asignada, provocando bloqueos en el sistema.



Figura 5.11: Monitorización de docker RAM

Para solventar esta limitación, se modificó la configuración en el archivo `docker-compose.yml` [12], incrementando la memoria RAM y la memoria de intercambio (*swap*) asignada al contenedor. Asimismo, se ajustaron los límites globales de Docker, tal como se detalla en la [Figura 5.12](#).

```

services:
  web:
    build: .
    container_name: agro_data
    ports:
      - "5000:5000"
    volumes:
      - ./:/app

    # Configuración directa de recursos
    mem_limit: 10g
    memswap_limit: 18g
    # Para CPUs (opcional)
    cpus: '2.0'
    cpu_shares: 512

```

Figura 5.12: Configuración recursos docker-compose.yml

## Creación de rutas de seguimiento de progreso

Dada la naturaleza prolongada de los procesos de entrenamiento y predicción, la interfaz generaba la impresión de haberse detenido durante su ejecución. Para mejorar la experiencia de usuario, se diseñó una página de seguimiento en tiempo real para ambas operaciones.

La primera aproximación consistió en una clase registradora de estados almacenada en la sesión de Flask, accesible mediante un endpoint API consultado periódicamente desde el frontend con JavaScript. Sin embargo, este enfoque no funcionó como se esperaba: el endpoint no reflejaba las actualizaciones del progreso generadas en procesos paralelos.

## Reestructuración del formato de almacenamiento del progreso

Tras revisar la documentación, se identificó que la sesión de Flask no está diseñada para acceso concurrente, lo que explicaba el comportamiento observado. Entre varias alternativas, se optó por almacenar el estado del progreso en una estructura externa a la sesión, en lugar de implementar un sistema basado en hilos, por su menor complejidad y mayor robustez.

### Reestructuración del formato de almacenamiento de resultados

Finalmente, se encontró un último problema al intentar guardar en sesión los resultados de la predicción, que incluían gráficos y superaban el límite de 4KB impuesto por Flask [15].

Se adoptó una solución análoga a la del seguimiento de progreso: los resultados se almacenaron en una estructura de datos independiente, mejor adaptada para gestionar volúmenes mayores de información y garantizar su persistencia y disponibilidad.

```

2  PREDICTION_PROGRESS = {}
1
3  PREDICTION_RESULTS = {}
```

Figura 5.13: Estructuras de almacenamiento de progreso y resultados

## Despliegue y Distribución de Servicios

El despliegue de los servicios desarrollados constituye una fase crítica del proyecto, cuyo objetivo primordial es garantizar la accesibilidad de la aplicación en entornos preconfigurados que permitan una ejecución inmediata (*out-of-the-box*).

### Arquitectura de Contenedores

Bajo esta premisa, la arquitectura del sistema se ha cimentado desde su concepción sobre el uso de contenedores mediante **Docker**. Esta estrategia de virtualización a nivel de sistema operativo asegura la portabilidad y la consistencia entre los entornos de desarrollo y producción.

Adicionalmente, se ha desarrollado un contenedor específico para el entorno de *Jupyter Notebook* [34]. El propósito de este componente es exponer el servicio a través de una técnica de *tunneling* [9], eliminando la necesidad de realizar instalaciones de dependencias para poder ejecutar este.

### Análisis de Alternativas de Hosting

Una vez consolidados los servicios, se evaluaron diversas estrategias para la exposición de los mismos a la red pública:

- **Despliegue en Máquinas Virtuales (VPS):** Se analizó la migración de los contenedores a servicios de infraestructura en la nube (tales como Oracle Cloud, Railway o Hostinger). No obstante, esta opción fue descartada debido a que los planes gratuitos o básicos no disponen de los recursos de computación (CPU y RAM) mínimos para un rendimiento óptimo. Por otro lado, el escalado a planes de nivel empresarial supone un coste económico inasumible para el alcance de este trabajo.
- **Exposición mediante Tunneling y Docker:** Finalmente, se optó por un enfoque basado en la apertura de túneles hacia los puertos de ejecución de los servicios, en concreto, con el proveedor **Cloudflare** [9]. Si bien este método puede comprometer la seguridad si se realiza de forma directa sobre el host, el uso de Docker permite aislar el servicio del sistema operativo base. De este modo, el acceso se restringe al entorno estanco del contenedor, mitigando vulnerabilidades sobre el host. Se considera este enfoque como el idóneo para la demostración ante el tribunal, aunque se reconoce su limitación en escenarios de escalado productivo masivo.

### Seguridad en el Despliegue (DevSecOps)

Con el fin de robustecer el sistema, se han integrado principios de **DevSecOps** centrados en la securización de las imágenes. Se han seguido las recomendaciones propuestas en la guía **OWASP Docker Top 10** [32], aplicando prácticas tales como la gestión de usuarios no privilegiados dentro del contenedor.



---

## 6. Trabajos relacionados

---

### 6.1. Trabajos Relacionados

El desarrollo de sistemas de apoyo a la decisión para el riego ha evolucionado desde modelos teóricos basados en balances hídricos manuales hacia plataformas integrales que combinan sensores en tiempo real e inteligencia artificial. En esta sección se analizan tres referentes del sector que comparten sinergias con el modelo propuesto para la finca 'Canduela', permitiendo contextualizar el valor añadido de este TFG.

#### **SIAR-PR (Servicio Integral de Asesoramiento al Regante) [19]**

Esta herramienta, desarrollada en Castilla-La Mancha, se centra en la aplicación de estrategias de Riego Deficitario Controlado (RDC) para cultivos leñosos como el almendro y la vid.

**Funcionamiento:** Utiliza un balance diario de agua en el suelo basado en datos climáticos locales, características del suelo y la fenología del cultivo. Permite ajustar el riego según la disponibilidad hídrica, priorizando las etapas donde el cultivo es menos sensible al estrés hídrico.

**Fortalezas:** Su principal valor es la especificidad regional. Está calibrado para las condiciones semiáridas de España y destaca por su simplicidad, permitiendo al agricultor obtener programaciones de riego sin necesidad de sensores complejos en parcela.

**Diferencia con el proyecto:** Mientras que SIAR-PR es un modelo de asesoramiento basado en balances hídricos estándar, este TFG apuesta por un enfoque *Data-Driven* mediante el uso de dispositivos IoT específicos en la parcela, permitiendo una personalización del riego mucho más granular y adaptada al microclima de la finca.

## CropX [10]

CropX es una de las plataformas comerciales más avanzadas a nivel global, centrada en la optimización del uso del agua.

**Funcionamiento:** Ofrece un ecosistema integral que incluye sensores de humedad del suelo, mapas de variabilidad de campo y monitorización de la Evapotranspiración Real ( $ET_a$ ). Su sistema proporciona recomendaciones precisas sobre cuándo y cuánto regar.

**Fortalezas:** Sobresale por su capacidad de Riego de Tasa Variable (VRI) [40] y su automatización. Permite adaptar el riego a la topografía y tipo de suelo de diferentes zonas de una misma parcela, actuando de forma proactiva antes de que la planta muestre signos de estrés.

**Diferencia con el proyecto:** CropX es una solución empresarial de gran escala que requiere una infraestructura de sensores propietaria cerrada. Este trabajo se diferencia al proponer un modelo predictivo semanal abierto, programado en Python, y diseñado específicamente para la gestión de políticas RDC en una explotación concreta.

## Arable [4]

Arable representa el estado del arte en la captura de datos climáticos y de la planta *in-situ* a través de su dispositivo multiespectral Arable Mark.

**Funcionamiento:** Utiliza un sensor 'todo en uno' que mide la lluvia, radiación solar, humedad de la hoja y, mediante espectrometría, el Índice de Área Foliar (LAI) [41] y la temperatura del dosel vegetal.

**Fortalezas:** Su punto fuerte es la fusión de datos meteorológicos con la respuesta fisiológica de la planta. Al medir directamente la temperatura del dosel, puede detectar el estrés hídrico metabólico antes de que el suelo se agote o la planta sufra daños visibles.

**Diferencia con el proyecto:** Arable se centra en el hardware de alta precisión y la visión espectral. Este TFG tiene un alcance y precisión mucho menor, sin embargo, no depende de dispositivos espectrales costosos.



---

## 7. Conclusiones y Líneas de trabajo futuras

---

En resumen, el objetivo principal de este trabajo se ha cumplido de forma satisfactoria. Se ha validado la viabilidad de aplicar un sistema de predicción de riego junto con la implementación de un sistema de riego deficitario controlado. A partir de datos crudos de sensores agrícolas, se ha logrado procesarlos y aplicar algoritmos de minería de datos para obtener predicciones que, combinadas con conocimiento agrícola, determinan la cantidad de riego óptima para las plantas.

La implementación metodológica y técnica del sistema se ejecutó exitosamente, aunque los resultados obtenidos no alcanzaron los niveles de efectividad esperados. Este hecho puede atribuirse principalmente a dos limitaciones prácticas que escapaban del alcance del proyecto:

- La incapacidad de entrenar los modelos con ciclos completos de un año debido a las limitaciones hardware disponibles.
- La escasez de datos históricos suficientes para construir modelos más robustos.

Adaptarme al área de minería de datos supuso un reto considerable, donde los conocimientos adquiridos en la asignatura correspondiente fueron de gran ayuda. Sin embargo, la integración de modelos fuera del ecosistema de scikit-learn representó un desafío tecnológico adicional que requirió investigación y adaptación constante.

En conjunto, pese a los contratiempos en los resultados cuantitativos, el proyecto se concluyó con éxito considerando las herramientas y recursos

disponibles. Incluso se superaron las expectativas iniciales al añadir funcionalidades adicionales como la aplicación web y generar una estructura de predicción escalable para desarrollos futuros. Este trabajo no solo me ha permitido aplicar lo aprendido durante la carrera, sino también explorar nuevas áreas que han ampliado mis conocimientos y reforzado mi interés por el desarrollo de aplicaciones, demostrando que un enfoque riguroso puede producir avances valiosos más allá de las métricas iniciales.

## 7.1. Líneas de trabajo futuras

### Integración en el proyecto smartcrops

La principal línea de continuidad se establece mediante la integración de la estructura de entrenamiento y predicción en el proyecto **SmartCrops 5.1** [38] en el que participa la **Universidad de Burgos (UBU)** entre más organizaciones del mundo agroalimentario 7.1 y académico 7.2.



Figura 7.1: Socios empresariales colaboradores del proyecto



Figura 7.2: Centros de investigación colaboradores del proyecto

Este proyecto surge para mitigar el riesgo de desertificación y el elevado estrés hídrico en España (que consume el 70 % del agua en agricultura),

impulsando la transformación digital del sector. Su objetivo es alcanzar una producción agrícola tecnificada y resiliente mediante el uso de *IA*, *Big Data* y **riego inteligente**, optimizando el consumo de recursos y aplicando prácticas regenerativas que garanticen la sostenibilidad climática hacia el horizonte 2050 [39].

Mi proyecto podría aprovechar las estructuras de recogida de datos y el ecosistema software ya disponibles dentro del proyecto aportando la lógica de predicción de necesidades de riego y adaptarlo a otro tipo de cultivos leñosos trabajando sobre las tareas del plan de trabajo propuesto en la web de **SmartCrops** [39]:

- **Tarea 1.4:** Digitalización y optimización tecnológica para la implantación de herramientas para la toma de decisiones que faciliten la sostenibilidad de los sistemas de cultivo bajo las condiciones de cambio climático. Asimilación de observaciones de la realidad en gemelos digitales de los cultivos.
- **Tarea 1.5:** Indicadores de sostenibilidad en sistemas de producción de cultivos sometidos a condiciones de cambio climático y estrés hídrico. Integración en gemelos digitales.

### Lineas de trabajo futuras adicionales

Adicionalmente, se identifican varias líneas de desarrollo futuro fuera del marco de la universidad que permitirían mejorar y ampliar el sistema implementado:

1. **Integración de predicción meteorológica a corto plazo:** Incorporar datos de predicción de la AEMET (Agencia Estatal de Meteorología) en tiempo real permitiría ajustar las recomendaciones de riego según las precipitaciones previstas, mejorando la precisión del sistema y optimizando el uso del agua.
2. **Configuración flexible de modelos de predicción:** Actualmente el sistema está optimizado para los datos específicos de nuestra finca de almendros. Esta línea propone desarrollar una arquitectura adaptable que pueda configurarse para cualquier dataset procedente de distintas fincas y cultivos, permitiendo la personalización de variables, algoritmos y preprocesamiento según las necesidades de cada caso.

3. **Entrenamiento distribuido en clúster computacional:** Migrar el proceso de entrenamiento a una infraestructura de mayor capacidad computacional que permita utilizar conjuntos de datos más extensos y ciclos de entrenamiento más complejos.
4. **Análisis explicativo con modelos de IA avanzada:** Integrar APIs de inteligencia artificial para generar explicaciones naturales de las predicciones y recomendaciones contextualizadas basadas en patrones detectados.

---

## Bibliografía

---

- [1] Agrozapiens. La revolución de la agricultura 5.0: Innovación y sostenibilidad para el futuro. <https://agrozapiens.com/la-revolucion-de-la-agricultura-5-0-innovacion-y-sostenibilidad-para-el-futuro/>, 2025. Visitado el 8 de febrero de 2026.
- [2] Richard G. Allen, Luis S. Pereira, Dirk Raes, and Martin Smith. *Crop Evapotranspiration: Guidelines for Computing Crop Water Requirements*. Number 56 in FAO Irrigation and Drainage Paper. Food and Agriculture Organization of the United Nations, Rome, 1998.
- [3] David J Anderson. *Kanban: successful evolutionary change for your technology business*. Blue hole press, 2010.
- [4] Arable Labs Inc. Arable: The global leader in crop intelligence, s.f. Accedido el 9 de febrero de 2026.
- [5] Armin Ronacher. Flask: A lightweight wsgi web application framework. <https://flask.palletsprojects.com/>, 2026. Accedido: 9 de enero de 2026.
- [6] Atlassian Corporation Plc. Jira: Project management and issue tracking software. <https://www.atlassian.com/software/jira>, 2026. Accedido: 9 de enero de 2026.
- [7] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [8] Cesens. Censens – plataforma de análisis y visualización de datos, 2024. Consultado el 10 de octubre de 2024.

- [9] Cloudflare Documentation. Cloudflare tunnel: Connect your resources to cloudflare, 2024. Accedido el 9 de febrero de 2026.
- [10] CropX Technologies. Irrigation planning: Use every drop wisely, s.f. Accedido el 9 de febrero de 2026.
- [11] DeepSeek-AI. Deepseek-r1, deepseek-v3, 2026. Consultado entre mayo de 2025 y febrero de 2026.
- [12] Docker Documentation. Compose file deployment reference. Accedido el 8 de febrero de 2026.
- [13] Docker, Inc. Docker: Empowering app development. <https://www.docker.com/>, 2026. Accedido: 9 de enero de 2026.
- [14] Ahmed M. Elshewey, Mahmoud Y. Shams, Abdelghafar M. Elhady, Samaa M. Shohieb, Abdelaziz A. Abdelhamid, Abdelhameed Ibrahim, and Zahraa Tarek. A novel wd-sarimax model for temperature forecasting using daily delhi climate dataset. *Sustainability*, 15(1), 2023.
- [15] Flask-Session Development Team. Flask-session: Adds server-side session support to your flask application, 2024. Accedido el 8 de febrero de 2026.
- [16] GetCrops. Penta – variedad de almendro pentasílabo, 2023. Descripción de las etapas fenológicas y enológicas de la variedad Penta de almendro.
- [17] Git Development Community. Git: Distributed version control system. <https://git-scm.com/>, 2026. Accedido: 9 de enero de 2026.
- [18] GitHub, Inc. Github: Where the world builds software. <https://github.com/>, 2026. Accedido: 9 de enero de 2026.
- [19] Juan D González-Teruel, Victor Blanco, Pedro José Blaya-Ros, Rafael Domingo, Fulgencio Soto Valles, and Roque Torres-Sánchez. Estimación del nivel de estrés hídrico en frutales mediante técnicas machine learning para aplicación en sistemas de riego inteligentes. 2021.
- [20] Google. Gemini (versión 3), 2026. Consultado entre mayo de 2025 y febrero de 2026.
- [21] Google Maps. Coordenadas geográficas 41°48'46.7"N 4°34'43.5"W. <https://www.google.com/maps/place/41°48'46.7%22N+4°34'43.5%22W/@41.8129615,-4.580375,1150m/data=!3m2!1e3!4b1!4m4!3m3!8m2!3d41.812959!4d-4.578752>. Consultado el 8 de febrero de 2026.

- [22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [23] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.
- [24] Joblib Development Team. Joblib: running python functions as pipeline jobs. <https://pypi.org/project/joblib/>, 2024. Accedido el 8 de febrero de 2026.
- [25] Federico Dicenta López-Higuera, José Egea Caballero, Teresa Cremades, Pedro Martínez Gómez, Encarnación Ortega Pastor, Manuel Rubio Angulo, R Sánchez-Pérez, PJ Martínez García, and Jesús López Alcolea. Makako: una variedad de almendro del cebas-csic de floración extra-tardía y auto-compatible. *Revista de fruticultura*, (58):6–11, 2017.
- [26] MacTeX Team. Mactex: Latex distribution for macos. <http://www.tug.org/mactex/>, 2026. Accedido: 9 de enero de 2026.
- [27] Microsoft Corporation. Visual studio code. <https://code.visualstudio.com/>, 2026. Accedido: 9 de enero de 2026.
- [28] Taiichi Ohno. *Toyota production system: beyond large-scale production*. Productivity press, 2019.
- [29] Christopher Olah et al. Understanding lstm networks.
- [30] OpenAI. Chatgpt (versiónes gpt-4.1 / mini, gpt-4o, gpt-5 mini / instant, gpt-5 mini), 2026. Consultado entre mayo de 2025 y febrero de 2026.
- [31] Overleaf, Inc. Overleaf: Online latex editor. <https://www.overleaf.com/>, 2026. Accedido: 9 de enero de 2026.
- [32] OWASP Foundation. Docker security project (docker top 10). <https://github.com/OWASP/Docker-Security>, 2023. Accedido el: 4 de febrero de 2025.
- [33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python, 2011. Versión 1.3.2.
- [34] Project Jupyter. Jupyter notebook. <https://jupyter.org/>, 2026. Accedido: 9 de enero de 2026.

- [35] Python Software Foundation. Python programming language. <https://www.python.org/>, 2026. Accedido: 9 de enero de 2026.
- [36] Pascual Romero. Análisis económico del cultivo de almendro en riego deficitario controlado (rdc). *imida. es*, 2005.
- [37] Ken Schwaber and Jeff Sutherland. La guía de scrum. *Scrumguides. Org*, 1:21, 2013.
- [38] SmartCrops 5.1. Nuevas herramientas digitales, tecnológicas y agroecológicas para la producción sostenible y resiliente de cultivos leñosos y hortícolas en climas semiáridos en condiciones de cambio climático. <https://smartcrops.es/>, 2024. Accedido: 9 de febrero de 2026.
- [39] SmartCrops 5.1. Retos y objetivos: Transformación digital y desarrollo sostenible en el sector agrario, 2024. Accedido: 9 de febrero de 2026.
- [40] Universidad de Minnesota. Riego de tasa variable, s.f. Accedido el 9 de febrero de 2026.
- [41] Wikipedia. Índice de área foliar — wikipedia, la enciclopedia libre, 2025. [Internet; descargado 4-diciembre-2025].