



Optimización con inteligencia artificial del despacho de riego solar en cultivos de pistacho

Ing. Tomás Corteggiano

Carrera de Especialización en Inteligencia Artificial

Director: Dr. Ing. Carlos Larisson

Jurados:

Mg. Ing. José Luis Hernández (UNRC)
Dr. Ing. David De Yong (UNRC)
Mg. Ing. Sebastián Tosco (UNRC)

Ciudad de Mendoza, noviembre de 2025



Optimización con inteligencia artificial del despacho de riego solar en cultivos de pistacho

Ing. Tomás Corteggiano

Carrera de Especialización en Inteligencia Artificial

Director: Dr. Ing. Carlos Larisson

Jurados:

Mg. Ing. José Luis Hernández (UNRC)
Dr. Ing. David De Yong (UNRC)
Mg. Ing. Sebastián Tosco (UNRC)

Ciudad de Mendoza, abril de 2026

Índice general

| | |
|---|----|
| Resumen | 1 |
| 1. Introducción | 1 |
| 1.1. Contexto y problemática del riego solar | 1 |
| 1.2. Estado del arte: modelado y optimización | 2 |
| 1.2.1. El modelado: consistencia vs. precisión | 2 |
| 1.2.2. Optimización: optimalidad vs. factibilidad | 3 |
| 1.3. Motivación | 4 |
| 1.4. Objetivos y alcance | 5 |
| 2. Introducción específica | 7 |
| 2.1. Teoría del modelado híbrido: corrección de residuales | 7 |
| 2.1.1. Componente físico ($f_{físico}$) | 8 |
| 2.1.2. Componente corrector (f_{AI}) | 8 |
| 2.2. Formulación del problema de optimización | 9 |
| 2.2.1. Formulación matemática formal | 9 |
| 2.2.2. Función objetivo ponderada | 9 |
| 2.2.3. Restricciones del problema | 10 |
| 2.3. Teoría de optimización: algoritmos genéticos | 10 |
| 2.3.1. Codificación del cromosoma (representación de la solución) | 10 |
| 2.3.2. Operadores de selección | 11 |
| 2.3.3. Operadores de cruce (crossover / recombinación) | 11 |
| 2.3.4. Operadores de mutación (diversidad) | 12 |
| 2.4. Manejo de restricciones: funciones de penalización | 12 |
| 3. Diseño e implementación | 15 |
| 3.1. Arquitectura general del software | 15 |
| 3.1.1. Subsistema de entrenamiento y ciclo de vida (MLOps) | 15 |
| 3.1.2. Subsistema de optimización y servicio | 16 |
| 3.2. Implementación del gemelo digital híbrido | 18 |
| 3.3. Implementación del optimizador de despacho (GA) | 18 |
| 3.4. Implementación de funciones de costo y penalizaciones | 18 |
| 4. Ensayos y resultados | 19 |
| 4.1. Entorno y banco de pruebas | 19 |
| 4.2. Pruebas del gemelo digital híbrido | 19 |
| 4.3. Caso de estudio: simulación de despacho | 19 |
| 5. Conclusiones | 21 |
| 5.1. Resultados obtenidos | 21 |
| 5.2. Trabajo futuro | 21 |
| Bibliografía | 23 |

Índice general

| | |
|---|----|
| Resumen | 1 |
| 1. Introducción | 1 |
| 1.1. Contexto y problemática del riego solar | 1 |
| 1.2. Estado del arte: modelado y optimización | 2 |
| 1.2.1. El modelado: consistencia vs. precisión | 2 |
| 1.2.2. Optimización: optimalidad vs. factibilidad | 3 |
| 1.3. Motivación | 4 |
| 1.4. Objetivos y alcance | 5 |
| 2. Introducción específica | 7 |
| 2.1. Teoría del modelado híbrido: corrección de residuales | 7 |
| 2.1.1. Componente físico ($f_{físico}$) | 7 |
| 2.1.2. Componente corrector (f_{AI}) | 8 |
| 2.2. Formulación del problema de optimización | 8 |
| 2.2.1. Formulación matemática formal | 8 |
| 2.2.2. Función objetivo ponderada | 8 |
| 2.2.3. Restricciones del problema | 9 |
| 2.3. Teoría de optimización: algoritmos genéticos | 9 |
| 2.3.1. Codificación del cromosoma (representación de la solución) | 9 |
| 2.3.2. Operadores evolutivos | 10 |
| 2.4. Herramientas computacionales y métricas | 10 |
| 2.4.1. Bibliotecas y entorno de desarrollo | 10 |
| 2.4.2. Métricas de evaluación | 11 |
| 3. Diseño e implementación | 13 |
| 3.1. Arquitectura general del software | 13 |
| 3.1.1. Subsistema de entrenamiento y ciclo de vida (MLOps) | 13 |
| 3.1.2. Subsistema de optimización y servicio | 14 |
| 3.2. Implementación del gemelo digital híbrido | 16 |
| 3.2.1. Construcción y parametrización del modelo físico | 16 |
| 3.2.2. Ingeniería de datos y modelo de corrección | 17 |
| 3.2.3. Ciclo de vida y gestión de artefactos (MLOps) | 18 |
| 3.2.4. Orquestación y ejecución en el servicio de inferencia | 18 |
| 3.3. Implementación del optimizador de despacho (GA) | 19 |
| 3.4. Implementación de funciones de costo y penalizaciones | 19 |
| 4. Ensayos y resultados | 21 |
| 4.1. Entorno y banco de pruebas | 21 |
| 4.2. Pruebas del gemelo digital híbrido | 21 |
| 4.3. Caso de estudio: simulación de despacho | 21 |
| 5. Conclusiones | 23 |

IV

| | |
|---------------------------|----|
| 5.1. Resultados obtenidos | 23 |
| 5.2. Trabajo futuro | 23 |

Bibliografía

25

Índice de figuras

- | | |
|---|----|
| 1.1. Diagrama conceptual de los paradigmas de modelado: caja blanca (física), caja negra (IA pura) y caja gris (híbrido). | 3 |
| 3.1. Diagrama de capas del sistema de riego, donde el foco del trabajo es la capa de gestión y optimización. | 17 |

Índice de figuras

- | | |
|--|----|
| 1.1. Diagrama conceptual de los paradigmas de modelado: caja blanca (física), caja negra (IA pura) y caja gris (híbrido). | 3 |
| 3.1. Diagrama de capas del sistema de riego, donde el foco del trabajo es la capa de gestión y optimización. | 15 |
| 3.2. Esquema del gemelo digital híbrido: integración del pipeline de entrenamiento (MLOps), el modelo físico WNTR y el corrector de IA en el servicio de inferencia. | 19 |

Índice de tablas

| | |
|--|----|
| 2.1. Nomenclatura del modelo de optimización. | 9 |
| 2.2. Análisis comparativo de operadores de selección en AG. | 11 |
| 2.3. Análisis comparativo de operadores de cruce relevantes. | 12 |

Índice de tablas

| | |
|---|---|
| 2.1. Nomenclatura del modelo de optimización. | 8 |
|---|---|

(NL), que proviene directamente de la función de evaluación. El modelo hidráulico que predice el caudal y la presión, ya sea de caja blanca o negra, es una función compleja de la física de fluidos, no una simple suma algebraica.

La combinación de decisiones binarias (un espacio de búsqueda combinatorio) y una función de evaluación no lineal clasifica este problema en la categoría de complejidad NP-hard [17]. Esto significa que es computacionalmente intratable: el tiempo requerido para encontrar la solución óptima garantizada crece exponencialmente a medida que el problema escala (más válvulas o más intervalos). El espacio de soluciones a explorar es demasiado grande.

Esta realidad computacional genera un conflicto directo e irresoluble con las necesidades de un sistema de riego operativo. El estado del arte, si bien ofrece solvers exactos para MINLP que buscan el óptimo global, no puede garantizar el tiempo de convergencia.

Dado este compromiso entre la optimalidad y la factibilidad, la práctica establecida para problemas de *scheduling* (programación) y asignación de recursos NP-hard es el uso de metaheurísticas [18, 19]. Algoritmos como los algoritmos genéticos (AG) proponen un enfoque alternativo: en lugar de buscar una garantía de optimalidad global, ofrecen una garantía de tiempo de ejecución. Están diseñados para encontrar soluciones de alta calidad (casi óptimas) dentro de un presupuesto computacional fijo [20]. Para un problema de ingeniería en un entorno productivo, esta capacidad de obtener una solución factible en un tiempo acotado es un requisito operativo fundamental, lo que convierte a esta estrategia en un enfoque viable.

1.3. Motivación

La motivación de este trabajo es proponer una arquitectura de software **unificada** que resuelva los dos dilemas fundamentales identificados en el estado del arte. **Este proyecto** se fundamenta en la hipótesis de que es posible lograr tanto la precisión en el modelado como la factibilidad en la optimización, sin **sacrificar** la robustez física ni el rendimiento operativo. Ante el desafío del modelado, que obliga a elegir entre la consistencia de la caja blanca y la precisión de la caja negra, este trabajo implementa un enfoque de caja gris o Gemelo Digital Híbrido. Esta arquitectura utiliza el simulador físico (basado en EPANET/WNTR) [21] para obtener una línea de base robusta y físicamente coherente. Adicionalmente, emplea un modelo de inteligencia artificial (IA) no para modelar el sistema completo, sino con la tarea específica de predecir y corregir el error residual del modelo físico [22, 23]. De esta forma, se busca unificar la consistencia física de los primeros principios con la precisión empírica de los modelos de **IA**.

El desafío de la optimización (MINLP) también requiere un enfoque específico. Si bien los solvers exactos son la herramienta fundamental para garantizar la optimalidad global, su tiempo de convergencia es, por la naturaleza NP-hard del problema, indefinido. Un cálculo que puede tardar horas o días no es una herramienta útil para la toma de decisiones agronómicas diarias. Por esta razón, **el proyecto** encontró un enfoque alternativo que prioriza la velocidad de respuesta. Se seleccionó una metaheurística, el algoritmo genético, fundamentada en su probada robustez para problemas de programación complejos [18]. Esta estrategia permite al sistema encontrar soluciones de alta calidad (casi óptimas) dentro

(NL), que proviene directamente de la función de evaluación. El modelo hidráulico que predice el caudal y la presión, ya sea de caja blanca o negra, es una función compleja de la física de fluidos, no una simple suma algebraica.

La combinación de decisiones binarias (un espacio de búsqueda combinatorio) y una función de evaluación no lineal clasifica este problema en la categoría de complejidad NP-hard [17]. Esto significa que es computacionalmente intratable: el tiempo requerido para encontrar la solución óptima garantizada crece exponencialmente a medida que el problema escala (más válvulas o más intervalos). El espacio de soluciones a explorar es demasiado grande.

Esta realidad computacional genera un conflicto directo e irresoluble con las necesidades de un sistema de riego operativo. El estado del arte, si bien ofrece solvers exactos para MINLP que buscan el óptimo global, no puede garantizar el tiempo de convergencia.

Dado este compromiso entre la optimalidad y la factibilidad, la práctica establecida para problemas de *scheduling* (programación) y asignación de recursos NP-hard es el uso de metaheurísticas [18, 19]. Algoritmos como los algoritmos genéticos (AG) proponen un enfoque alternativo: en lugar de buscar una garantía de optimalidad global, ofrecen una garantía de tiempo de ejecución. Están diseñados para encontrar soluciones de alta calidad (casi óptimas) dentro de un presupuesto computacional fijo [20]. Para un problema de ingeniería en un entorno productivo, esta capacidad de obtener una solución factible en un tiempo acotado es un requisito operativo fundamental, lo que convierte a esta estrategia en un enfoque viable.

1.3. Motivación

La motivación de este trabajo es proponer una arquitectura de software **unificada** que resuelva los dos dilemas fundamentales identificados en el estado del arte. **Esta solución** se fundamenta en la hipótesis de que es posible lograr tanto la **precisión** en el modelado como la **factibilidad** en la optimización, sin **sacrificar** la robustez física ni el rendimiento operativo.

Ante el desafío del modelado, que obliga a elegir entre la consistencia de la caja blanca y la precisión de la caja negra, **el objetivo es** unificar la consistencia física de los primeros principios con la precisión empírica de los modelos de **inteligencia artificial** (IA).

Por otro lado, la optimización (MINLP) también requiere un enfoque específico. Si bien los solvers exactos son la herramienta fundamental para garantizar la optimalidad global, su tiempo de convergencia es, por la naturaleza NP-hard del problema, indefinido. Un cálculo que puede tardar horas o días no es una herramienta útil para la toma de decisiones agronómicas diarias. Por esta razón, **se requiere** un enfoque alternativo que **priorice** la velocidad de **respuesta**. Esta estrategia **debe permitir** al sistema encontrar soluciones de alta calidad (casi óptimas) dentro de un límite de tiempo, lo que resulta **aceptable y práctico** para las **necesidades** de la operación real.

1.4. Objetivos y alcance

5

de un límite de tiempo fijo, lo que resulta aceptable y práctico para las necesidades de la operación real.

1.4. Objetivos y alcance

Basado en la problemática y la motivación expuestas, el objetivo general de este trabajo es diseñar, implementar y validar el software que constituye la capa de gestión y optimización para un sistema de riego solar. El proyecto se enfoca en desarrollar los dos componentes técnicos fundamentales que responden directamente a los dilemas identificados en el estado del arte.

El primer objetivo específico es la creación de un **Gemelo Digital Híbrido**. Esto implica implementar un simulador físico y, de forma crucial, desarrollar un modelo de inteligencia artificial cuya función es predecir y corregir el error residual de dicho modelo físico. Este componente servirá como la función de evaluación precisa y robusta para el optimizador.

El segundo objetivo específico es el diseño de un optimizador metaheurístico. Se implementará un AG capaz de navegar el complejo espacio de búsqueda **MINLP**. Este agente utilizará el Gemelo Digital Híbrido como su función de aptitud para generar cronogramas de despacho casi óptimos, priorizando la factibilidad temporal sobre la optimalidad exacta. El alcance de la implementación incluye también el desarrollo de la función de costos híbrida y unificada, que permite al agrónomo ponderar los objetivos de uniformidad, eficiencia y sostenibilidad del acuífero.

Es importante delimitar que el alcance de este trabajo es exclusivamente de software y se centra en el motor de decisión. El **proyecto** no incluye el diseño, **provisión**, instalación o mantenimiento de ningún componente de hardware (como bombas, sensores, PLCs o gateways). Asimismo, el desarrollo de la interfaz de usuario **final** o capa de presentación (dashboard) y el soporte operativo **post-entrega** quedan fuera del alcance de esta memoria.

1.4. Objetivos y alcance

5

1.4. Objetivos y alcance

Basado en la problemática y la motivación expuestas, el objetivo general de este trabajo es diseñar, implementar y validar el software que constituye la capa de gestión y optimización para un sistema de riego solar. El proyecto se enfoca en desarrollar los dos componentes técnicos fundamentales que responden directamente a los dilemas identificados en el estado del arte.

El primer objetivo específico es la creación de un **gemelo digital híbrido**. Esto implica implementar un simulador físico y, de forma crucial, desarrollar un modelo de inteligencia artificial cuya función es predecir y corregir el error residual de dicho modelo físico. Este componente servirá como la función de evaluación precisa y robusta para el optimizador.

El segundo objetivo específico es el diseño de un optimizador metaheurístico. Se implementará un AG capaz de navegar el complejo espacio de búsqueda **MINLP**. Este agente utilizará el Gemelo Digital Híbrido como su función de aptitud para generar cronogramas de despacho casi óptimos, priorizando la factibilidad temporal sobre la optimalidad exacta. El alcance de la implementación incluye también el desarrollo de la función de costos híbrida y unificada, que permite al agrónomo ponderar los objetivos de uniformidad, eficiencia y sostenibilidad del acuífero.

Es importante delimitar que el alcance de este trabajo es exclusivamente de software y se centra en el motor de decisión. El **trabajo** no incluye el diseño, **provisión**, instalación o mantenimiento de ningún componente de hardware (como bombas, sensores, PLCs o gateways). Asimismo, el desarrollo de la interfaz de usuario **final** o capa de presentación (dashboard) y el soporte operativo **post-entrega** quedan fuera del alcance de esta memoria.

Capítulo 2

Introducción específica

En este capítulo se presentan los fundamentos teóricos que sustentan la arquitectura de optimización desarrollada. Se detalla el modelado híbrido de caja gris, la formulación matemática del problema como un MINLP, la teoría de los AG como solución metaheurística y el método de funciones de penalización para el manejo de restricciones.

2.1. Teoría del modelado híbrido: corrección de residuales

La simulación de sistemas físicos complejos, como lo es un sistema hidráulico, se aborda generalmente desde dos enfoques principales [24].

El primer enfoque es el modelado de caja blanca (*White-Box*), de naturaleza determinística, basado en leyes físicas de la mecánica de fluidos [24, 25]. El motor de simulación hidráulico EPANET es un ejemplo representativo de este enfoque [26, 27]. La ventaja principal de los modelos de caja blanca es su interpretabilidad y la garantía de predicciones físicamente consistentes. Sin embargo, los modelos que se basan únicamente en leyes físicas tienen una limitación fundamental: debido a simplificaciones, inevitablemente producen desviaciones significativas, o sesgos, en comparación con la realidad [28, 29]. Estos sesgos surgen de dinámicas no modeladas y de la incertidumbre en los parámetros intrínsecos, como la rugosidad exacta de las tuberías (el coeficiente C en la ecuación de Hazen-Williams) o las pérdidas de carga menores no contempladas en el modelo.

El segundo enfoque es el modelado de caja negra (*Black-Box*). Estos modelos son de naturaleza estocástica, como las redes neuronales (NN) o los perceptrones multicapa (MLP), que mapean entradas a salidas basándose puramente en datos observados [24, 25]. Su fortaleza es la capacidad de modelar relaciones no lineales extremadamente complejas sin necesidad de conocer el modelo físico del sistema [30]. Sin embargo, sus desventajas son críticas para sistemas de ingeniería. Primero, requieren voltímenes de datos de entrenamiento que a menudo son muy grandes [28, 29]. Segundo, y más importante, son propensos a generar predicciones que pueden ser físicamente inconsistentes, como violar la conservación de masa, un riesgo inaceptable para un sistema de control de recursos hídricos.

Como solución, en este proyecto se propone un modelado de caja gris (*Gray-Box*), un paradigma que busca combinar las fortalezas de ambos enfoques [30]. Específicamente, se implementa la arquitectura de corrección de residuales.

En este paradigma, el modelo de inteligencia artificial no intenta aprender la totalidad del sistema desde cero. En su lugar, el modelo físico ($f_{físico}$) proporciona

Capítulo 2

Introducción específica

En este capítulo se presentan los fundamentos teóricos y las herramientas que sustentan la arquitectura de optimización propuesta. Se detalla el modelado híbrido de caja gris, la formulación del problema MINLP, la configuración del algoritmo genético y las métricas de evaluación empleadas.

2.1. Teoría del modelado híbrido: corrección de residuales

La simulación de sistemas hidráulicos presenta un compromiso entre la consistencia física y la precisión operativa. Mientras que los modelos de caja blanca (determinísticos) como EPANET garantizan coherencia física pero sufren sesgos por incertidumbre paramétrica, los modelos de caja negra (estocásticos) ofrecen alta precisión pero carecen de robustez física [21, 22].

Para balancear estas limitaciones, este trabajo implementa un modelado de caja gris (*gray-box*) mediante una arquitectura de corrección de residuales [23]. En este esquema, el modelo físico ($f_{físico}$) aporta la línea base robusta, mientras que un modelo de inteligencia artificial (f_{AI}) predice y corrige el error sistemático o residual (R) del primero.

La formulación matemática de esta arquitectura híbrida se define en la ecuación 2.1:

$$f_{hybrid} = f_{físico} + f_{AI}(\mathbf{X}) \quad (2.1)$$

donde el objetivo de entrenamiento para f_{AI} es el residual $R = Y_{real} - Y_{físico}$.

2.1.1. Componente físico ($f_{físico}$)

El componente físico se desarrolla utilizando el motor de simulación EPANET [24, 25]. Este modelo se rige por leyes físicas claves que definen la complejidad del problema:

- Pérdida de carga por fricción: se utiliza la ecuación de Hazen-Williams (ecuación 2.2) [26], una relación empírica que define la pérdida de carga (H_f) en una tubería.

$$H_f = \frac{10,67 \cdot L}{C^{1,852} \cdot D^{4,87}} \cdot Q^{1,852} \quad (2.2)$$

- Comportamiento de emisores (goteros): el caudal de salida (q_e) se modela mediante la ecuación caudal-presión del emisor [27], una relación empírica de tipo potencia (ecuación 2.3) que vincula el caudal con la presión (p) en el nodo. En este trabajo se utiliza un exponente γ de 0,46, característico del

la predicción principal y la robustez, mientras que el modelo de IA (f_{AI}) tiene una tarea mucho más acotada y factible: aprender y predecir el error residual (el sesgo sistemático) del modelo físico [29].

La formulación matemática de esta arquitectura híbrida se muestra en la ecuación 2.1:

$$f_{hybrid} = f_{físico} + f_{AI}(X) \quad (2.1)$$

Donde el objetivo de entrenamiento (el *target*) para f_{AI} es el residual R , definido según la ecuación 2.2:

$$R = Y_{real} - Y_{físico} \quad (2.2)$$

La literatura académica lo describe como una herramienta de predicción eficiente [28, 29], donde el modelo $f_{físico}$ aporta la consistencia y el f_{AI} aporta la precisión, corrigiendo los sesgos propios del primero.

A continuación se analizan ambas partes de la ecuación 2.1.

2.1.1. Componente físico ($f_{físico}$)

El componente físico se desarrolla utilizando el motor de simulación EPANET [26, 27]. Este modelo se rige por leyes físicas claves que definen la complejidad del problema:

1. Pérdida de carga por fricción: se utiliza la ecuación de Hazen-Williams (ecuación 2.3) [31], una relación empírica que define la pérdida de carga (H_f) en una tubería.

$$H_f = \frac{10,67 \cdot L}{C^{1,852} \cdot D^{4,81}} \cdot Q^{1,852} \quad (2.3)$$

2. Comportamiento de emisores (goteros): el caudal de salida (q_e) se modela mediante la ecuación caudal-presión del emisor [32], una relación empírica de tipo potencia (ecuación 2.4) que vincula el caudal con la presión (p) en el nodo. En este trabajo se utiliza un exponente γ de 0,46, característico del equipamiento instalado, introduciendo otra no linealidad.

$$q_e = C_d \cdot p^\gamma \quad (2.4)$$

Además, se emplea la simulación en modo Análisis Dirigido por Presión (PDA) [33]. A diferencia del Análisis Dirigido por Demanda (DDA) [33], que asume que la demanda siempre se satisface, el PDA ajusta el caudal entregado si la presión en un nodo cae, lo que ofrece una simulación mucho más realista para un sistema de riego solar donde la presión de la bomba varía constantemente.

2.1.2. Componente corrector (f_{AI})

El componente de IA, f_{AI} , se implementa como un Perceptrón Multicapa (MLP).

El vector de características X está compuesto por variables que describen tanto el estado operativo del sistema como sus acciones de control. Entre las primeras se incluyen magnitudes medidas en campo, como la presión a la salida de la bomba y el nivel freático del acuífero; entre las segundas, las decisiones de operación, como la apertura o cierre de válvulas. Además, la salida generada por el modelo físico se incorpora explícitamente como una de las entradas del MLP. De este modo, la red neuronal aprende a estimar el sesgo sistemático del modelo hidráulico

equipamiento instalado, introduciendo otra no linealidad.

$$q_e = C_d \cdot p^\gamma \quad (2.3)$$

2.1.2. Componente corrector (f_{AI})

El componente de corrección se implementa mediante un perceptrón multicapa (MLP). Este recibe como entrada el estado operativo del sistema (presión de bomba, nivel freático y configuración de válvulas) junto con la salida del modelo físico, aprendiendo a estimar la diferencia entre la simulación teórica y los datos reales medidos.

2.2. Formulación del problema de optimización

Una vez que se dispone de una función de aptitud f_{hybrid} de alta fidelidad, es necesario definir formalmente el problema de optimización.

2.2.1. Formulación matemática formal

El problema de optimización del despacho de riego consiste en encontrar el cronograma de apertura y cierre de válvulas que maximice un conjunto de objetivos, sujeto a las restricciones físicas de la red hidráulica y la disponibilidad de recursos. En tabla 2.1, se presenta una nomenclatura y formulación general del modelo.

TABLA 2.1. Nomenclatura del modelo de optimización.

| Símbolo | Definición |
|----------------------|--|
| V | Matriz de decisiones de despacho (variable de decisión). |
| $v_{i,t}$ | Variable binaria: 1 si la válvula i está abierta en el tiempo t , 0 si está cerrada. |
| T | Horizonte de planificación (ej. 24 horas, discretizado en T intervalos). |
| I | Conjunto de todas las válvulas (sectores de riego). |
| $f_{hybrid}(V)$ | Función de aptitud. |
| $Q_{entregado,i}(V)$ | Caudal total entregado al sector i bajo el plan V . |
| $Q_{objetivo,i}$ | Caudal objetivo para el sector i . |
| $P_{consumido,t}(V)$ | Potencia consumida por la bomba en el tiempo t bajo el plan V . |
| $P_{disponible,t}$ | Potencia disponible pronosticada en el tiempo t . |
| $N_{recurso,t}(V)$ | Nivel del recurso en el tiempo t bajo el plan V . |
| N_{min} | Nivel mínimo de seguridad del recurso. |
| w_u, w_t, w_a | Ponderaciones estratégicas para los objetivos de la función. |

2.2.2. Función objetivo ponderada

Un enfoque común para problemas multiobjetivo es formular una función de aptitud unificada. Esta función es una suma ponderada que puede incluir objetivos agronómicos, de eficiencia y de sostenibilidad. Esta se muestra en la ecuación 2.4:

$$\max_V (w_u \cdot U(V) - w_t \cdot T_{bomba}(V) - w_a \cdot S_{acuífero}(V)) \quad (2.4)$$

2.2. Formulación del problema de optimización

9

en función de las condiciones reales de operación, actuando como un corrector residual que ajusta las predicciones físicas hacia valores más próximos a las observaciones experimentales.

2.2. Formulación del problema de optimización

Una vez que se dispone de una función de aptitud f_{hybrid} de alta fidelidad, es necesario definir formalmente el problema de optimización.

2.2.1. Formulación matemática formal

El problema de optimización del despacho de riego consiste en encontrar el cronograma de apertura y cierre de válvulas que maximice un conjunto de objetivos, sujeto a las restricciones físicas de la red hidráulica y la disponibilidad de recursos. En tabla 2.1, se presenta una nomenclatura y formulación general del modelo.

TABLA 2.1. Nomenclatura del modelo de optimización.

| Símbolo | Definición |
|-------------------------------|--|
| \mathbf{V} | Matriz de decisiones de despacho (variable de decisión). |
| $v_{i,t}$ | Variable binaria: 1 si la válvula i está abierta en el tiempo t , 0 si está cerrada. |
| \mathcal{T} | Horizonte de planificación (ej. 24 horas, discretizado en T intervalos). |
| \mathcal{I} | Conjunto de todas las válvulas (sectores de riego). |
| $f_{hybrid}(\mathbf{V})$ | Función de aptitud. |
| $Q_{entregado,i}(\mathbf{V})$ | Caudal total entregado al sector i bajo el plan \mathbf{V} . |
| $Q_{objetivo,i}$ | Caudal objetivo para el sector i . |
| $P_{consumida,t}(\mathbf{V})$ | Potencia consumida por la bomba en el tiempo t bajo el plan \mathbf{V} . |
| $P_{disponible,t}$ | Potencia disponible pronosticada en el tiempo t . |
| $N_{recurso,t}(\mathbf{V})$ | Nivel del recurso en el tiempo t bajo el plan \mathbf{V} . |
| N_{min} | Nivel mínimo de seguridad del recurso. |
| w_u, w_t, w_a | Ponderaciones estratégicas para los objetivos de la función. |

2.2.2. Función objetivo ponderada

Un enfoque común para problemas multiobjetivo es formular una función de aptitud unificada. Esta función es una suma ponderada que puede incluir objetivos agronómicos, de eficiencia y de sostenibilidad. Esta se muestra en la ecuación 2.5:

$$\max_{\mathbf{V}} (w_u \cdot U(\mathbf{V}) - w_t \cdot T_{bombeo}(\mathbf{V}) - w_a \cdot S_{acuífero}(\mathbf{V})) \quad (2.5)$$

Donde:

- $U(\mathbf{V})$: una métrica de uniformidad (ej. minimizar el déficit cuadrático $\sum(Q_{objetivo,i} - Q_{entregado,i})^2$), que se busca maximizar.
- $T_{bombeo}(\mathbf{V})$: una métrica del tiempo de bombeo (ej. $\sum v_{i,t}$), que se busca minimizar.

2.3. Teoría de optimización: algoritmos genéticos

9

donde:

- $U(\mathbf{V})$: una métrica de uniformidad, que se busca maximizar.
- $T_{bombeo}(\mathbf{V})$: una métrica del tiempo de bombeo, que se busca minimizar.
- $S_{acuífero}(\mathbf{V})$: una métrica del estrés sobre el acuífero, que se busca minimizar.

2.2.3. Restricciones del problema

La maximización está sujeta a un conjunto de restricciones físicas y operativas:

1. Restricción de potencia: la potencia consumida por la bomba no puede exceder la potencia disponible en ningún momento, de acuerdo a la ecuación 2.5.

$$P_{consumida,t}(f_{hybrid}(\mathbf{V})) \leq P_{disponible,t} \quad \forall t \in \mathcal{T} \quad (2.5)$$

2. Restricción de sostenibilidad (ej. nivel freático): el nivel del recurso no debe caer por debajo del límite de seguridad, según la ecuación 2.6.

$$N_{recurso,t}(f_{hybrid}(\mathbf{V})) \geq N_{min} \quad \forall t \in \mathcal{T} \quad (2.6)$$

3. Restricciones hidráulicas (implícitas): las leyes físicas de la red están encapsuladas dentro de la evaluación de la función f_{hybrid} .

4. Variables de decisión binarias: las decisiones de control de válvulas son binarias, como lo indica la ecuación 2.7.

$$v_{i,t} \in \{0, 1\} \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \quad (2.7)$$

2.3. Teoría de optimización: algoritmos genéticos

El algoritmo genético consiste en una búsqueda estocástica global basada en los principios de la evolución natural y la supervivencia del más apto [28, 29]. Este algoritmo es adecuado para navegar espacios de búsqueda complejos, no convexos y de alta dimensionalidad. Su uso acoplado a simuladores como EPANET para la optimización de redes de agua es una técnica documentada en la literatura [24, 25].

El AG opera sobre una población de cromosomas, donde cada cromosoma representa una solución candidata completa al problema de optimización.

2.3.1. Codificación del cromosoma (representación de la solución)

El cromosoma es la estructura de datos que codifica un plan de despacho de riego [30]. La elección de la codificación es fundamental para el rendimiento del algoritmo.

Se utiliza una codificación por valor binario (*value encoding*). El cromosoma es una cadena de valores (enteros, reales o binarios) donde cada gen representa un valor específico. Si el horizonte de planificación T se discretiza en N_T intervalos y hay N_V válvulas, un cromosoma C es un vector binario de longitud $N_V \times N_T$. Aunque

- $S_{acuifero}(V)$: una métrica del estrés sobre el acuífero (ej. la extracción total de agua), que se busca minimizar.

2.2.3. Restricciones del problema

La maximización está sujeta a un conjunto de restricciones físicas y operativas:

1. Restricción de potencia: la potencia consumida por la bomba no puede exceder la potencia disponible en ningún momento, según la ecuación 2.6.

$$P_{consumida,t}(f_{hybrid}(V)) \leq P_{disponible,t} \quad \forall t \in \mathcal{T} \quad (2.6)$$

2. Restricción de sostenibilidad (ej. nivel freático): el nivel del recurso no debe caer por debajo del límite de seguridad, según la ecuación 2.7.

$$N_{recurso,t}(f_{hybrid}(V)) \geq N_{min} \quad \forall t \in \mathcal{T} \quad (2.7)$$

3. Restricciones hidráulicas (implícitas): las leyes físicas de la red (Hazen-Williams, ley de potencia) están encapsuladas dentro de la evaluación de la función f_{hybrid} .
4. Variables de decisión binarias: las decisiones de control de válvulas son binarias, como lo indica la ecuación 2.8.

$$v_{i,t} \in \{0, 1\} \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \quad (2.8)$$

2.3. Teoría de optimización: algoritmos genéticos

El algoritmo genético consiste en una búsqueda estocástica global basada en los principios de la evolución natural y la supervivencia del más apto [34, 35]. Este algoritmo es adecuado para navegar espacios de búsqueda complejos, no convexos y de alta dimensionalidad. Su uso acoplado a simuladores como EPANET para la optimización de redes de agua es una técnica documentada en la literatura [26, 27].

El AG opera sobre una población de cromosomas, donde cada cromosoma representa una solución candidata completa al problema de optimización.

2.3.1. Codificación del cromosoma (representación de la solución)

El cromosoma es la estructura de datos que codifica un plan de despacho de riego [36]. La elección de la codificación es fundamental para el rendimiento del algoritmo.

- *Permutation encoding* (codificación por permutación): en este esquema, el cromosoma es una permutación de elementos. Se utiliza para problemas de secuenciación u ordenamiento, como el Problema del Viajante (TSP) [37].
- *Value encoding* (codificación por valor): en este esquema, el cromosoma es una cadena de valores (enteros, reales o binarios) donde cada gen representa un valor específico, no una posición en una secuencia [37].

se almacena como un vector, conceptualmente representa una matriz donde cada elemento $v_{i,t}$ es un gen, como muestra la ecuación 2.8:

$$C = \begin{bmatrix} v_{1,1} & v_{1,2} & \dots & v_{1,N_T} \\ v_{2,1} & v_{2,2} & \dots & v_{2,N_T} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N_V,1} & v_{N_V,2} & \dots & v_{N_V,N_T} \end{bmatrix} \quad (2.8)$$

donde $v_{i,t} \in \{0, 1\}$ representa el estado (cerrado/abierto) de la válvula i en el tiempo t .

2.3.2. Operadores evolutivos

Para garantizar la convergencia y diversidad de la población, se consideran los siguientes operadores:

- Selección por torneo: se eligen k individuos al azar y se selecciona el de mejor aptitud [31].
- Cruce uniforme: dado que el cromosoma es una matriz binaria sin dependencia posicional estricta, el cruce uniforme permite una recombinación eficiente de los genes. Cada bit del hijo se selecciona probabilísticamente de uno de los dos padres, promoviendo una mayor exploración del espacio de búsqueda [32].
- Mutación *bit-flip*: se aplica una mutación puntual donde un gen (estado de válvula) invierte su valor ($0 \rightarrow 1$ o $1 \rightarrow 0$) con una baja probabilidad. Esto introduce diversidad genética y previene el estancamiento en óptimos locales.

2.4. Herramientas computacionales y métricas

Para la materialización de los conceptos teóricos expuestos y la validación de los algoritmos de optimización, se selecciona un conjunto específico de herramientas de software y métricas de desempeño.

2.4.1. Bibliotecas y entorno de desarrollo

El desarrollo se fundamenta en el lenguaje de programación Python, seleccionado por su extenso ecosistema para la ciencia de datos y la ingeniería. Las bibliotecas principales se categorizan según su función en la arquitectura:

- Simulación hidráulica: para la implementación del componente físico ($f_{físico}$), se utiliza la biblioteca Water Network Tool for Resilience (WNTR) [33]. Esta herramienta, basada en el motor estándar EPANET, permite la simulación hidráulica dinámica, facilitando la inyección de parámetros de control en tiempo de ejecución.
- Aprendizaje profundo: el componente corrector de residuales (f_{AI}) se construye sobre PyTorch [34]. Esta biblioteca ofrece la flexibilidad necesaria para diseñar arquitecturas de redes neuronales personalizadas y aprovecha la aceleración por hardware mediante tensores, lo cual es crítico para minimizar la latencia durante las miles de evaluaciones requeridas por el AG.

2.3. Teoría de optimización: algoritmos genéticos

11

Para el problema de despacho de riego, donde la decisión es asignar un estado (abierto/cerrado) a una válvula en un intervalo de tiempo, la codificación por valor binario es la más adecuada. Si el horizonte de planificación T se discretiza en N_T intervalos y hay N_V válvulas, un cromosoma C es un vector binario de longitud $N_V \times N_T$. Aunque se almacena como un vector, conceptualmente representa una matriz donde cada elemento $v_{i,t}$ es un gen, como muestra la ecuación 2.9:

$$C = \begin{bmatrix} v_{1,1} & v_{1,2} & \dots & v_{1,N_T} \\ v_{2,1} & v_{2,2} & \dots & v_{2,N_T} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N_V,1} & v_{N_V,2} & \dots & v_{N_V,N_T} \end{bmatrix} \quad (2.9)$$

Donde $v_{i,t} \in \{0,1\}$ representa el estado (cerrado/abierto) de la válvula i en el tiempo t .

2.3.2. Operadores de selección

La selección determina qué cromosomas (soluciones) de la población actual son elegidos para reproducirse y crear la siguiente generación, aplicando el principio de supervivencia del más apto [38]. En la tabla 2.2 se comparan los operadores de selección más comunes.

TABLA 2.2. Análisis comparativo de operadores de selección en AG.

| Operador | Mecanismo | Ventajas | Desventajas y riesgos |
|--------------------------------|--|--|---|
| Roulette wheel selección (RWS) | Asigna una probabilidad de selección p_i proporcional a la aptitud f_i : $p_i = f_i / \sum f_k$ [38]. | Simple. | Riesgo de convergencia prematura. Un individuo "súper" (con aptitud muy alta) puede dominar la población rápidamente, lo que elimina la diversidad genética [38]. |
| Tournament selección (TOS) | Elige k (ej. $k = 2$) individuos al azar de la población y selecciona al de mayor aptitud para la reproducción [38]. | Computacionalmente eficiente. Menor riesgo de convergencia prematura [39, 40]. | Si k es muy grande converge más rápido pero con riesgo de caer en óptimos locales. |
| Rank selection (LRS) | Selecciona individuos basándose en el ranking de su aptitud, no en el valor absoluto. Asigna probabilidades linealmente al ranking [38, 41]. | Evita los problemas de escalado de RWS. | Más complejo computacionalmente que TOS; puede ser más lento en converger. |

La literatura a menudo reporta que TOS ofrece un equilibrio superior entre exploración y explotación, y un mejor rendimiento general que RWS [39, 40].

2.3.3. Operadores de cruce (crossover / recombinación)

El cruce (*crossover*) es el operador principal de exploración. Combina la información genética de dos cromosomas padres para crear hijos (nuevas soluciones) [35,

2.4. Herramientas computacionales y métricas

11

- Procesamiento numérico y datos: la manipulación eficiente de los cromosomas (matrices binarias) y las operaciones vectoriales del AG se realizan mediante NumPy, asegurando un bajo costo computacional en los operadores de cruce y mutación. Por su parte, Pandas se emplea para la gestión de series temporales y la ingeniería de características previa al entrenamiento.
- Gestión del ciclo de vida: para garantizar la reproducibilidad científica del modelo híbrido, se utiliza MLflow. Esta plataforma permite registrar sistemáticamente los hiperparámetros del modelo, las versiones de los datos de entrenamiento y las métricas de validación, asegurando la trazabilidad completa del experimento.

2.4.2. Métricas de evaluación

La evaluación del desempeño del sistema se aborda desde dos perspectivas complementarias: la precisión predictiva del gemelo digital híbrido y la eficacia del motor de búsqueda del AG.

Para cuantificar la capacidad del modelo híbrido (f_{hybrid}) de replicar la realidad y corregir el sesgo del modelo físico, se emplea principalmente la raíz del error cuadrático medio (RMSE).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.9)$$

donde y_i representa el valor real medido y \hat{y}_i la estimación del modelo. Adicionalmente, se utiliza el coeficiente de determinación (R^2) para medir la proporción de la varianza de los datos reales que es explicada por el modelo, validando su capacidad para capturar la dinámica del sistema.

Por otro lado, la calidad del proceso de optimización se evalúa mediante el monitoreo del costo de la mejor solución a lo largo de las generaciones, lo que permite verificar la convergencia del algoritmo.

[38]. Los operadores dependen críticamente de la codificación del cromosoma, como se observa en la tabla 2.3.

TABLA 2.3. Análisis comparativo de operadores de cruce relevantes.

| Operador | Codificación | Mecanismo | Caso de Uso Principal |
|------------------------|------------------------|---|--|
| Single-point crossover | Valor (binario o real) | Se elige un punto de corte aleatorio. Los padres intercambian sus colas para crear dos hijos. | El operador de cruce más simple. |
| Uniform crossover | Valor (binario o real) | Por cada gen en el cromosoma, se lanza una moneda (con probabilidad p) para decidir si el hijo hereda el gen del padre 1 o del padre 2 [38]. | A menudo es el más efectivo para codificaciones binarias, ya que permite una alta recombinación de bloques de genes. |

2.3.4. Operadores de mutación (diversidad)

La mutación es un operador secundario que introduce variaciones aleatorias en un cromosoma. Su propósito es mantener la diversidad genética y prevenir que la población converja prematuramente a un óptimo local [35, 38].

- *Bit flip mutation* (para codificación por valor binario): es el operador relevante para este problema. Se selecciona un gen (un bit $v_{i,t}$) al azar y se invierte su valor ($0 \rightarrow 1$ o $1 \rightarrow 0$).
- *Swap mutation* (para codificación por permutación): se seleccionan dos genes al azar en el cromosoma y se intercambian sus posiciones [42].

2.4. Manejo de restricciones: funciones de penalización

El desafío final en la aplicación de un AG a un problema de ingeniería del mundo real es el manejo de las restricciones. Los operadores de cruce y mutación son estocásticos y no tienen conocimiento de la física del sistema, por lo tanto, generarán inevitablemente cromosomas (planes de riego) que son infactibles.

El método de funciones de penalización es la técnica de manejo de restricciones (CHT) más común para metaheurísticas [43, 27, 44]. La filosofía es transformar un problema de optimización restringida en un problema no restringido [43, 45]. Esto se logra moviendo las restricciones desde el *solver* hacia la función objetivo [43]. Se añade un término de penalización a la función de aptitud, que castiga a las soluciones infactibles [46, 43]. La formulación matemática de la función objetivo es la ecuación 2.10:

$$Fitness_{penalizada}(x) = Fitness_{original}(x) + p(d(x, B)) \quad (2.10)$$

Donde $Fitness_{original}(x)$ es el objetivo, $d(x, B)$ es una métrica de la magnitud de la violación de la restricción, y $p(\cdot)$ es la función de penalización que traduce esa violación en un incremento del costo [45].

La eficacia del AG depende de cómo se diseña $p(\cdot)$ [46]. Las estrategias varían desde la “pena de muerte” (*death penalty*), que rechaza cualquier solución infacible, hasta enfoques más robustos como las penalizaciones dinámicas, donde la penalización aumenta con cada generación (ej. $p(V, t) = (\lambda \cdot t) \cdot V$) [46, 28].

Capítulo 3

Diseño e implementación

En este capítulo se describe la solución de software completa, detallando la arquitectura y la implementación técnica. La solución se centra en la integración de la simulación hidráulica y la inteligencia artificial para crear un gemelo digital híbrido. Finalmente, se explica cómo este gemelo se conecta a través de una interfaz de programación (API) con el motor de optimización para automatizar el despacho de riego.

3.1. Arquitectura general del software

El diseño del sistema de riego se basa en una arquitectura de cuatro capas: la capa física, la de control y adquisición, la de gestión y optimización, y la de presentación (ver figura 3.1). Este trabajo se enfoca exclusivamente en la capa de gestión y optimización, que es responsable de transformar los datos históricos y de telemetría en un plan de despacho de riego óptimo.

El software se implementó bajo una arquitectura de microservicios [35]. Esto garantiza la separación de responsabilidades y la escalabilidad horizontal de los dos componentes centrales: el entrenamiento continuo del modelo predictivo y su consumo operativo.

El diseño se basa en dos subsistemas, encapsulados en contenedores Docker para asegurar un entorno de ejecución aislado y consistente: el subsistema de entrenamiento y ciclo de vida (MLOps) y el subsistema de optimización y servicio. La vinculación entre ellos se establece a través de un registro central de modelos, lo que permite al motor de decisión consumir siempre la versión más precisa del gemelo digital híbrido, mientras el *pipeline* de MLOps opera de forma asíncrona.

3.1.1. Subsistema de entrenamiento y ciclo de vida (MLOps)

Este subsistema tiene como objetivo principal la sostenibilidad de la precisión del modelo predictivo. Mantiene el gemelo digital híbrido actualizado al ejecutar un ciclo de vida automatizado que mitiga el (*model drift*) [36]. Este flujo de trabajo, fundamental para un entorno de producción, está orquestado por *pipelines* de Apache Airflow [37].

La arquitectura comprende un *stack* de infraestructura para el manejo de los artefactos y la trazabilidad de los experimentos:

- Data Lake (MinIO): repositorio central de objetos que almacena los datos de telemetría históricos y los datasets limpios, actuando como la fuente única de verdad para el entrenamiento.

- Registro de Artefactos (MLflow): administra el ciclo de vida completo del modelo [38]. Registra los hiperparámetros, métricas y los artefactos binarios (el modelo y los transformadores de características) que son promovidos a producción para su consumo por el servicio de inferencia.
- Orquestación (PostgreSQL/Airflow): la base de datos PostgreSQL gestiona los metadatos y el estado de los *pipelines* de Airflow, asegurando la fiabilidad y reintentos del proceso de integración y despliegue continuo (CI/CD) [39].

El *pipeline* de entrenamiento se estructura en una secuencia de módulos lógicos de procesamiento de datos:

1. Módulo de ingesta y consolidación: responsable de la lectura y unificación de los datos de telemetría histórica del campo desde el Data Lake.
2. Módulo de limpieza: filtra los registros inconsistentes y asegura la calidad del *dataset*.
3. Módulo de ingeniería de características: genera las variables sintéticas necesarias para el modelo de corrección de residuales, como el número de actuadores activos.
4. Módulo de entrenamiento: ejecuta el entrenamiento. Si el nuevo modelo supera los umbrales de desempeño, es promovido al alias de *production* en el registro de modelos, lo que completa el ciclo de integración y despliegue continuo (CI/CD) [39].

3.1.2. Subsistema de optimización y servicio

Este subsistema constituye el corazón operativo del sistema, implementando el motor de decisión encargado de resolver el problema de programación matemática. Su arquitectura es diseñada específicamente para superar el cuello de botella computacional que representa la simulación hidráulica secuencial. Para ello, se diferencian dos componentes acoplados mediante una estrategia de escalabilidad horizontal.

1. Capa de servicio de inferencia distribuida (*Snapshot API*): el diseño de arquitectura resuelve el desafío de rendimiento mediante la implementación de un clúster de microservicios de simulación. Cada unidad es un servicio RESTful asíncrono desarrollado con FastAPI [40], diseñado para ser ligero y sin estado (*stateless*) en cuanto a la petición, pero manteniendo en memoria el estado completo del gemelo digital híbrido (modelo físico WNTR más el modelo corrector de IA).

Para lograr la aceleración requerida por el algoritmo genético, se implementa una estrategia de orquestación mediante Docker Swarm [41]. Esta tecnología permite instanciar múltiples réplicas del microservicio de simulación, distribuyéndolas dinámicamente a través de los nodos físicos disponibles en el clúster. El tráfico entrante es gestionado por una instancia de NGINX [42], que actúa como balanceador de carga, distribuyendo las miles de peticiones de evaluación generadas por el optimizador entre las distintas réplicas activas. Esta arquitectura permite paralelizar masivamente el cálculo de la función de costo, reduciendo el tiempo de búsqueda de forma lineal con respecto a la cantidad de recursos de hardware agregados.

Capítulo 3

Diseño e implementación

En este capítulo se describe la solución de software completa, detallando la arquitectura y la implementación técnica. La solución se centra en la integración de la simulación hidráulica y la inteligencia artificial para crear un gemelo digital híbrido. Finalmente, se explica cómo este gemelo se conecta a través de una interfaz de programación (API) con el motor de optimización, automatizando así el despacho de riego.

3.1. Arquitectura general del software

El diseño del sistema de riego se basa en una arquitectura de cuatro capas: la capa física, la de control y adquisición, la de gestión y optimización, y la de presentación (ver figura 3.1). Este trabajo se enfoca exclusivamente en la capa de gestión y optimización, que es responsable de transformar los datos históricos y de telemetría en un plan de despacho de riego óptimo.

El software se implementó bajo una arquitectura de microservicios [47]. Esta arquitectura garantiza la separación de responsabilidades y la escalabilidad horizontal de los dos componentes centrales: el entrenamiento continuo del modelo predictivo y su consumo operativo.

El diseño se basa en dos subsistemas, encapsulados en contenedores Docker para asegurar un entorno de ejecución aislado y consistente: el subsistema de entrenamiento y ciclo de vida (MLOps) y el subsistema de optimización y servicio. La vinculación entre ellos se establece a través de un registro central de modelos, lo que permite al motor de decisión consumir siempre la versión más precisa del gemelo digital híbrido, mientras el *pipeline* de MLOps opera de forma asíncrona.

3.1.1. Subsistema de entrenamiento y ciclo de vida (MLOps)

Este subsistema tiene como objetivo principal la sostenibilidad de la precisión del modelo predictivo. Mantiene el gemelo digital híbrido actualizado al ejecutar un ciclo de vida automatizado que mitiga el (*model drift*) [48]. Este flujo de trabajo, fundamental para un entorno de producción, está orquestado por *pipelines* de Apache Airflow [49].

La arquitectura comprende un *stack* de infraestructura para el manejo de los artefactos y la trazabilidad de los experimentos:

- Data Lake (*MinIO*): repositorio central de objetos que almacena los datos de telemetría históricos y los *datasets* limpios, actuando como la fuente única de verdad para el entrenamiento.

3.1. Arquitectura general del software

2. Motor de optimización: el motor implementa la lógica metaheurística del AG. A diferencia de una implementación secuencial tradicional, este módulo dispone de una interfaz cliente asíncrona que aprovecha la concurrencia (*multithreading*) para enviar lotes de solicitudes de simulación simultáneas.

Durante cada generación del AG, la evaluación de la aptitud de la población no se realiza individuo por individuo, sino que se distribuye a través de la red hacia el balanceador de carga. Esto permite que múltiples planes de riego sean simulados al mismo tiempo por las distintas instancias del servicio de inferencia, lo que permite aprovechar al máximo la capacidad de cómputo instalada y cumplir con los estrictos requisitos temporales.

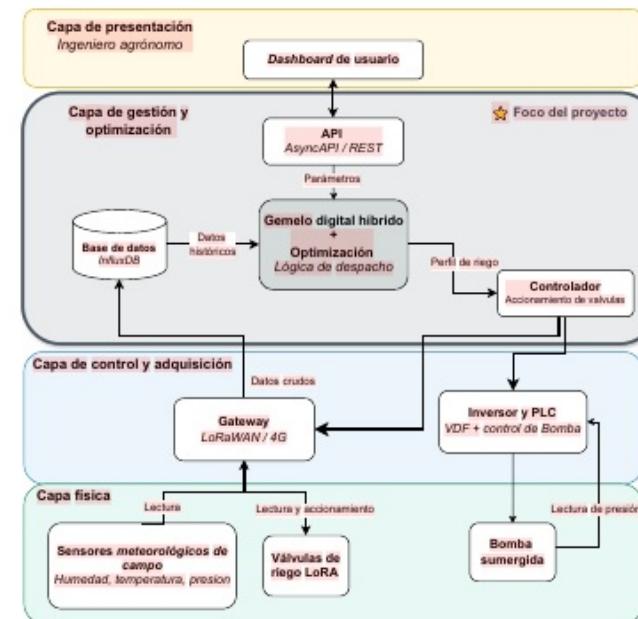


FIGURA 3.1. Diagrama de capas del sistema de riego, donde el foco del trabajo es la capa de gestión y optimización.

- Registro de Artefactos (*MLflow*): administra el ciclo de vida completo del modelo [50]. Registra los hiperparámetros, métricas y los artefactos binarios (el modelo y los transformadores de características) que son promovidos a producción para su consumo por el servicio de inferencia.
- Orquestación (*PostgreSQL/Airflow*): la base de datos *PostgreSQL* gestiona los metadatos y el estado de los *pipelines* de *Airflow*, asegurando la fiabilidad y reintentos del proceso de integración y despliegue continuo (CI/CD) [51].

El *pipeline* de entrenamiento se estructura en una secuencia de módulos lógicos de procesamiento de datos:

1. Módulo de ingesta y consolidación: responsable de la lectura y unificación de los datos de telemetría histórica del campo desde el *Data Lake*.
2. Módulo de limpieza: filtra los registros inconsistentes y asegura la calidad del *dataset*.
3. Módulo de ingeniería de características: genera las variables sintéticas necesarias para el modelo de corrección de residuales, como el número de actuadores activos.
4. Módulo de entrenamiento: ejecuta el entrenamiento. Si el nuevo modelo supera los umbrales de desempeño, es promovido al alias de "production" en el registro de modelos, completando el ciclo de integración y despliegue continuo (CI/CD) [51].

3.1.2. Subsistema de optimización y servicio

Este subsistema constituye el corazón operativo del sistema, implementando el motor de decisión encargado de resolver el problema de programación matemática. Su arquitectura es diseñada específicamente para superar el cuello de botella computacional que representa la simulación hidráulica secuencial. Para ello, se diferencian dos componentes acoplados mediante una estrategia de escalabilidad horizontal.

1. Capa de servicio de inferencia distribuida (*Snapshot API*): el diseño de arquitectura resuelve el desafío de rendimiento mediante la implementación de un clúster de microservicios de simulación. Cada unidad es un servicio RESTful asíncrono desarrollado con *FastAPI* [52], diseñado para ser ligero y sin estado (*stateless*) en cuanto a la petición, pero manteniendo en memoria el estado completo del gemelo digital híbrido (modelo físico WNTR más el modelo corrector de IA).

Para lograr la aceleración requerida por el algoritmo genético, se implementa una estrategia de orquestación mediante *Docker Swarm* [53]. Esta tecnología permite instanciar múltiples réplicas del microservicio de simulación, distribuyéndolas dinámicamente a través de los nodos físicos disponibles en el clúster. El tráfico entrante es gestionado por una instancia de *NGINX* [54], que actúa como balanceador de carga, distribuyendo las miles de peticiones de evaluación generadas por el optimizador entre las distintas réplicas activas. Esta arquitectura permite paralelizar masivamente el cálculo de la función de costo, reduciendo el tiempo de búsqueda de forma lineal con respecto a la cantidad de recursos de hardware agregados.

3.2. Implementación del gemelo digital híbrido

El gemelo digital se desarrolló bajo el paradigma de caja gris, fusionando un modelo hidráulico de primeros principios con un componente de aprendizaje profundo (*deep learning*). Este último tiene la función específica de predecir y corregir los errores residuales de la simulación física. Para asegurar la vigencia del sistema, la solución se integró en un ciclo de vida automatizado que ejecuta entrenamientos periódicos del modelo ante la llegada de nuevos datos de campo.

La figura 3.2 presenta el esquema conceptual del sistema implementado, ilustrando el flujo de información desde la ingestión de datos de campo hasta la generación del estado corregido del sistema.

A continuación, se detalla la ingeniería aplicada en cada uno de sus componentes constitutivos.

3.2.1. Construcción y parametrización del modelo físico

El componente físico ($f_{\text{físico}}$) se implementó utilizando la biblioteca *Water Network Tool for Resilience* (WNTR) [33]. Se optó por definir la topología de la red mediante funciones de Python en lugar de utilizar archivos de configuración estáticos. Esta estrategia permite modificar las propiedades de la infraestructura, como la curva de la bomba o la longitud de los laterales, alterando directamente las variables del código fuente, lo cual elimina la necesidad de regenerar archivos externos ante cada cambio en los parámetros de diseño.

La red se modeló representando explícitamente los componentes hidráulicos principales: la fuente de agua subterránea, el grupo de bombeo, el cabezal de filtrado y la red de distribución. Para garantizar la fidelidad de la simulación base, se configuraron los siguientes parámetros constitutivos extraídos de las especificaciones técnicas:

- **Modelo de fricción en tuberías:** para la cuantificación de las pérdidas por fricción, se seleccionó la ecuación de Hazen-Williams. Se estableció un coeficiente de rugosidad adimensional $C = 130.0$, aplicado de manera uniforme a los tramos de conducción principal y a los laterales de riego, valor que corresponde a tuberías plásticas con un desgaste operativo estándar.
- **Caracterización de la bomba:** el comportamiento del sistema de bombeo se representó mediante la curva característica de altura manométrica-caudal ($H - Q$). Esta curva se definió mediante la interpolación de 16 puntos operativos, abarcando el rango completo desde la presión a válvula cerrada ($H \approx 124$ m) hasta el caudal máximo.
- **Comportamiento de los emisores:** se configuró un exponente de flujo $\gamma = 0.46$, propio del régimen turbulento en los emisores instalados, y un coeficiente de descarga base ajustado a las especificaciones del fabricante.

Sin embargo, la simulación individual de los miles de goteros presentes en el campo generaba una carga computacional incompatible con los tiempos de respuesta requeridos. Para solucionar este desafío, se implementó un algoritmo de agregación espacial en la función de generación de laterales. Este algoritmo discretiza cada lateral de riego en un número fijo de segmentos. En lugar de instanciar cada emisor físico, se asigna un emisor equivalente a cada nodo de discretización, cuyo coeficiente de descarga se calcula como la suma aritmética de los coeficientes de

3.2. Implementación del gemelo digital híbrido

17

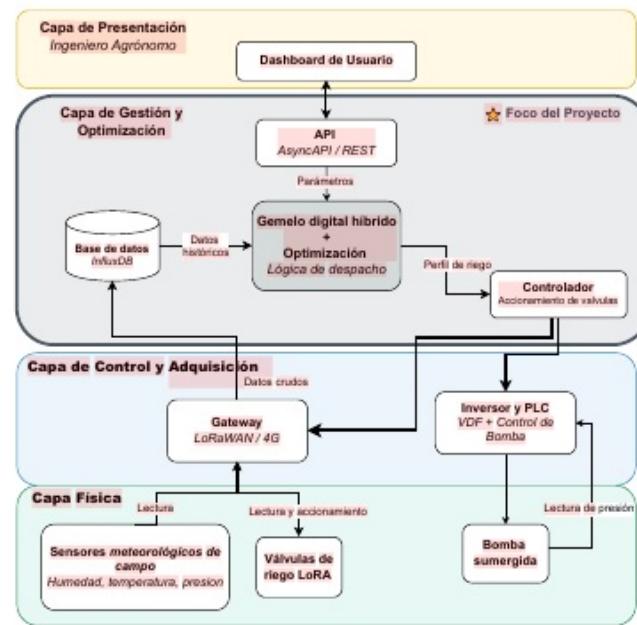


FIGURA 3.1. Diagrama de capas del sistema de riego, donde el foco del trabajo es la capa de gestión y optimización.

2. Motor de optimización: el motor implementa la lógica metaheurística del AG. A diferencia de una implementación secuencial tradicional, este módulo dispone de una interfaz cliente asíncrona que aprovecha la concurrencia (*multithreading*) para enviar lotes de solicitudes de simulación simultáneas.

Durante cada generación del AG, la evaluación de la aptitud de la población no se realiza individuo por individuo, sino que se distribuye a través de la red hacia el balanceador de carga. Esto permite que múltiples planes de riego sean simulados al mismo tiempo por las distintas instancias del servicio de inferencia, aprovechando al máximo la capacidad de cómputo instalada y cumpliendo con los estrictos requisitos temporales.

3.2. Implementación del gemelo digital híbrido

3.3. Implementación del optimizador de despacho (GA)

3.4. Implementación de funciones de costo y penalizaciones

3.2. Implementación del gemelo digital híbrido

17

los goteros reales contenidos en dicho segmento. Esta técnica reduce la dimensión de la matriz hidráulica que resuelve el motor WNTR, disminuyendo el tiempo de ejecución de la simulación de segundos a milisegundos.

Finalmente, la ejecución del modelo se configuró bajo el esquema de análisis dirigido por presión (PDA). A diferencia del modelo de demanda base (DDA), que fija el consumo independientemente del estado hidráulico, el esquema PDA calcula el caudal real entregado por los emisores como una función de la presión nodal instantánea.

3.2.2. Ingeniería de datos y modelo de corrección

El componente de inteligencia artificial (f_{AI}) se diseñó con el objetivo específico de predecir el residual (R), definido como la diferencia entre el caudal real medido y la estimación teórica del modelo físico base. Para garantizar la calidad de la información de entrenamiento, el flujo de datos inicia con una etapa de limpieza rigurosa en la que se aplican filtros para eliminar registros inconsistentes, tales como instantes donde los datos reportan caudales positivos y válvulas cerradas.

Sobre el conjunto de datos depurado, se ejecutó un proceso de ingeniería de características orientado a capturar las no linealidades del sistema hidráulico. A las variables base disponibles, la presión real medida en cabecera y el caudal simulado por el modelo físico, se sumaron variables sintéticas. Se generó el conteo total de válvulas activas (*valve_count*) mediante la suma de los estados binarios de los sectores, y se creó un término de interacción física (*presión_x_valve_count*) multiplicando la presión real por dicho conteo. Esta última variable actúa como un estimador de la impedancia hidráulica total del sistema, facilitando a la red neuronal la distinción entre variaciones de caudal debidas a la operación de la bomba y aquellas causadas por apertura y cierre de válvulas.

Previo al entrenamiento, se aplicó una estrategia de saneamiento estadístico sobre la variable objetivo (el residual), eliminando valores atípicos (*outliers*) mediante el método del rango intercuartílico (IQR). Posteriormente, los datos se dividieron cronológicamente en subconjuntos de entrenamiento, validación y prueba, y se normalizaron las características numéricas utilizando un escalador estándar (*StandardScaler*). Es fundamental destacar que el escalador se ajusta exclusivamente con los parámetros estadísticos del conjunto de entrenamiento y luego se utiliza para transformar los conjuntos restantes, evitando así la fuga de información estadística (*data leakage*) hacia los datos de validación.

Finalmente, el modelo de corrección se implementó utilizando un MLP desarrollado con la biblioteca PyTorch [34]. La arquitectura de la red, compuesta por capas densas con funciones de activación no lineales, se optimizó para realizar la regresión del valor residual a partir del vector de características procesado. El proceso de entrenamiento minimiza la función de pérdida de error MSE e integra un mecanismo de parada temprana (*early stopping*) que monitorea el desempeño en el conjunto de validación, previniendo el sobreajuste y garantizando la robustez del modelo ante nuevas condiciones operativas.

3.2.3. Ciclo de vida y gestión de artefactos (MLOps)

La implementación del gemelo digital trasciende el código fuente. Se monta sobre una infraestructura de operaciones de aprendizaje automático (MLOps) orquestada por Apache Airflow [37]. Este sistema gestiona el flujo de datos desde su ingesta hasta el despliegue, estructurado en las siguientes etapas:

- Gestión de datos en capas: se utiliza un Data Lake basado en MinIO para almacenar los datos en tres estadios de refinamiento: crudos (*raw*), intermedios (*intermediate*) y procesados (*primary*). Esta arquitectura de capas permite la trazabilidad y la reproducibilidad de cualquier versión del *dataset*.
- Registro de experimentos: cada ejecución del *pipeline* de entrenamiento registra automáticamente las métricas de desempeño (RMSE, R^2) y los artefactos resultantes (el modelo serializado y los objetos escaladores) en un servidor de MLflow [38]. Esto permite analizar la evolución del desempeño del modelo a lo largo del tiempo.
- Validación y promoción automática: el *pipeline* incluye una etapa de decisión autónoma. Tras el entrenamiento, el sistema evalúa el nuevo modelo frente a un conjunto de prueba. Solo si el nuevo modelo supera en precisión al vigente (reducción del RMSE), se promueve automáticamente mediante el etiquetado de alias en el registro, quedando disponible inmediatamente para el servicio de inferencia.

3.2.4. Orquestación y ejecución en el servicio de inferencia

El servicio de inferencia (*Snapshot API*) es el componente que ejecuta el gemelo digital híbrido. Su función principal es coordinar la comunicación entre los modelos almacenados y las solicitudes de simulación que envía el optimizador. El código se diseñó para cargar todo lo necesario en la memoria RAM al iniciar el servidor, evitando así lecturas lentas de disco durante la operación.

Al arrancar, el servicio se conecta al registro de MLflow y descarga la versión del modelo de inteligencia artificial marcada como *production*. También inicializa el modelo físico WNTR. Esta preparación previa permite que el sistema responda a las peticiones de simulación más rápidamente.

Cada vez que el optimizador solicita evaluar un plan de riego, el servicio ejecuta los siguientes pasos de forma secuencial:

1. Cálculo físico base: el servicio ejecuta primero el simulador WNTR. Este paso entrega un valor de caudal y las presiones para la configuración de válvulas solicitada.
2. Preparación de datos para la IA: el sistema toma los resultados físicos y calcula las variables adicionales que necesita la red neuronal, como la cantidad de válvulas abiertas y su interacción con la presión. Luego ajusta la escala de estos valores utilizando el artefacto guardado en MLflow durante el entrenamiento.
3. Corrección del error: la red neuronal recibe los datos preparados y predice el error residual del modelo físico. El sistema suma este error a la predicción física inicial para obtener el caudal final corregido.

Capítulo 4

Ensayos y resultados

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

4.1. Entorno y banco de pruebas

4.2. Pruebas del gemelo digital híbrido

4.3. Caso de estudio: simulación de despacho

3.2. Implementación del gemelo digital híbrido

4. Respuesta: con el caudal corregido, el sistema estima cómo se distribuye el caudal residual entre los sectores activos y calcula cuánto desciende el nivel del acuífero usando un modelo de pozo específico. Estos valores finales se envían de vuelta al optimizador para que evalúe si el plan es bueno o malo.

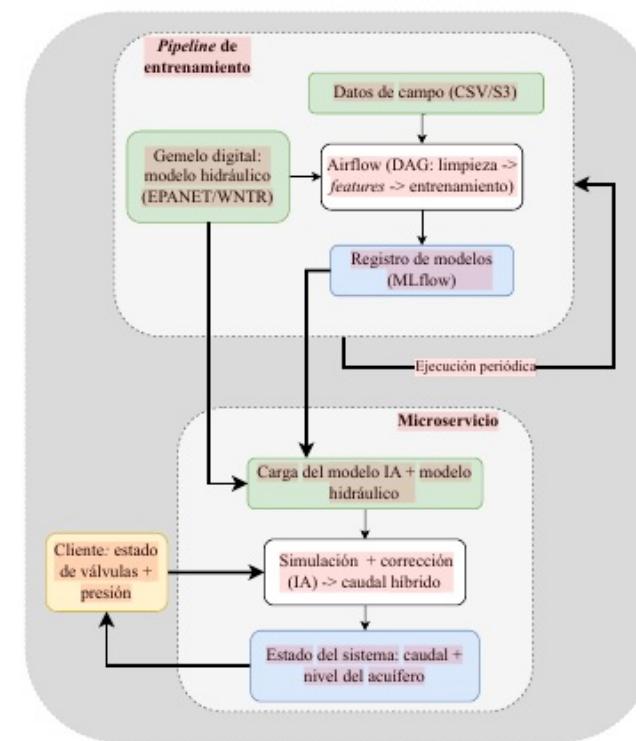


FIGURA 3.2. Esquema del gemelo digital híbrido: integración del pipeline de entrenamiento (MLOps), el modelo físico WNTR y el corrector de IA en el servicio de inferencia.

- 3.3. Implementación del optimizador de despacho (GA)
- 3.4. Implementación de funciones de costo y penalizaciones

Capítulo 5

Conclusiones

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

5.1. Resultados obtenidos

5.2. Trabajo futuro

Capítulo 4

Ensayos y resultados

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

4.1. Entorno y banco de pruebas

4.2. Pruebas del gemelo digital híbrido

4.3. Caso de estudio: simulación de despacho

Bibliografía

- [1] Albert Sagala y col. «Implementation of Portable Off-Grid Solar Water Pump for Irrigation Systems». En: *Jurnal Mantik* (2022). URL: https://www.researchgate.net/publication/358920031_Implementation_of_Portable_Off-Grid_Solar_Water_Pump_for_Irrigation_Systems.
- [2] Giuseppina Mandalaro y col. «Pistachio Nuts (*Pistacia vera L.*): Production, Nutrients, Bioactives and Novel Health Effects». En: *Plants* (2021). doi: 10.3390/plants11010018. URL: <https://doi.org/10.3390/plants11010018>.
- [3] Lidia Núñez y col. «Pistachio Phenology and Yield in a Cold-Winter Region of Spain: The Status of the Cultivation and Performance of Three Cultivars». En: *Horticulturae* (2024). doi: 10.3390/horticulturae10121235. URL: <https://doi.org/10.3390/horticulturae10121235>.
- [4] R. Kanber. «Growth, yield and periodicity of pistachio under different irrigation and fertilization regimes». En: *Water and nutrient management for slope agriculture and horticulture*. Options Méditerranéennes. CIHEAM - Mediterranean Agronomic Institute (proceedings / technical paper). Disponible en: <https://om.ciheam.org/> (consulta la edición original para la paginación exacta). CIHEAM / Options Méditerranéennes, Series B, Studies y Research, 2003.
- [5] MDPI. *Internet of Things-Based Automated Solutions Utilizing Machine Learning for Smart and Real-Time Irrigation Management: A Review*. <https://www.mdpi.com/1424-8220/24/23/7480>. Accedido: 3 de septiembre de 2025. 2024.
- [6] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989. ISBN: 978-0201157673.
- [7] Lewis A. Rossman. *EPANET 2: Users Manual*. Water Supply and Water Resources Division, National Risk Management Research Laboratory. U.S. Environmental Protection Agency (EPA). Cincinnati, OH, USA, 2000. URL: <https://www.epa.gov/water-research/epanet>.
- [8] Huy Truong. *DITEC-WDN: A Large-Scale Dataset of Hydraulic Scenarios across Multiple Water Distribution Networks*. <https://arxiv.org/html/2503.17167v2>. 2025. arXiv: 2503.17167v2.
- [9] M. E. T. et al. «Calibration via Multi-period State Estimation in Water Distribution Systems». En: *World Environmental and Water Resources Congress 2017*. ASCE, 2017.
- [10] Barbara Rakitsch Maja Rudolph Stefan Kurz. *Hybrid Modeling Design Patterns*. <https://arxiv.org/pdf/2401.00033>. 2024. arXiv: 2401.00033.
- [11] G. P. G. de Oliveira y col. «Calibration Model for Water Distribution Network Using Pressures Estimated by Artificial Neural Networks». En: *Procedia Engineering*, Vol. 186. Elsevier, 2017, págs. 434-441.
- [12] Jie Li y col. «Physics-based and data-driven hybrid modeling in manufacturing: a review». En: *Tsinghua University Press* (2024).

Capítulo 5

Conclusiones

Todos los capítulos deben comenzar con un breve párrafo introductorio que indique cuál es el contenido que se encontrará al leerlo. La redacción sobre el contenido de la memoria debe hacerse en presente y todo lo referido al proyecto en pasado, siempre de modo impersonal.

5.1. Resultados obtenidos

5.2. Trabajo futuro

- [13] Meng-Han Tao et al. «A comprehensive review of physics-informed deep learning: Applications, advancements, and challenges». En: *The Innovation* 5.6 (2024), pág. 102548.
- [14] Wikipedia contributors. *Grey box model — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Grey_box_model. Accedido: 3 de septiembre de 2025. 2024.
- [15] Henrik Madsen. «Grey Box Modelling of Hydrological Systems With Focus on Uncertainties». Tesis doct. Technical University of Denmark (DTU), 2011.
- [16] Christodoulos A. Floudas. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. New York, NY: Oxford University Press, 1995. ISBN: 9780195095295.
- [17] Phil Husbands. *Genetic Algorithms for Scheduling*. Inf. téc. Accedido: 24 de octubre de 2025. University of Sussex, 1996.
- [18] À. Corominas y J. M. de la Fuente. *GENETIC ALGORITHMS FOR SHOP SCHEDULING PROBLEMS: A SURVEY*. Inf. téc. Accedido: 24 de octubre de 2025. Universitat Politècnica de Catalunya, 2000.
- [19] A. S. M. M. Jaber y S. M. B. M. Shapiai. «Performance Comparison of Metaheuristic Optimization Algorithms in Solving Production Scheduling Problems». En: *Journal of Intelligent Computation and Technology* 1.1 (2021), págs. 1-11.
- [20] M. B. M. Kamar y M. A. M. Ali. «A comparative study of metaheuristics algorithms based on their performance of complex benchmark problems». En: *Decision Making: Applications in Management and Engineering* 6.1 (2023). Accedido: 24 de octubre de 2025.
- [21] U.S. Environmental Protection Agency. *Water Network Tool for Resilience (WNTR) Documentation*. Accessed: YYYY-MM-DD. U.S. Environmental Protection Agency. Washington, D.C., 2024. URL: <https://usepa.github.io/WNTR/>.
- [22] Ryan Holbrook. *Hybrid Models*. <https://www.kaggle.com/code/ryanholbrook/hybrid-models>. Accedido: 3 de septiembre de 2025. 2025.
- [23] Xinyang Wu et al. *A Hybrid Science-Guided Machine Learning Approach for Modeling Chemical Processes*. <https://arxiv.org/pdf/2112.01475.pdf>. 2021. arXiv: 2112.01475.
- [24] William Braham. *White, Black, and Gray-Box Modelling*. University of Pennsylvania. 2025. URL: <https://www.design.upenn.edu/work/white-black-and-gray-box-modelling>.
- [25] Various. «A Comparative Study of White Box, Black Box and Grey Box Testing Techniques». En: *TheSAI* (2012). URL: <https://thesai.org/Downloads/Volume3No6/Paper%203-A%20Comparative%20Study%20of%20White%20Box,%20Black%20Box%20and%20Grey%20Box%20Testing%20Techniques.pdf>.
- [26] Y. Tao y col. «Multi-objective optimization of water distribution networks based on non-dominated sequencing genetic algorithm». En: *PLoS ONE* 17.11 (2022). URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0277954>.
- [27] Various. «Optimization of Water Distribution Networks Using Genetic Algorithm Based SOP-WDN Program». En: *MDPI* 14.6 (2022). URL: <https://www.mdpi.com/2073-4441/14/6/851>.
- [28] M. Rügner y col. «Physics-based machine learning predictions despite or in combination with scarce data». En: *PMSA* 1.1 (2021). URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8069582/>.

- [29] S. E. W. T. M. H. van der Meer. «Hybrid Modelling by Machine Learning Corrections of Analytical Model Predictions towards High-Fidelity Simulation Solutions». En: *ResearchGate* (2021). URL: https://www.researchgate.net/publication/350813252_Hybrid_Modelling_by_Machine_Learning_Corrections_of_Analytical_Model_Predictions_towards_High-Fidelity_Simulation_Solutions.
- [30] Various. «Efficient Building Simulations Using a Novel Grey-Box Model for an Integrated Water-Based Heating and Cooling System». En: *MDPI* (2024). URL: <https://www.mdpi.com/2075-5309/15/11/1959>.
- [31] Larry W. Mays. *Water Distribution Systems Handbook*. McGraw-Hill, 2001. Cap. Hydraulic Principles of Pipe Flow.
- [32] J. Keller y D. Karmeli. «Trickle Irrigation Design Parameters». En: *Journal of the Irrigation and Drainage Division* 101.IR3 (1975), págs. 275-291.
- [33] Lewis A. Rossman. *EPANET 2 Users Manual*. U.S. Environmental Protection Agency. Cincinnati, OH, 2000.
- [34] Various. *Genetic Algorithms for Combinatorial Optimization Problems*. Inf. téc. Ghent University, 2005. URL: <https://backoffice.biblio.ugent.be/download/1147278/1147528>.
- [35] Various. «A Review of Timetable Scheduling System Using Genetic Algorithm». En: *International Journal of Trend in Research and Development* 6.1 (2019). URL: https://www.researchgate.net/publication/392508312_A_Review_of_Timetable_Scheduling_System_Using_Genetic_Algorithm.
- [36] Various. «Variable-Length Chromosome Genetic Algorithm for Network Scheduling Problem». En: *PMC* (2021). URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8228978/>.
- [37] Various. «Performance of Genetic Algorithm with Different Encoding Scheme». En: *GJETA* (2025). URL: <https://gjeta.com/sites/default/files/GJETA-2025-0203.pdf>.
- [38] Various. «Truncation Selection Operator for Enhancing the Performance of Wireless Sensor Networks». En: *MDPI* 11.1 (2022). URL: <https://www.mdpi.com/2079-9292/11/1/28>.
- [39] Various. «A Comparative Review Between Various Selection Techniques In Genetic Algorithm For Finding Optimal Solutions». En: *ResearchGate* (2022). URL: https://www.researchgate.net/publication/364952362_A_Comparative_Review_Between_Various_Selection_Techniques_In_Genetic_Algorithm_For_Finding_Optimal_Solutions.
- [40] Various. «A Comparison of Selection Schemes used in Genetic Algorithms for solving the Travelling Salesman Problem». En: *CORE* (). URL: <https://core.ac.uk/download/pdf/159178846.pdf>.
- [41] Various. «Comparative Analysis of Rank and Roulette Wheel Selection Methods for Optimizing Spatial Layout Arrangements». En: *TheSAI* (2024). URL: https://thesai.org/Downloads/Volume16No6/Paper_56-Comparative_Analysis_of_Rank_and_Roulette_Wheel_Selection.pdf.
- [42] Stack Overflow Community. *Genetic algorithms crossover and mutation operators for paths*. <https://stackoverflow.com/questions/12687963/genetic-algorithms-crossover-and-mutation-operators-for-paths>. 2012.
- [43] arXiv. *Inverse Problems and Data Assimilation*. <https://arxiv.org/abs/1810.06191>. Accessed: October 15, 2025. 2018.

Bibliografía

- [1] Albert Sagala y col. «Implementation of Portable Off-Grid Solar Water Pump for Irrigation Systems». En: *Jurnal Mantik* (2022). URL: https://www.researchgate.net/publication/358920031_Implementation_of_Portable_Off-Grid_Solar_Water_Pump_for_Irrigation_Systems.
- [2] Giuseppina Mandalari y col. «Pistachio Nuts (*Pistacia vera L.*): Production, Nutrients, Bioactives and Novel Health Effects». En: *Plants* (2021). DOI: 10.3390/plants11010018. URL: <https://doi.org/10.3390/plants11010018>.
- [3] Lidia Núñez y col. «Pistachio Phenology and Yield in a Cold-Winter Region of Spain: The Status of the Cultivation and Performance of Three Cultivars». En: *Horticulturae* (2024). DOI: 10.3390/horticulturae10121235. URL: <https://doi.org/10.3390/horticulturae10121235>.
- [4] R. Kanber. «Growth, yield and periodicity of pistachio under different irrigation and fertilization regimes». En: *Water and nutrient management for slope agriculture and horticulture*. Options Méditerranéennes. CIHEAM - Mediterranean Agronomic Institute (proceedings / technical paper). Disponible en: <https://om.ciheam.org/> (consulta la edición original para la página exacta). CIHEAM / Options Méditerranéennes, Series B, Studies y Research, 2003.
- [5] MDPI. *Internet of Things-Based Automated Solutions Utilizing Machine Learning for Smart and Real-Time Irrigation Management: A Review*. <https://www.mdpi.com/1424-8220/24/23/7480>. Accedido: 3 de septiembre de 2025. 2024.
- [6] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989. ISBN: 978-0201157673.
- [7] Lewis A. Rossman. *EPANET 2: Users Manual*. Water Supply and Water Resources Division, National Risk Management Research Laboratory. U.S. Environmental Protection Agency (EPA). Cincinnati, OH, USA, 2000. URL: <https://www.epa.gov/water-research/epanet>.
- [8] Huy Truong. *DITEC-WDN: A Large-Scale Dataset of Hydraulic Scenarios across Multiple Water Distribution Networks*. <https://arxiv.org/html/2503.17167v2>. 2025. arXiv: 2503.17167v2.
- [9] M. E. T. et al. «Calibration via Multi-period State Estimation in Water Distribution Systems». En: *World Environmental and Water Resources Congress 2017*. ASCE, 2017.
- [10] Barbara Rakitsch Maja Rudolph Stefan Kurz. *Hybrid Modeling Design Patterns*. <https://arxiv.org/pdf/2401.00033>. 2024. arXiv: 2401.00033.
- [11] G. P. G. de Oliveira y col. «Calibration Model for Water Distribution Network Using Pressures Estimated by Artificial Neural Networks». En: *Procedia Engineering*, Vol. 186. Elsevier, 2017, págs. 434-441.
- [12] Jie Li y col. «Physics-based and data-driven hybrid modeling in manufacturing: a review». En: *Tsinghua University Press* (2024).

- [44] B. Grové y M. C. Du Plessis. «Optimising intra-seasonal irrigation water allocation: a comparison of mixed integer nonlinear programming and differential evolution». En: *Water SA* 45.1 (2019). URL: <https://www.watersa.net/article/view/6674>.
- [45] *Mass Balance Constraints - WEAP*. https://www.weap21.org/webhelp/mass_balance_constraints.htm. Accessed: October 15, 2025.
- [46] Autor Anónimo. *Un Marco para la Estimación de Estado Completo en Redes Hidráulicas mediante Modelado Híbrido de Residuales y Solucionadores de Problemas Inversos*. Documento interno del proyecto. 2025.
- [47] Sam Newman. *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media, 2015. ISBN: 9781491950357.
- [48] Thomas Schlachter y Felix Sager. «Hybrid modeling of physical systems using neural networks for residual corrections». En: *Journal of Process Control* 96 (2020), págs. 1-11. DOI: [10.1016/j.jprocont.2020.10.002](https://doi.org/10.1016/j.jprocont.2020.10.002).
- [49] The Apache Software Foundation. *Apache Airflow Documentation*. <https://airflow.apache.org/docs/>. Consultado el 27 de octubre de 2025. 2024.
- [50] Matei Zaharia y col. «MLflow: A Platform for the Machine Learning Lifecycle». En: *Proc. of the 2nd International Workshop on Systems for ML* (2018). Presentado en SysML 2018. URL: <https://www.mlflow.org/>.
- [51] Kathryn Holovaty y Matthew Kopecky. «MLOps: Principles and Practices». En: *Gartner Research Report* (2020). Informe de arquitectura sobre el ciclo de vida de modelos en producción.
- [52] Sebastián Ramírez. *FastAPI Documentation*. Framework web moderno y de alto rendimiento para construir APIs con Python. 2024. URL: <https://fastapi.tiangolo.com/>.
- [53] Docker Inc. *Docker Swarm Mode Overview*. Herramienta de orquestación nativa para gestionar clústeres de Docker. 2024. URL: <https://docs.docker.com/engine/swarm/>.
- [54] F5, Inc. *NGINX Documentation*. Servidor web y balanceador de carga de alto rendimiento. 2024. URL: <https://nginx.org/en/docs/>.

- [13] Meng-Han Tao et al. «A comprehensive review of physics-informed deep learning: Applications, advancements, and challenges». En: *The Innovation* 5.6 (2024), pág. 102548.
- [14] Wikipedia contributors. *Grey box model* — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Grey_box_model. Accedido: 3 de septiembre de 2025. 2024.
- [15] Henrik Madsen. «Grey Box Modelling of Hydrological Systems With Focus on Uncertainties». Tesis doctoral. Technical University of Denmark (DTU), 2011.
- [16] Christodoulos A. Floudas. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. New York, NY: Oxford University Press, 1995. ISBN: 9780195095295.
- [17] Phil Husbands. *Genetic Algorithms for Scheduling*. Inf. téc. Accedido: 24 de octubre de 2025. University of Sussex, 1996.
- [18] À. Corominas y J. M. de la Fuente. *GENETIC ALGORITHMS FOR SHOP SCHEDULING PROBLEMS: A SURVEY*. Inf. téc. Accedido: 24 de octubre de 2025. Universitat Politècnica de Catalunya, 2000.
- [19] A. S. M. M. Jaber y S. M. B. M. Shapiai. «Performance Comparison of Metaheuristic Optimization Algorithms in Solving Production Scheduling Problems». En: *Journal of Intelligent Computation and Technology* 1.1 (2021), págs. 1-11.
- [20] M. B. M. Kamal y M. A. M. Ali. «A comparative study of metaheuristics algorithms based on their performance of complex benchmark problems». En: *Decision Making: Applications in Management and Engineering* 6.1 (2023). Accedido: 24 de octubre de 2025.
- [21] William Braham White, Black, and Gray-Box Modelling. University of Pennsylvania. 2025. URL: <https://www.design.upenn.edu/work/white-black-and-gray-box-modelling>.
- [22] M. Rügner y col. «Physics-based machine learning predictions despite or in combination with scarce data». En: *PM&SA* 1.1 (2021). URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8069582>.
- [23] S. E. W. T. M. H. van der Meer. «Hybrid Modelling by Machine Learning Corrections of Analytical Model Predictions towards High-Fidelity Simulation Solutions». En: *ResearchGate* (2021). URL: https://www.researchgate.net/publication/350813252_Hybrid_Modelling_by_Machine_Learning_Corrections_of_Analytical_Model_Predictions_towards_High-Fidelity_Simulation_Solutions.
- [24] Y. Tao y col. «Multi-objective optimization of water distribution networks based on non-dominated sequencing genetic algorithm». En: *PLoS ONE* 17.11 (2022). URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0277954>.
- [25] Various. «Optimization of Water Distribution Networks Using Genetic Algorithm Based SOP-WDN Program». En: *MDPI* 14.6 (2022). URL: <https://www.mdpi.com/2073-4441/14/6/851>.
- [26] Larry W. Mays. *Water Distribution Systems Handbook*. McGraw-Hill, 2001. Cap. Hydraulic Principles of Pipe Flow.
- [27] J. Keller y D. Karmeli. «Trickle Irrigation Design Parameters». En: *Journal of the Irrigation and Drainage Division* 101.IR3 (1975), págs. 275-291.
- [28] Various. *Genetic Algorithms for Combinatorial Optimization Problems*. Inf. téc. Ghent University, 2005. URL: <https://backoffice.biblio.ugent.be/download/1147278/1147528>.