

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)
Jurado 3 (pertenencia)

Ciudad de Mendoza, noviembre de 2025

Resumen

La presente memoria describe el desarrollo de un sistema avanzado para la optimización del riego en una plantación de pistachos, con el objetivo de generar un cronograma de despacho de riego óptimo para una empresa agrícola. El sistema busca balancear objetivos clave como la uniformidad del riego, la eficiencia del bombeo y la sostenibilidad del acuífero mediante una función de costos unificada.

Para superar las limitaciones de los modelos físicos, se creó un gemelo digital híbrido de alta precisión combinando modelado físico con técnicas de Inteligencia Artificial (IA) para corregir errores residuales. Además, se utilizó un Algoritmo Genético, un algoritmo de búsqueda avanzado, para resolver eficientemente el complejo Problema de Programación Mixta-Entera No Lineal (MINLP) asociado al despacho de riego.

Índice general

Resumen	I
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
Función objetivo ponderada	9
Restricciones del problema	10
2.3. Teoría de optimización: algoritmos genéticos	10
2.3.1. Anatomía de un algoritmo genético	10
Codificación del cromosoma (representación de la solución)	10
Operadores de selección	11
Operadores de cruce (<i>crossover</i> / recombinación)	11
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
Subsistema de entrenamiento y ciclo de vida (MLOps)	15
Subsistema de optimización y servicio	17
3.2. Implementación del gemelo digital híbrido	17
3.3. Implementación del optimizador de despacho (GA)	17
3.4. Implementación de funciones de costo y penalizaciones	17
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

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 proyecto es la capa de gestión y optimización.	16

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

Capítulo 1

Introducción

En este capítulo se presenta una introducción general del trabajo. Se describe la problemática fundamental del riego solar y se analiza el estado del arte en las áreas claves del modelado de sistemas y su optimización. Finalmente, se expone la justificación de la solución propuesta y se delimitan los objetivos y el alcance del desarrollo.

1.1. Contexto y problemática del riego solar

La implementación de sistemas de riego 100 % solares es una solución tecnológica de amplio uso en el sector agrícola, particularmente en plantaciones que operan de forma autónoma (es decir, en modo *off-grid*[1]), donde la red eléctrica de distribución pública no está disponible o su conexión resulta económicamente inviable. Estas instalaciones se componen típicamente de una capa física, que cuenta con una bomba sumergible y válvulas de campo, y una capa de control y adquisición, que cuenta con un variador de frecuencia (VFD) y un PLC que comanda dicho hardware. En la operación estándar, esta arquitectura de control es robusta para ejecutar comandos, pero carece de la capa superior de optimización que define el plan de operación.

Esta ausencia de una capa de gestión inteligente traslada por completo la carga de optimización al operario. El despacho de agua se convierte en una consecuencia pasiva de la meteorología, siguiendo la curva de irradiación solar, o, en el mejor de los casos, sigue un plan estático (basado en temporizadores) definido manualmente por el ingeniero agrónomo. Este enfoque manual representa una carga significativa de horas de ingeniería y es inherentemente subóptimo, ya que no puede reaccionar a la variabilidad diaria de las condiciones hídricas o energéticas.

La necesidad de esta optimización inteligente se vuelve crítica en contextos agrícolas como el del cultivo de pistacho, en que los ciclos de inversión son largos y la planta requiere varios años para entrar en producción [2, 3, 4]. En regiones donde el agua es un recurso escaso o sujetas a sequías estacionales, una gestión deficiente del riego no solo reduce la eficiencia, sino que compromete la viabilidad y sostenibilidad a largo plazo de la plantación.

Por lo tanto, la problemática central que este trabajo aborda es esta falta de inteligencia operativa. La proliferación de la telemetría y los sensores de campo (IoT) en la agricultura moderna han generado un vasto potencial de optimización que,

en la mayoría de los sistemas de riego, permanece sin explotar [5]. Los controladores actuales se limitan a registrar estos datos o a reaccionar a umbrales simples, pero carecen de la capacidad de utilizarlos para una planificación proactiva.

El verdadero desafío que se resuelve en este trabajo es integrar la inteligencia artificial para dos tareas claves: primero, para transformar este flujo de datos en conocimiento predictivo (un modelo de alta fidelidad del sistema); y segundo, para actuar sobre dicho conocimiento. Se requiere un motor de optimización, en este caso, un algoritmo genético, que funcione como un buscador capaz de generar un cronograma de despacho óptimo a largo plazo. Esta capacidad de búsqueda y planificación proactiva, que balancea múltiples objetivos, es la pieza de gestión que falta en los sistemas actuales, que obligan a operar muy por debajo de su potencial de eficiencia.

1.2. Estado del arte: modelado y optimización

El problema central de este trabajo, generar un cronograma de despacho de riego óptimo para un sistema solar autónomo, no es un desafío singular, sino la intersección de dos conceptos técnicos que el estado del arte ha abordado históricamente por vías separadas:

1. El desafío del modelado: se requiere un simulador de muy alta fidelidad, un gemelo digital, que actúe como la función de evaluación del optimizador. Este modelo debe predecir con precisión la respuesta hidráulica (caudales, presiones, nivel freático) ante cualquier plan de riego candidato.
2. El desafío de la optimización: se necesita un algoritmo de búsqueda capaz de encontrar la combinación óptima de decisiones (qué válvula abrir y cuándo) dentro de un espacio de soluciones de dimensionalidad masiva, respetando un conjunto de restricciones físicas (energía, agua, presión) y un tiempo de cómputo estricto.

A continuación, se analiza el panorama de las soluciones existentes para cada uno de estos desafíos, identificando las limitaciones que motivan el desarrollo de este trabajo.

1.2.1. El modelado: consistencia vs. precisión

El algoritmo de optimización es, por naturaleza, agnóstico a la física del problema. Depende estrictamente de una función de evaluación [6] (en este caso, el gemelo digital) para cuantificar la calidad, o *fitness*, de un cronograma de riego candidato. El éxito de la optimización, por lo tanto, depende enteramente de la fidelidad de este modelo. El enfoque tradicional para construir esta función se basa en modelos de caja blanca o de primeros principios, como el estándar industrial EPANET [7], que resuelven las ecuaciones fundamentales de la mecánica de fluidos [8, 9]. Su fortaleza indiscutible es la consistencia física: al estar basados en leyes universales, son robustos y pueden generalizar su comportamiento a escenarios de operación novedosos [10]. Sin embargo, su debilidad es la incertidumbre en sus parámetros físicos. El rendimiento del modelo depende de una calibración precisa de variables estáticas (ej. rugosidad de tuberías) que son difíciles de medir y cambian con el tiempo [11]. Esto genera un error sistemático o residual entre la simulación idealizada y la realidad del campo.

Para solucionar esta brecha de precisión, en el extremo opuesto del espectro se encuentran los modelos puramente empíricos de caja negra (IA), que buscan aprender el comportamiento del sistema basándose únicamente en datos históricos de telemetría. La ventaja de este enfoque es la capacidad de alcanzar una alta precisión, capturando dinámicas complejas que el modelo físico ignora [12]. No obstante, esta solución es frágil: requiere volúmenes masivos de datos para cubrir todo el espacio operativo [13], no es robusta para extrapolar a escenarios no vistos y, crucialmente, no garantiza la consistencia física [13].

El panorama de las soluciones de modelado presenta, por lo tanto, un dilema fundamental: el ingeniero debe elegir entre la consistencia física de la caja blanca (sufriendo su imprecisión) o la precisión de la caja negra (sufriendo su falta de robustez). Esta dicotomía, junto con la alternativa conceptual de un modelo híbrido o de caja gris que busca unificar ambas [14, 15], se ilustra en la figura 1.1.

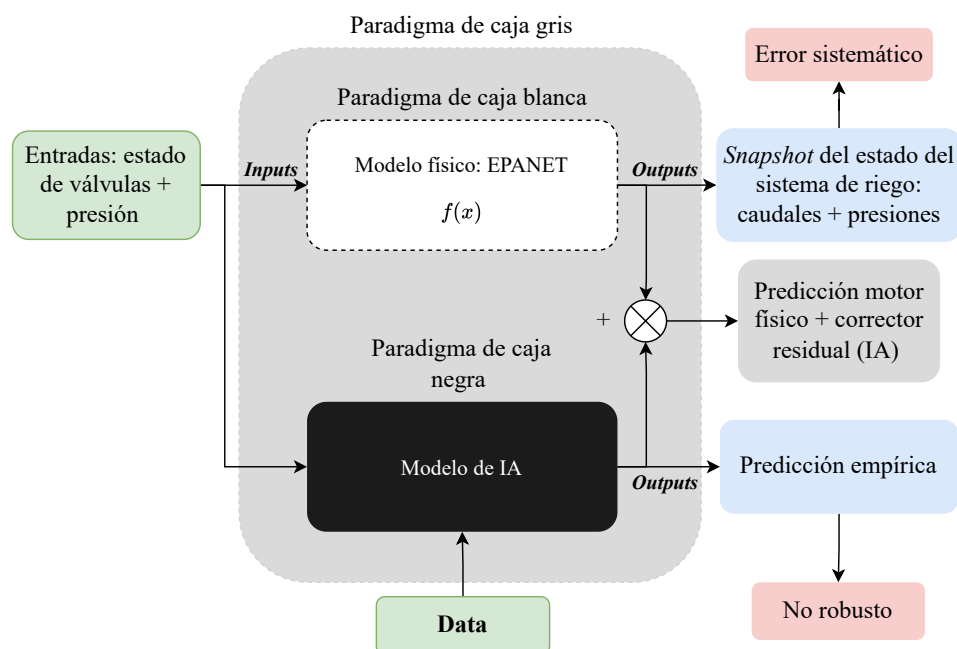


FIGURA 1.1. Diagrama conceptual de los paradigmas de modelado: caja blanca (física), caja negra (IA pura) y caja gris (híbrido).

1.2.2. Optimización: optimalidad vs. factibilidad

El segundo desafío fundamental radica en el propio algoritmo de búsqueda. La elección de la herramienta de optimización no es arbitraria, sino que está dictada por la naturaleza matemática inherente al problema del despacho de riego. Al formularlo, el problema se revela como un Problema de Programación Mixta-Entera No Lineal (MINLP)[16], una clasificación con profundas implicaciones computacionales.

La naturaleza Mixta-Entera (MI) surge de las variables de decisión: el estado de una válvula o bomba en un intervalo de tiempo es una decisión fundamentalmente binaria (0 para cerrado o 1 para abierto). A esto se suma la No Linealidad

(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

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.

Capítulo 2

Introducción específica

En este capítulo se presentan los fundamentos teóricos que sustentan la arquitectura de optimización del proyecto. Se detalla el modelado híbrido de caja gris, la formulación matemática del problema como un MINLP, la teoría de los Algoritmos Genéticos 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 volúmenes 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

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 es :

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

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

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

La literatura académica lo describe como una herramienta de predicción eficiente [28, 29], donde el modelo f_{fisico} 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_{fisico})

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, una relación empírica que define la pérdida de carga (H_f) en una tubería.

$$H_f = \frac{10,67 \cdot L}{C^{1,852} \cdot D^{4,87}} \cdot Q^{1,852} \quad (2.3)$$

2. Comportamiento de emisores (goteros): el caudal de salida (q_e) en un gotero se modela usando la ley de potencia, que relaciona el caudal con la presión (p) en el nodo. En el proyecto 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). A diferencia del Análisis Dirigido por Demanda (DDA), que asume que la demanda siempre se satisface, el PDA ajusta el caudal entregado si la presión en un nodo cae, ofreciendo 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 \mathbf{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

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. A continuación, 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.

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:

$$\max_{\mathbf{V}} (w_u \cdot \mathcal{U}(\mathbf{V}) - w_t \cdot \mathcal{T}_{bombeo}(\mathbf{V}) - w_a \cdot \mathcal{S}_{acuifero}(\mathbf{V})) \quad (2.5)$$

Donde :

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

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

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.

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

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

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

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.

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

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 [31, 32]. 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].

2.3.1. Anatomía de un algoritmo genético

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

Codificación del cromosoma (representación de la solución)

El cromosoma es la estructura de datos que codifica un plan de despacho de riego [33]. 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) [34].
- *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 [34].

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:

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

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

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"[35]. 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 selection (RWS)	Asigna una probabilidad de selección p_i proporcional a la aptitud f_i : $p_i = f_i / \sum f_k$ [35].	Simple.	Riesgo de convergencia prematura. Un individuo "súper" (con aptitud muy alta) puede dominar la población rápidamente, eliminando la diversidad genética [35].
Tournament selection (TOS)	Elige k (ej. $k = 2$) individuos al azar de la población y selecciona al de mayor aptitud para la reproducción [35].	Computacionalmente eficiente; menor riesgo de convergencia prematura [36, 37].	Si k es muy grande converge más rápido pero con riesgo de caer en óptimos locales.
Rank selection (LRS)	Selecciona individuos basándose en el <i>ranking</i> de su aptitud, no en el valor absoluto. Asigna probabilidades linealmente al ranking [35, 38].	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 [36, 37].

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) [32,

[35]. 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 real)	o 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 real)	o 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 [35].	A menudo es el más efectivo para codificaciones binarias, ya que permite una alta recombinación de bloques de genes.

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 [32, 35].

- *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 ($0 \rightarrow 1$ ó $1 \rightarrow 0$).
- *swap mutation* (para codificación por permutación): Se seleccionan dos genes al azar en el cromosoma y se intercambian sus posiciones [39].

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 [40, 27, 41]. La filosofía es transformar un problema de optimización *restringida* en un problema *no restringido* [40, 42]. Esto se logra moviendo las restricciones desde el *solver* hacia la *función objetivo* [40]. Se añade un término de penalización a la función de aptitud, que castiga a las soluciones infactibles [43, 40]. La formulación matemática general para un problema de minimización es [40, 42]:

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

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 [42].

La eficacia del AG depende de cómo se diseña $p(\cdot)$ [43]. Las estrategias varían desde la "pena de muerte" (*death penalty*), que rechaza cualquier solución infactible, hasta enfoques más robustos como las penalizaciones dinámicas, donde la penalización aumenta con cada generación (ej. $p(V, t) = (\lambda \cdot t) \cdot V$) [43, 28].

Capítulo 3

Diseño e implementación

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.

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 [44]. 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.

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*) [45]. Este flujo de trabajo, fundamental para un entorno de producción, está orquestado por *pipelines* de *Apache Airflow* [46].

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.

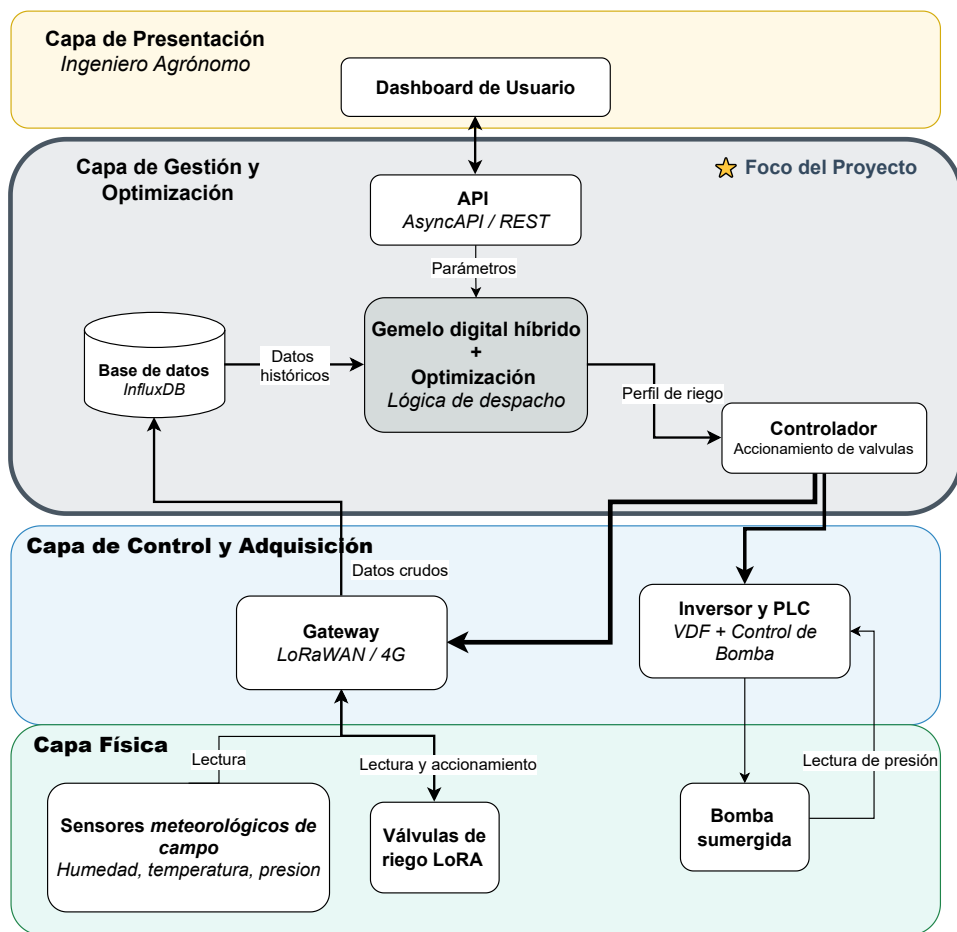


FIGURA 3.1. Diagrama de capas del sistema de riego, donde el foco del proyecto es la capa de gestión y optimización.

- Registro de Artefactos (*MLflow*): administra el ciclo de vida completo del modelo [47]. 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) [48].

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) [48].

Subsistema de optimización y servicio

Este subsistema constituye el corazón operativo del sistema, implementando el motor de decisión. Se diferencian dos partes importantes.

1. Capa de servicio de inferencia (*Snapshot API*): El diseño de arquitectura resuelve el desafío de rendimiento mediante un microservicio RESTful asíncrono implementado con *FastAPI*. Al iniciar, carga el gemelo digital híbrido, compuesto por el modelo físico WNTR [49] y el modelo de IA de corrección de residuales, directamente en memoria. Esto transforma el complejo proceso de simulación en una llamada de servicio de baja latencia. La *API* está diseñada para ser escalable horizontalmente, utilizando *NGINX* como balanceador de carga. Esto permite al optimizador enviar de manera concurrente miles de peticiones de simulación por segundo, evaluando un gran número de soluciones candidatas en paralelo.
2. Motor de optimización (*algoritmo genético*): El motor implementa el AG. Su conexión con la *Snapshot API* se da a través de una interfaz que distribuye la evaluación de la función de aptitud de cada individuo (plan de riego) a través de las instancias del microservicio de inferencia.

3.2. Implementación del gemelo digital híbrido

3.3. Implementación del optimizador de despacho (GA)

3.4. Implementación de funciones de costo y penalizaciones

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

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

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 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).

- [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>. 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] Various. *Genetic Algorithms for Combinatorial Optimization Problems*. Inf. téc. Ghent University, 2005. URL: <https://backoffice.biblio.ugent.be/download/1147278/1147528>.
- [32] 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.
- [33] Various. «Variable-Length Chromosome Genetic Algorithm for Network Scheduling Problem». En: *PMC* (2021). URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8228978/>.
- [34] Various. «Performance of Genetic Algorithm with Different Encoding Scheme». En: *GJETA* (2025). URL: <https://gjeta.com/sites/default/files/GJETA-2025-0203.pdf>.
- [35] 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>.
- [36] 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.
- [37] 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>.
- [38] 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.
- [39] 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.
- [40] arXiv. *Inverse Problems and Data Assimilation*. <https://arxiv.org/abs/1810.06191>. Accessed: October 15, 2025. 2018.
- [41] 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>.
- [42] *Mass Balance Constraints - WEAP*. https://www.weap21.org/webhelp/mass_balance_constraints.htm. Accessed: October 15, 2025.
- [43] 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.

- [44] Sam Newman. *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media, 2015. ISBN: 9781491950357.
- [45] Thomas Schlachter y Felix Sager. «Hybrid modeling of physical systems using neural networks for residual correction». 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).
- [46] The Apache Software Foundation. *Apache Airflow Documentation*. <https://airflow.apache.org/docs/>. Consultado el 27 de octubre de 2025. 2024.
- [47] 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/>.
- [48] 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.
- [49] Taylor L. Shaw y col. «WNTR: A Water Network Tool for Resilience». En: *Journal of Computing in Civil Engineering* 32.2 (2018), pág. 04017086. DOI: [10.1061/\(ASCE\)CP.1943-5487.0000720](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000720).