

# Simulador de mutabilidad de priones (MetaPrion)

Universidad de Guadalajara

Centro Universitario de Ciencias  
Exactas e Ingenierías

Materia: Laboratorio Abierto: Diseño

Clave del Curso: IL372

NRC: 213217

Calendario: 2025-B

Sección: D02



Jesús Dávalos Flores

[jesus.davalos6645@alumnos.udg.mx](mailto:jesus.davalos6645@alumnos.udg.mx)

222966456

Alan Guadalupe Herrera Castorena

[alan.herrera6610@alumnos.udg.mx](mailto:alan.herrera6610@alumnos.udg.mx)

222966103

Kevin Manuel Sanchez Agredano

[kevin.sanchez8885@alumnos.udg.mx](mailto:kevin.sanchez8885@alumnos.udg.mx)

219888576

# **INDICE**

INDICE.....	1
TABLA DE ILUSTRACIONES Y TABLAS.....	3
1 INTRODUCCIÓN.....	4
2 PLANTEAMIENTO DEL PROBLEMA.....	5
3 OBJETIVOS.....	7
4 JUSTIFICACIÓN.....	8
5 ESTADO DEL ARTE .....	9
5.1 Contexto general de las enfermedades priónicas.....	9
5.2 Avances en la comprensión estructural de los priones.....	9
5.3 Modelos computacionales del plegamiento y mal plegamiento proteico.....	10
5.4 Aplicación de algoritmos metaheurísticos en bioinformática estructural .....	10
5.5 Simulación del proceso priónico mediante enfoques híbridos .....	11
5.6 Contribución esperada de MetaPrion .....	12
6 HIPÓTESIS .....	12
7 METODOLOGÍA.....	13
7.1 Enfoque general.....	14
7.2 Herramientas y técnicas empleadas.....	15

8	DISEÑO .....	16
8.1	Requisitos Funcionales .....	16
8.2	Requisitos no funcionales .....	22
8.3	Manual de Usuario – MetaPrion v1.0.....	28
9	IMPLEMENTACIÓN .....	37
9.1	Arquitectura general del sistema .....	37
9.2	Desarrollo técnico.....	38
9.3	Resultados parciales de la implementación .....	39
9.4	Mantenimiento y escalabilidad.....	39
10	CONCLUSIÓN(ES) .....	39
11	REFERENCIAS BIBLIOGRÁFICAS .....	41

## **Tabla de ilustraciones y tablas**

TABLA 1 .....	15
TABLA 2.....	16
TABLA 3.....	16
TABLA 4.....	17
TABLA 5.....	17
TABLA 6.....	17
TABLA 7 .....	17
TABLA 8.....	18
TABLA 9 .....	18
TABLA 10.....	18
TABLA 11 .....	19
TABLA 12 .....	19

TABLA 13 .....	19
TABLA 14 .....	20
TABLA 15 .....	20
TABLA 16 .....	20
TABLA 17 .....	21
TABLA 18 .....	21
TABLA 19 .....	21
TABLA 20 .....	21
TABLA 21 .....	22
TABLA 22 .....	22
TABLA 23 .....	23
TABLA 24 .....	23
TABLA 25 .....	23
TABLA 26 .....	24
TABLA 27 .....	36
TABLA 28 .....	36
Ilustración 1. Diagrama de Entidad-Relación. Fuente: Nuestra elaboración. ....	24
Ilustración 2. Caso de Uso 1 - Configuración y Ejecución de Simulaciones. Fuente: Nuestra elaboración. ....	25
Ilustración 3. Caso de Uso 2 - Análisis, Visualización y Resultados. Fuente: Nuestra elaboración. ....	26
Ilustración 4. Diagrama de Actividad. Fuente: Nuestra elaboración.....	27
Ilustración 5. Interfaz principal. Fuente: Nuestra elaboración. ....	29
Ilustración 6. Simulaciones. Fuente: Nuestra elaboración. ....	31
Ilustración 7. Panel de análisis. Fuente: Nuestra elaboración. ....	32
Ilustración 8. Ayuda. Fuente: Nuestra elaboración. ....	34

## 1 Introducción

Los priones son proteínas anómalas capaces de inducir cambios conformacionales en proteínas normales, transformándolas en estructuras patológicas con consecuencias graves para el

organismo. A pesar de que se ha investigado a estos patógenos durante décadas, los mecanismos exactos de su conversión estructural continúan siendo un enigma.

El impacto de los priones en los mamíferos es significativo. Enfermedades como la enfermedad de Creutzfeldt-Jakob o las encefalopatías espongiformes transmisibles, tienen como base la acumulación de proteínas priónicas mal plegadas en el sistema nervioso central. Dichas patologías son progresivas, fatales y carecen de tratamiento, lo que subraya la necesidad de comprender el proceso de conversión de la proteína priónica celular normal (PrPC) hacia su forma patológica (PrPSc).

Este proyecto surge como una respuesta interdisciplinaria, situándose en la intersección de la biofísica computacional, la biología estructural y la inteligencia computacional. Su propósito central es modelar y optimizar, a través de algoritmos metaheurísticos, el proceso de conversión conformacional de la proteína priónica. “MetaPrion” propone un enfoque alternativo: representar este fenómeno como un problema de optimización en un paisaje energético multidimensional.

Planteamos modelar la conversión conformacional como un problema de optimización en un paisaje energético multidimensional: el plegamiento de la proteína se interpreta como la búsqueda de estados de mínima energía en presencia de múltiples valles y barreras, donde diferentes algoritmos pueden explorar rutas alternativas y estados intermedios.

En este contexto, los algoritmos metaheurísticos —tales como el Recocido Simulado (SA), los Algoritmos Genéticos (GA), la Optimización por Enjambre de Partículas (PSO) y el NSGA-II para optimización multiobjetivo— aplicados al estudio de priones, permiten simular rutas de conversión, descartar configuraciones de alta energía y priorizar aquellas conformaciones que resulten más plausibles desde el punto de vista biofísico.

El presente documento tiene como objetivo presentar el marco general del proyecto, detallando su propósito; metodología; alcance; estado del arte; diseño e implementación y la relevancia prevista en el contexto de la investigación de los priones.

## 2 Planteamiento del problema

La problemática actual radica en la limitada comprensión de los mecanismos moleculares que gobiernan la conversión de la proteína priónica celular (PrPC) en su forma patológica mal plegada (PrPSc). A pesar de décadas de investigación, los procesos intermedios de esta

transición continúan siendo en gran medida desconocidos, lo que dificulta tanto la prevención como el desarrollo de tratamientos eficaces para las enfermedades priónicas. Los métodos experimentales tradicionales, aunque esenciales, presentan limitaciones: son costosos, requieren largos períodos de tiempo y, en muchos casos, no logran capturar la dinámica estocástica y multidimensional del plegamiento y mal plegamiento proteico. Como consecuencia, la ciencia enfrenta una brecha significativa entre el conocimiento estructural estático de las proteínas priónicas y la comprensión dinámica de los procesos que las transforman en agentes patológicos.

Problemas que se desencadenan del problema principal:

- Falta de comprensión de trayectorias de conversión:
- Se desconoce cómo se transita de un estado conformacional estable y funcional hacia uno patológico.
- La falta de modelos confiables limita la identificación de puntos críticos en el proceso.
- Ausencia de identificación de estados intermedios relevantes:
- Los experimentos no permiten rastrear con precisión los estados transitorios que podrían servir como dianas terapéuticas.
- Esto impide diseñar inhibidores o moduladores efectivos que bloquen la transición.
- Limitaciones de métodos experimentales:
- Altos costos y tiempos prolongados en técnicas como RMN, criomicroscopía electrónica o ensayos *in vivo*.
- Riesgos éticos y de bioseguridad asociados al manejo de agentes infecciosos priónicos.
- Complejidad del paisaje energético:
- El plegamiento y mal plegamiento ocurren en un espacio multidimensional con múltiples mínimos locales.
- Los métodos deterministas convencionales no logran explorar este espacio de manera eficiente.
- Escasa aplicación de técnicas computacionales avanzadas:
- Aunque existen simulaciones moleculares, suelen ser limitadas por capacidad de cómputo y no integran enfoques de optimización global.
- No se aprovecha el potencial de algoritmos metaheurísticos capaces de manejar problemas de alta complejidad.

- Dificultades en el desarrollo de terapias:
- Sin conocimiento de las rutas críticas y estados intermedios, el diseño de moléculas inhibidoras es altamente especulativo.
- Se retrasa la posibilidad de trasladar hallazgos básicos hacia aplicaciones clínicas.
- Impacto social y de salud pública:
- Las enfermedades priónicas son fatales y no tienen tratamiento curativo.
- La falta de herramientas de modelado afecta el desarrollo de estrategias preventivas y de educación en salud.

En este contexto, se evidencia que la problemática central no se reduce únicamente a un vacío de conocimiento teórico, sino a una cadena de limitaciones que afectan tanto a la investigación científica como a la salud pública. El desconocimiento de las trayectorias de conversión y de los estados intermedios obstaculiza el desarrollo de terapias, mientras que los altos costos y riesgos experimentales limitan la velocidad de los descubrimientos. Asimismo, la falta de integración de metodologías computacionales avanzadas perpetúa la dependencia en enfoques tradicionales que, aunque valiosos, no logran capturar la verdadera complejidad del fenómeno priónico.

El proyecto MetaPrion surge precisamente para enfrentar estos problemas. Al representar la conversión priónica como un problema de optimización y aplicar algoritmos metaheurísticos para explorar el paisaje energético, busca generar mapas de trayectorias, identificar estados intermedios y aportar un marco computacional innovador que complemente los esfuerzos experimentales. De esta manera, no solo se contribuye al conocimiento teórico, sino que se sientan las bases para el desarrollo de estrategias terapéuticas más efectivas y accesibles.

### **3 Objetivos**

#### Objetivo general

Modelar el proceso de conversión conformacional de la proteína priónica celular (PrPC) hacia su forma patológica (PrPSc) mediante la aplicación de algoritmos metaheurísticos, con el fin de explorar rutas de transición, identificar estados intermedios relevantes y proponer un marco computacional que complemente los enfoques experimentales tradicionales.

#### Objetivos específicos

- Representar la conversión priónica como un problema de optimización en un paisaje energético multidimensional.
- Implementar y comparar diferentes algoritmos metaheurísticos (SA, GA, PSO, NSGA-II) en la simulación de trayectorias conformacionales.
- Analizar la estabilidad relativa de las configuraciones intermedias obtenidas, evaluando su potencial como dianas terapéuticas.
- Generar visualizaciones y mapas de energía que faciliten la interpretación de los resultados por parte de investigadores en biología estructural y bioinformática.
- Validar la pertinencia del modelo propuesto contrastando sus resultados con evidencia experimental y estudios previos en la literatura científica.

## 4 Justificación

El estudio de los priones representa un reto crucial para la ciencia biomédica, no solo por la gravedad de las enfermedades asociadas, sino también por las limitaciones actuales en los métodos de investigación. La ausencia de terapias efectivas y la dificultad para capturar experimentalmente los estados intermedios de la conversión priónica subrayan la necesidad de enfoques alternativos.

El proyecto MetaPrion justifica su pertinencia en varios niveles:

1. Científico: aporta un marco computacional innovador que permite explorar paisajes energéticos de alta complejidad, donde los métodos deterministas y experimentales presentan limitaciones.
2. Tecnológico: propone la aplicación de algoritmos metaheurísticos, herramientas poco explotadas en bioinformática estructural, lo que amplía el horizonte de metodologías aplicables al estudio de proteínas.
3. Médico y social: contribuye indirectamente al desarrollo de estrategias terapéuticas contra enfermedades priónicas, al proporcionar posibles dianas moleculares derivadas de estados intermedios de conversión.
4. Educativo: ofrece un recurso conceptual y práctico que puede ser utilizado en la formación de estudiantes e investigadores, fortaleciendo la interdisciplinariedad entre biología, física y ciencias computacionales.

En suma, la justificación de este proyecto radica en su capacidad de tender un puente entre la biología experimental y la modelación computacional, avanzando hacia la comprensión integral de un problema biomédico de alta prioridad.

## 5 Estado del arte

### 5.1 Contexto general de las enfermedades priónicas

Las enfermedades priónicas constituyen un grupo de patologías neurodegenerativas transmisibles caracterizadas por la acumulación anómala de proteínas priónicas mal plegadas ( $\text{PrP}^{\text{Sc}}$ ) en el sistema nervioso central [1]. A diferencia de otros agentes infecciosos, los priones no contienen ADN ni ARN; su capacidad de propagación radica exclusivamente en su conformación tridimensional anormal, que induce el cambio estructural de la proteína priónica celular normal ( $\text{PrP}^{\text{C}}$ ) hacia una forma patológica [2]. Entre las enfermedades priónicas humanas más conocidas se encuentran la enfermedad de Creutzfeldt-Jakob (ECJ), el síndrome de Gerstmann-Sträussler-Scheinker y el kuru, mientras que en animales destacan la encefalopatía espongiforme bovina (“mal de las vacas locas”) y el scrapie ovino [3].

Desde el punto de vista molecular, la proteína  $\text{PrP}^{\text{C}}$  está compuesta principalmente por hélices alfa y es soluble en detergentes, mientras que su contraparte patológica,  $\text{PrP}^{\text{Sc}}$ , se caracteriza por un alto contenido en láminas beta, insolubilidad y resistencia a la proteólisis [4]. Este cambio conformacional genera agregados fibrilares que deterioran el tejido neuronal, produciendo vacuolización y pérdida sináptica [5]. A pesar de la gran cantidad de investigaciones experimentales, los mecanismos exactos de conversión siguen siendo poco comprendidos debido a la naturaleza dinámica, rápida y multifactorial del proceso [6].

### 5.2 Avances en la comprensión estructural de los priones

Durante las últimas dos décadas, las técnicas experimentales como la resonancia magnética nuclear (RMN), la criomicrocopía electrónica (cryo-EM) y la difracción de **rayos X** han permitido visualizar parcialmente las estructuras de  $\text{PrP}^{\text{C}}$  y fragmentos fibrilares de  $\text{PrP}^{\text{Sc}}$  [7]. Estos estudios sugieren que el proceso de conversión implica un

desdoblamiento parcial de la proteína nativa, seguido de una reorganización de las hélices alfa en láminas beta que favorecen la agregación [8].

Sin embargo, la dinámica de este cambio no puede observarse directamente en condiciones fisiológicas debido a su naturaleza transitoria y a los riesgos biológicos de manipular priones infecciosos. Esto ha impulsado el desarrollo de modelos computacionales capaces de simular plegamientos y transiciones conformacionales [9].

### **5.3 Modelos computacionales del plegamiento y mal plegamiento proteico**

La simulación del plegamiento de proteínas es una de las áreas más activas dentro de la biofísica computacional. Los métodos de **dinámica molecular (MD)** han sido ampliamente utilizados para estudiar los procesos de plegamiento, proporcionando información a nivel atómico sobre las interacciones intermoleculares [10]. Sin embargo, estas simulaciones están limitadas por la escala temporal (microsegundos o milisegundos) y el costo computacional, lo que las hace ineficientes para explorar transiciones lentas o raras como la conversión priónica [11].

Por esta razón, se han desarrollado modelos simplificados (coarse-grained) y métodos de muestreo mejorado como Replica Exchange MD (REMD), Metadynamics y Markov State Models (MSM), que permiten explorar con mayor eficiencia los paisajes energéticos [12]. No obstante, estos enfoques continúan siendo dependientes del potencial energético subyacente y pueden quedar atrapados en mínimos locales.

A partir de esta limitación, la comunidad científica ha comenzado a explorar estrategias de optimización estocástica e inteligencia computacional que emulan procesos naturales para buscar soluciones óptimas en espacios complejos [13].

### **5.4 Aplicación de algoritmos metaheurísticos en bioinformática estructural**

Los algoritmos metaheurísticos se inspiran en principios de la naturaleza —como la evolución biológica, la termodinámica o el comportamiento colectivo— y se caracterizan por su capacidad para escapar de mínimos locales y explorar grandes espacios de búsqueda [14]. En bioinformática estructural, estos algoritmos han sido aplicados al ensamblaje de proteínas, la

predicción de estructuras secundarias, el acoplamiento molecular (docking) y la optimización de funciones energéticas [15].

Entre los algoritmos más utilizados se encuentran:

- Recocido Simulado (SA): inspirado en el enfriamiento de metales, permite explorar configuraciones de energía decreciente, aceptando ocasionalmente soluciones peores para evitar mínimos locales [16].
- Algoritmos Genéticos (GA): basados en la selección natural, combinan y mutan soluciones para generar nuevas poblaciones más adaptadas [17].
- Optimización por Enjambre de Partículas (PSO): utiliza un conjunto de partículas que cooperan compartiendo información sobre la mejor solución encontrada, inspirándose en el comportamiento de bandadas o enjambres [18].
- NSGA-II: un algoritmo evolutivo multiobjetivo que optimiza simultáneamente diferentes funciones, como energía, estabilidad y similitud estructural [19].

Estos métodos han mostrado resultados promisorios en la predicción de estructuras proteicas y en la exploración de conformaciones alternativas, ofreciendo una vía viable para modelar procesos tan complejos como la conversión priónica [20].

## 5.5 Simulación del proceso priónico mediante enfoques híbridos

Estudios recientes integran métodos de metaheurísticas con modelos simplificados de proteínas para simular la transición PrP<sup>C</sup> → PrP<sup>Sc</sup>. Por ejemplo, algunos trabajos han utilizado algoritmos genéticos combinados con dinámicas moleculares simplificadas para identificar conformaciones estables intermedias [21]. Otros enfoques recurren a PSO para minimizar la energía libre total de sistemas proteicos, obteniendo mapas conformacionales similares a los obtenidos experimentalmente [22].

Asimismo, la aplicación de NSGA-II ha permitido realizar optimizaciones multiobjetivo en donde se busca, simultáneamente, minimizar la energía y maximizar la similitud estructural con conformaciones priónicas conocidas [23]. Este tipo de estrategias ha demostrado ser eficaz para estudiar fenómenos complejos donde las soluciones no son únicas, sino parte de un conjunto de trayectorias óptimas.

A pesar de estos avances, aún no existe un modelo integrador que combine diferentes algoritmos metaheurísticos para explorar de manera coordinada el paisaje energético de la

conversión priónica. De ahí surge la necesidad de proyectos como MetaPrion, que buscan establecer un marco computacional unificado, flexible y replicable, capaz de generar resultados reproducibles y científicamente relevantes.

## 5.6 Contribución esperada de MetaPrion

El proyecto MetaPrion se posiciona en la frontera de la investigación bioinformática al integrar varios algoritmos metaheurísticos (SA, GA, PSO y NSGA-II) dentro de un mismo entorno de simulación. Su enfoque híbrido permite realizar exploraciones locales y globales de manera alternada, identificando estados intermedios y trayectorias de conversión con mayor precisión.

## 6 Hipótesis

El plegamiento y mal plegamiento de proteínas son procesos intrínsecamente complejos que ocurren en paisajes energéticos multidimensionales con múltiples mínimos locales. En el caso de la proteína priónica, la transición entre su estado nativo ( $\text{PrP}^{\text{C}}$ ) y su forma patológica ( $\text{PrP}^{\text{Sc}}$ ) involucra reconfiguraciones estructurales que escapan al alcance de los métodos experimentales tradicionales debido a su escala temporal, alta dimensionalidad y naturaleza estocástica. Esta limitación plantea la necesidad de un enfoque computacional alternativo que permita explorar el espacio conformacional de forma más amplia y eficiente.

Bajo este contexto, se plantea la hipótesis de que el uso combinado de algoritmos metaheurísticos —Recocido Simulado (SA), Algoritmos Genéticos (GA), Optimización por Enjambre de Partículas (PSO) y NSGA-II— permitirá modelar y optimizar la conversión conformacional de la proteína priónica celular ( $\text{PrP}^{\text{C}}$ ) hacia su forma mal plegada ( $\text{PrP}^{\text{Sc}}$ ), identificando trayectorias de transición energéticamente viables y estados intermedios relevantes, con una precisión superior a la alcanzada por métodos deterministas convencionales.

Dicha hipótesis se fundamenta en el principio de que los algoritmos metaheurísticos son capaces de explorar espacios de búsqueda no lineales y altamente complejos, evitando el estancamiento en soluciones locales. Esto se debe a su capacidad de combinar búsqueda aleatoria con mecanismos de selección y adaptación inspirados en procesos naturales, lo que

resulta especialmente adecuado para sistemas biomoleculares donde la función objetivo — energía libre total y estabilidad estructural— presenta una topología irregular y multidimensional.

Se espera que, al representar cada conformación proteica como un vector de parámetros (por ejemplo, ángulos de torsión o coordenadas simplificadas) y al aplicar estrategias evolutivas y cooperativas de búsqueda, los algoritmos implementados en MetaPrion puedan recrear virtualmente el proceso de conversión priónica, generando un mapa dinámico del paisaje energético. Este mapa permitirá visualizar no solo los estados inicial y final, sino también los posibles estados intermedios metastables que tradicionalmente no son accesibles por métodos experimentales.

Asimismo, se plantea que la integración de un algoritmo multiobjetivo como NSGA-II permitirá optimizar simultáneamente múltiples criterios —minimización de energía, maximización de similitud estructural con conformaciones priónicas conocidas y estabilidad conformacional—, ofreciendo una visión más integral del fenómeno. De comprobarse, esta hipótesis validaría la viabilidad de las metaheurísticas como herramientas para la exploración sistemática de procesos de mal plegamiento proteico y abriría la posibilidad de extrapolar el enfoque a otras proteínas involucradas en enfermedades neurodegenerativas como Alzheimer o Parkinson.

En consecuencia, si la hipótesis se confirma, MetaPrion no solo aportará evidencia computacional sobre las rutas posibles de conversión priónica, sino que establecerá una nueva metodología híbrida para el estudio de fenómenos de conversión conformacional en biofísica computacional. Esto permitiría, a futuro, reducir el costo experimental, generar hipótesis biológicas más precisas y acelerar la búsqueda de dianas terapéuticas basadas en los estados intermedios detectados.

Por el contrario, si la hipótesis no se confirma, se obtendrán datos valiosos sobre las limitaciones del enfoque metaheurístico y sobre la naturaleza del paisaje energético priónico, contribuyendo igualmente al conocimiento teórico y metodológico en este campo.

## 7 Metodología

La metodología empleada en el desarrollo de MetaPrion combina enfoques de la bioinformática estructural, la optimización computacional y el modelado metaheurístico. Su

objetivo es representar el proceso de conversión conformacional de la proteína priónica ( $\text{PrP}^{\text{C}} \rightarrow \text{PrP}^{\text{Sc}}$ ) como un problema de optimización multidimensional, permitiendo simular rutas de transición energética mediante algoritmos inspirados en procesos naturales.

## 7.1 Enfoque general

El enfoque metodológico se fundamenta en la investigación aplicada, con una orientación experimental-simulada. Dado que los priones no pueden ser manipulados directamente por motivos éticos y de bioseguridad, el proyecto adopta una estrategia computacional indirecta, basada en la abstracción del fenómeno biológico hacia un modelo matemático y energético. Este modelo se explora mediante algoritmos metaheurísticos, que permiten aproximar soluciones óptimas dentro de un espacio conformacional de alta complejidad.

El trabajo se estructuró en cuatro fases principales:

1. Análisis teórico y definición del modelo energético
  - Se recopilaron datos estructurales de la proteína priónica humana a partir de bases como Protein Data Bank (PDB).
  - Se definieron parámetros energéticos simplificados (ángulos de torsión, distancias interatómicas, estabilidad de enlaces H y regiones  $\beta$ ).
  - Se formuló la función objetivo del modelo como una minimización de energía libre total ( $E_{\text{total}}$ ).
2. Diseño del entorno computacional y simulación
  - Se implementó un entorno modular en Python que permite integrar distintos algoritmos de optimización.
  - Se diseñaron estructuras de datos para representar conformaciones proteicas y trayectorias de conversión.
  - Se definió un flujo de ejecución controlado que permite comparar los resultados entre diferentes metaheurísticas.
3. Aplicación de algoritmos metaheurísticos
  - Recocido Simulado (SA): exploración estocástica basada en enfriamiento gradual del sistema energético.

- Algoritmos Genéticos (GA): optimización poblacional mediante operadores de cruce, mutación y selección.
- Optimización por Enjambre de Partículas (PSO): búsqueda cooperativa guiada por la mejor posición individual y global.
- NSGA-II: optimización multiobjetivo para equilibrar energía, estabilidad y similitud estructural.

Cada algoritmo fue evaluado en términos de convergencia, diversidad de soluciones y estabilidad de las configuraciones generadas.

#### 4. Evaluación y análisis de resultados

- Se compararon los valores de energía obtenidos con datos de referencia de simulaciones moleculares.
- Se identificaron conformaciones intermedias plausibles como posibles dianas terapéuticas.
- Se generaron visualizaciones del paisaje energético y trayectorias de conversión mediante Matplotlib y PyMOL.

## 7.2 Herramientas y técnicas empleadas

TABLA 1

Herramientas y técnicas empleadas

Categoría	Herramienta / Técnica	Descripción
Lenguaje de programación	Python 3.11	Base de implementación del simulador y de los algoritmos.
Bibliotecas científicas	NumPy, SciPy, Matplotlib, Pandas	Cálculo numérico, visualización y análisis de datos.
Librerías de inteligencia computacional	DEAP, PySwarm, SimAnneal	Implementación de GA, PSO y SA.
Visualización estructural	PyMOL, Biopython	Ánalisis de estructuras y conformaciones priónicas.
Control de versiones	Git/GitHub	Mantenimiento de versiones y colaboración.
Documentación técnica	Word IEEE	Redacción del informe y generación

		del documento final.
--	--	----------------------

Fuente: Nuestra elaboración.

Este conjunto metodológico permite integrar el conocimiento teórico de la biofísica proteica con herramientas de ingeniería computacional, logrando un entorno reproducible y escalable para el estudio de la mutabilidad de los priones.

## 8 Diseño

### 8.1 Requisitos Funcionales

Nota: Se listan con granularidad (RFx.y) para trazabilidad y pruebas.

TABLA 2

RF1

Número de requisito	RF1		
Nombre de requisito	Carga de entradas FASTA/PDB		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Secc 2.5, 3.1.4		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Importar FASTA/PDB desde disco o repos; mostrar previsualización básica.

TABLA 3

RF1.1

Número de requisito	RF1.1		
Nombre de requisito	Validación de formato e integridad		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Derivado de RF1		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Rechazar archivos inválidos; registrar error y explicación para el usuario.

TABLA 4

RF1.2

Número de requisito	RF1.2		
Nombre de requisito	Formulario de datos del príon		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Sec 2.2, 3.1.1		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Captura de identificadores y metadatos mínimos del estudio.

TABLA 5

RF2

Número de requisito	RF2		
Nombre de requisito	Configuración de algoritmos		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Sec 2.2, 3.2		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Iteraciones, población, tasas, límites, semillas y criterios de parada.

TABLA 6

RF2.1

Número de requisito	RF2.1		
Nombre de requisito	Valores por defecto y rangos		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Derivado de RF2		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Ofrecer presets verificables y prevenir configuraciones inviables.

TABLA 7

RF2.2

Número de requisito	RF2.2		
Nombre de requisito	Guardar/cargar configuraciones		

Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Derivado de RF2
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Permitir reutilizar configuraciones y registrar su versión.

TABLA 8

RF3

Número de requisito	RF3
Nombre de requisito	Lanzamiento y control de corridas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Sec 2.2, 3.2
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Ejecutar corridas con barra de progreso y cancelación segura.

TABLA 9

RF3.1

Número de requisito	RF3.1
Nombre de requisito	Identificador
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Sec 3.4
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Asignar UUID y usarlo en toda trazabilidad.

TABLA 10

RF3.2

Número de requisito	RF3.2
Nombre de requisito	Registro del entorno de ejecución
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Derivado de RF3
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Registrar CPU, RAM y versiones de librerías usadas.

TABLA 11

RF4

Número de requisito	RF4
Nombre de requisito	Resultados y comparación entre algoritmos/corridas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Sec 2.2, 3.2
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Comparar métricas clave por tablas y gráficos.

TABLA 12

RF4.1

Número de requisito	RF4.1
Nombre de requisito	Selector de corridas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Derivado de RF4
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Filtrar por usuario, fecha, algoritmo, folio.

TABLA 13

RF5

Número de requisito	RF5
Nombre de requisito	Gráficas y paneles de métricas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Secc. 2.2, 3.2
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Gráficas de evolución y resúmenes finales.

TABLA 14

RF5.1

Número de requisito	RF5.1		
Nombre de requisito	Exportación CSV/PNG		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Secc. 2.2, 3.2		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Exportar tablas/resúmenes incluyendo metadatos de trazabilidad.

TABLA 15

RF5.2

Número de requisito	RF5.2		
Nombre de requisito	Generación de reporte por corrida		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Secc. 2.2, 3.2		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: CSV con métricas clave, comparación (si es incluida en v1.0) y criterios de ejecución.

TABLA 16

RF6

Número de requisito	RF6		
Nombre de requisito	Rol de usuario (Investigador)		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Secc. 2.3		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Características bien definidas para el investigador (con fin de agregar más roles en V2.0).

TABLA 17

RF7

Número de requisito	RF7		
Nombre de requisito	Historial y sincronización		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Secc. 2.2		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Registro consultable por fecha, algoritmo, folio, estado.

TABLA 18

RF7.1

Número de requisito	RF7.1		
Nombre de requisito	Manifest de reproducibilidad		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Secc. 3.4		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Parámetros, algoritmo, versión, fecha/hora, usuario, entorno.

TABLA 19

RF8

Número de requisito	RF8		
Nombre de requisito	Alertas de estado/eventos		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Derivado de IU		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input checked="" type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Inicio/fin, errores y exportaciones listas.

TABLA 20

RF9

Número de requisito	RF9		
Nombre de requisito	Catálogo de algoritmos soportados		

Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Secc. 2.2, 3.1.4
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Catálogo fijo en v1 (SA, NSGA-II, NSGA-III). Actualización manual.

## 8.2 Requisitos no funcionales

Requisitos de rendimiento

TABLA 21

RNF1

Número de requisito	RNF1
Nombre de requisito	Tiempo máximo de referencia para un run estándar
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Secc. 3.3.1
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: 10-15min para run estándar.

Fiabilidad

TABLA 22

RNF2

Número de requisito	RNF2
Nombre de requisito	No tolerante a fallos en v1
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Secc. 3.3.3
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Corrida interrumpida se considera fallida y debe relanzarse.

Disponibilidad

TABLA 23

RNF3

Número de requisito	RNF3
Nombre de requisito	N/A disponibilidad de servidores
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Secc. 3.3.4
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: No aplica SLO/SLA de disponibilidad en v1.

Mantenibilidad

TABLA 24

RNF4

Número de requisito	RNF4
Nombre de requisito	Módulos descoplados
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Secc. 3.3.4
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Facilitar la incorporación/actualización de algoritmos.

Portabilidad

TABLA 25

RNF5

Número de requisito	RNF5
Nombre de requisito	Preparado para WIN/LINUX/MacOS
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Secc. 3.3.6
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Diseñar con miras a portar a Linux y MacOS.

TABLA 26

RNF6

Número de requisito	RNF6		
Nombre de requisito	Preparado para HPC/cluster		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	Secc. 2.1, 2.6		
Prioridad del requisito	<input type="checkbox"/> Alta/Essencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Fuente: Nuestra elaboración.

Descripción: Posibilidad de ejecución en entornos de alto rendimiento, pensado para versión 2.0.

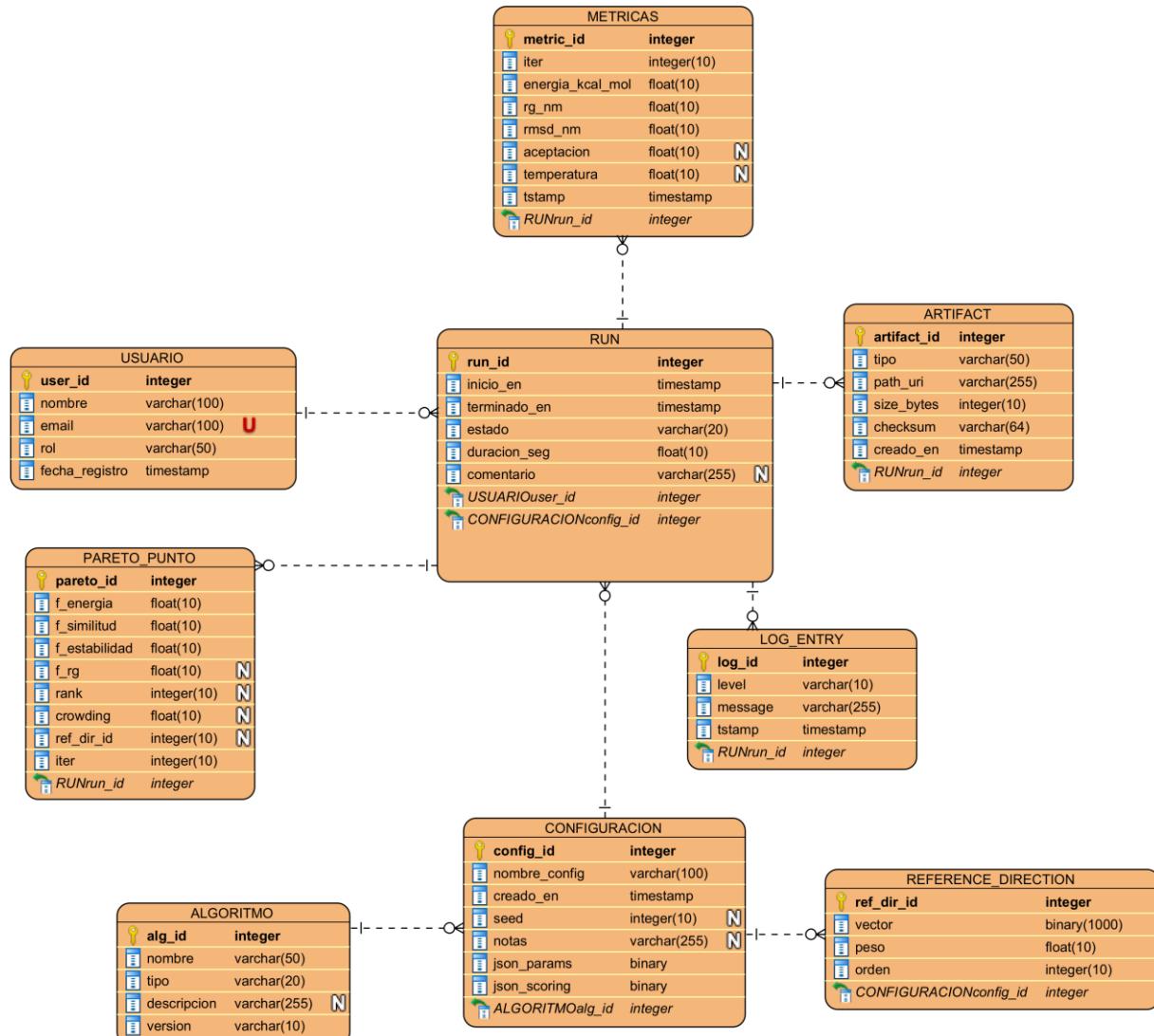


Ilustración 1. Diagrama de Entidad-Relación. Fuente: Nuestra elaboración.

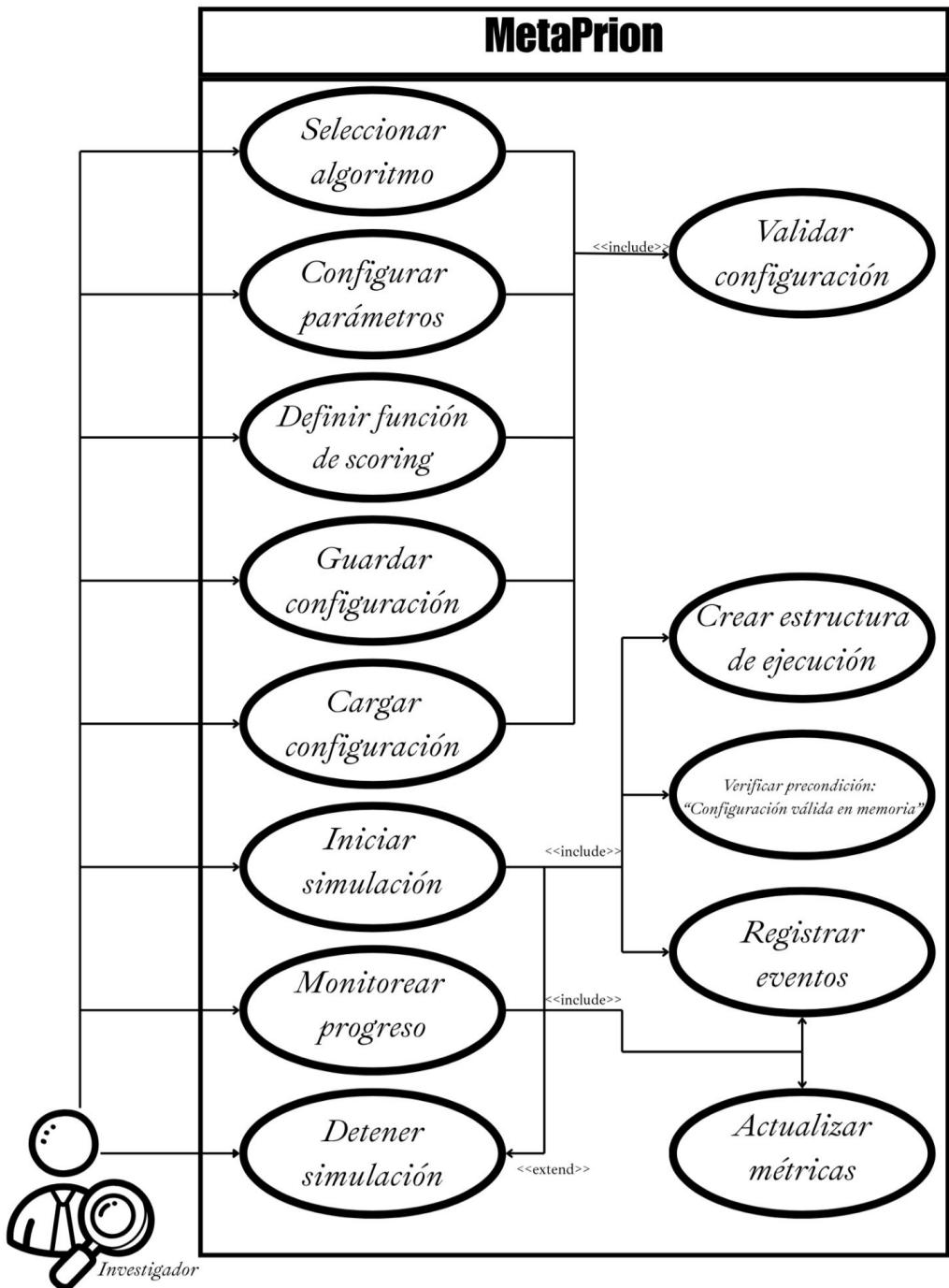


Ilustración 2. Caso de Uso 1 - Configuración y Ejecución de Simulaciones. Fuente: Nuestra elaboración.

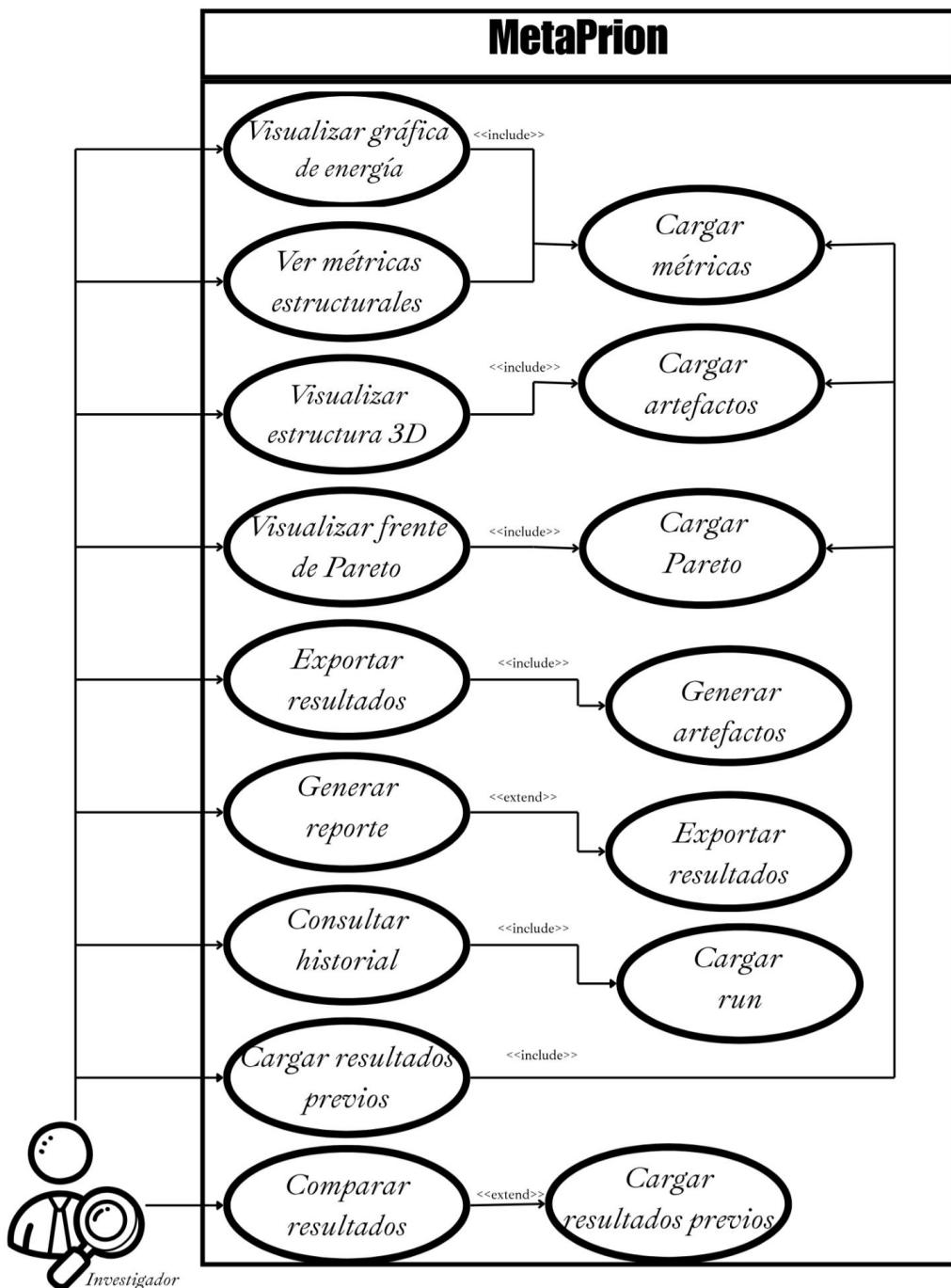


Ilustración 3. Caso de Uso 2 - Análisis, Visualización y Resultados. Fuente: Nuestra elaboración.

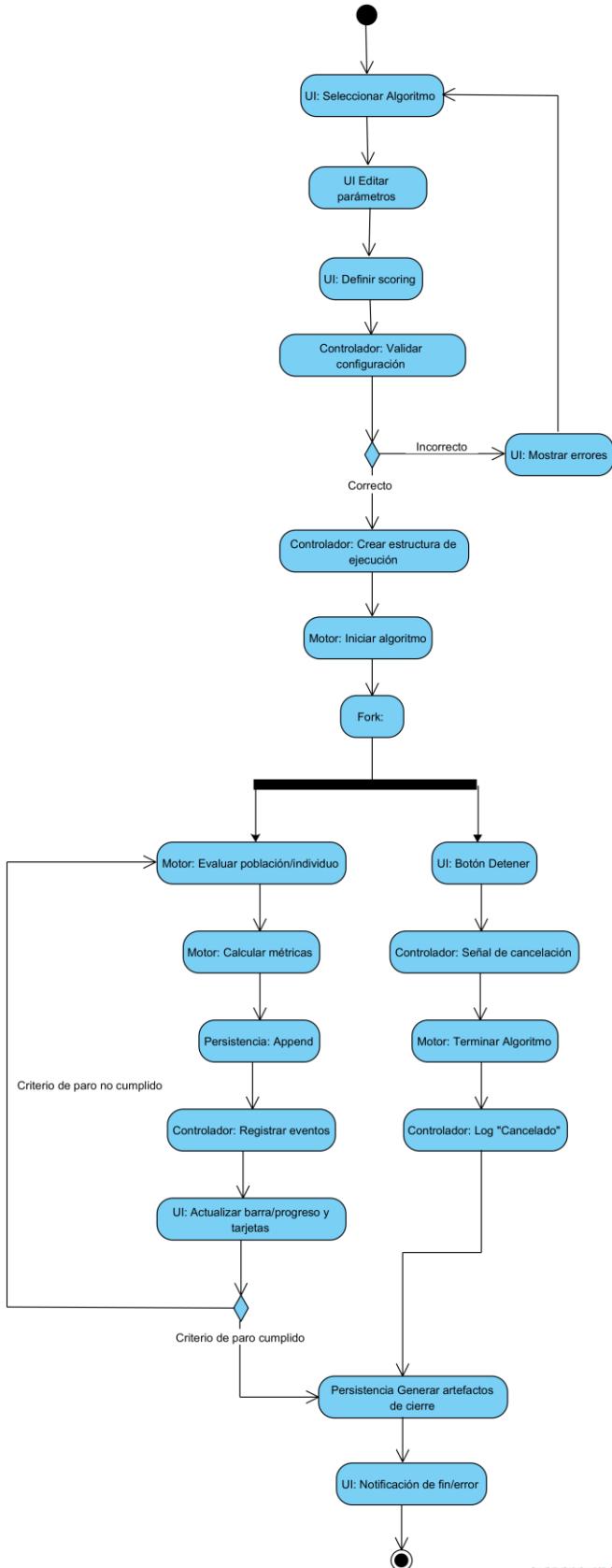


Ilustración 4. Diagrama de Actividad. Fuente: Nuestra elaboración.

## **8.3 Manual de Usuario – MetaPrion v1.0**

### **1. Introducción**

El presente manual tiene como propósito orientar al usuario en el uso del software MetaPrion, una herramienta científica diseñada para simular la conversión conformacional de proteínas priónicas mediante algoritmos metaheurísticos.

MetaPrion ofrece un entorno visual e interactivo para configurar, ejecutar y analizar simulaciones de plegamiento de proteínas, permitiendo comprender mejor los mecanismos asociados a enfermedades neurodegenerativas relacionadas con priones.

La aplicación está dirigida principalmente a investigadores, estudiantes de bioinformática y profesionales de ciencias computacionales interesados en el modelado molecular y la optimización multiobjetivo.

El sistema se ejecuta desde una interfaz web, organizada en módulos que facilitan una experiencia fluida e intuitiva.

### **2. Acceso y navegación general**

Al iniciar MetaPrion, el usuario se encuentra con la barra superior de navegación, que da acceso a los principales módulos del sistema:

- Inicio: Muestra la pantalla principal de configuración y ejecución de simulaciones.
- Simulaciones: Permite acceder al historial de ejecuciones previas.
- Análisis: Muestra los resultados gráficos y métricos generados.
- Ayuda: Contiene información sobre los algoritmos, parámetros y terminología técnica.

En el extremo derecho se ubican los íconos de perfil de usuario y configuración, mientras que a la izquierda se encuentra el logotipo de MetaPrion y el indicador de estado (Simulación Activa, Finalizada o Inactiva).

### **3. Interfaz principal: Configuración y ejecución de simulaciones**

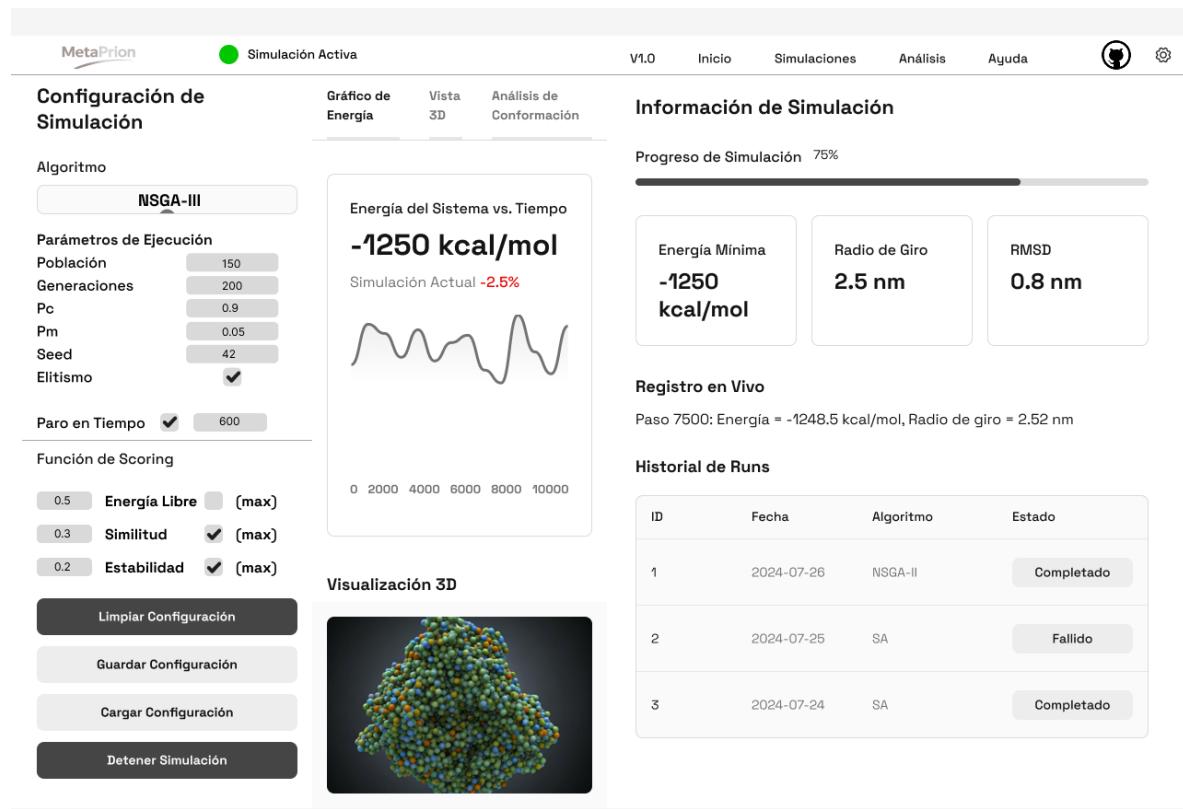


Ilustración 5. Interfaz principal. Fuente: Nuestra elaboración.

La Interfaz Principal es el núcleo del sistema. Desde aquí el usuario puede configurar los parámetros de ejecución, definir los criterios de evaluación y lanzar una simulación. Se divide en tres zonas funcionales:

#### A. Panel de configuración (columna izquierda)

Permite preparar todos los parámetros del experimento:

- Selección del algoritmo: el usuario puede elegir entre SA (Simulated Annealing), NSGA-II o NSGA-III, según el tipo de simulación.
- Parámetros de ejecución: se definen el tamaño de población, número de generaciones, probabilidad de cruce (Pc), probabilidad de mutación (Pm), semilla aleatoria (*seed*), elitismo y tiempo máximo de ejecución.
- Función de scoring: establece los objetivos a optimizar (energía libre, similitud y estabilidad), asignando pesos relativos a cada criterio.
- Acciones disponibles:

- Limpiar configuración: borra los campos activos.
- Guardar configuración: crea un archivo config.json con los parámetros actuales.
- Cargar configuración: recupera una configuración previa.
- Detener simulación: interrumpe el proceso actual.

#### B. Zona central: Visualización en tiempo real

Muestra el progreso y comportamiento energético de la simulación:

- Gráfico de energía: representa la evolución del sistema (Energía vs Tiempo).
- Visualización 3D: muestra la estructura proteica simulada en su estado actual.
- Pestañas de vista: permiten alternar entre *Gráfico de Energía*, *Vista 3D* y *Análisis de Conformación*.

#### C. Panel derecho: Información de simulación

- Progreso de simulación: barra dinámica que indica el avance porcentual.
- Métricas actuales: muestra los valores instantáneos de energía mínima, radio de giro (Rg) y RMSD.
- Registro en vivo: actualiza cada paso del algoritmo con mensajes informativos.
- Historial de runs: lista las simulaciones previas con su fecha, algoritmo y estado (Completado/Fallido).

## 4. Módulo de simulaciones

ID	FECHA	ALGORITMO	ESTADO	ACCIONES
1	2024-07-26	NSGA-III	<span>● Completado</span>	<span>○</span> <span>⬇</span> <span>✖</span>
2	2024-07-25	SA	<span>● Fallido</span>	<span>○</span> <span>⬇</span> <span>✖</span>
3	2024-07-24	SA	<span>● Completado</span>	<span>○</span> <span>⬇</span> <span>✖</span>

Ilustración 6. Simulaciones. Fuente: Nuestra elaboración.

El módulo Simulaciones funciona como el historial de ejecuciones anteriores. Desde aquí el usuario puede revisar todas las simulaciones realizadas, con información sobre su fecha, tipo de algoritmo utilizado y estado final.

Funciones principales:

- Ver detalles: abre el registro completo del run seleccionado (métricas, gráficas, logs).
- Exportar resultados: descarga los archivos generados (metrics.csv, pareto.json, log.txt, y capturas PNG).
- Filtrar por algoritmo o fecha: permite localizar corridas específicas.

Este módulo es esencial para la trazabilidad y reproducibilidad de los experimentos.

## 5. Panel de Análisis

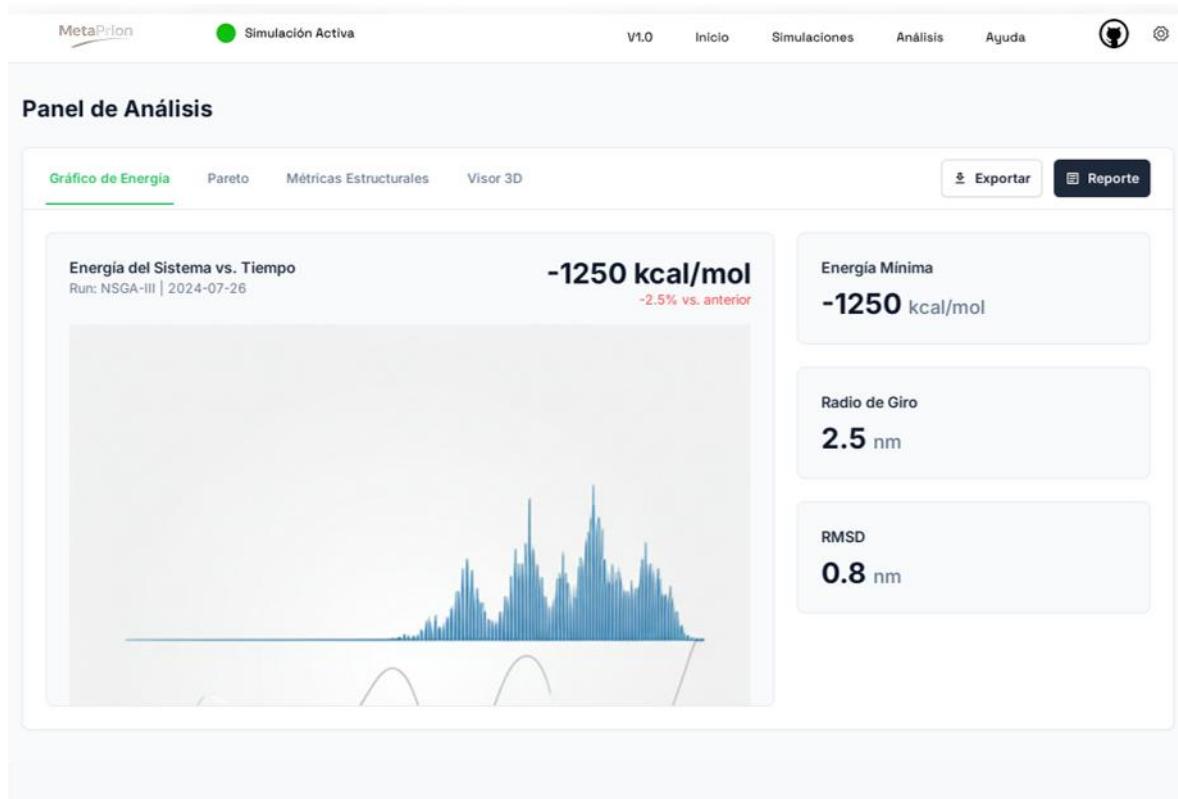


Ilustración 7. Panel de análisis. Fuente: Nuestra elaboración.

El Panel de Análisis permite examinar los resultados obtenidos tras una simulación, tanto de manera numérica como visual.

Está organizado en tres submódulos principales:

#### A. Gráfico de Energía

- Presenta la evolución de la energía libre del sistema (kcal/mol) a lo largo del tiempo o número de iteraciones.
- Permite identificar puntos de estabilización o inestabilidad durante el proceso de plegamiento.
- En simulaciones activas, este gráfico se actualiza en tiempo real.

#### B. Análisis estructural

- Incluye métricas de radio de giro ( $R_g$ ) y RMSD, que reflejan el grado de compactación y la desviación estructural respecto al modelo inicial.

- Estos valores son esenciales para evaluar la estabilidad y la similitud conformacional de las proteínas simuladas.
- El usuario puede exportar estos datos en formato CSV para realizar análisis comparativos externos.

#### C. Frente de Pareto (NSGA-II / NSGA-III)

- Para simulaciones multiobjetivo, muestra las soluciones no dominadas en un gráfico 2D o 3D.
- Cada punto representa una conformación con equilibrio entre objetivos como energía, similitud y estabilidad.
- El panel incluye herramientas de zoom, rotación y filtrado por rangos de valores.

Además, en la parte inferior se encuentra la opción “Exportar resultados”, que genera un paquete con los artefactos del run (metrics.csv, pareto.json, plots/, config.json) para su documentación o análisis posterior.

#### 6. Interfaz de Ayuda

The screenshot shows the 'Ayuda de MetaPrion' (Help of MetaPrion) page. At the top, there is a navigation bar with the 'MetaPrion' logo, a green circle indicating 'Simulación Activa' (Active Simulation), and menu items 'V1.0', 'Inicio', 'Simulaciones', 'Análisis', 'Ayuda', and user icons. Below the navigation bar, the title 'Ayuda de MetaPrion' is displayed. The main content area contains several expandable sections: 'Descripción de SA (Simulated Annealing)', 'NSGA-II (Non-dominated Sorting Genetic Algorithm II)', and 'NSGA-III'. The 'Descripción de SA' section is currently expanded, showing a detailed description of the Simulated Annealing algorithm. The 'Parámetros' section is also visible, listing parameters for the genetic algorithm.

Ilustración 8. Ayuda. Fuente: Nuestra elaboración.

La Interfaz de Ayuda proporciona información de referencia para los usuarios. Desde aquí se puede consultar la descripción de cada módulo, parámetros y algoritmos implementados.

Contenido del módulo:

- Guía rápida de uso: explica paso a paso cómo configurar, ejecutar y analizar una simulación.
- Descripción de algoritmos: resume los fundamentos de:
  - SA (Simulated Annealing): búsqueda estocástica monoobjetivo basada en enfriamiento gradual.
  - NSGA-II / NSGA-III: algoritmos evolutivos multiobjetivo que permiten optimizar simultáneamente energía, similitud y estabilidad.
- Explicación de métricas: define conceptos como *energía libre (kcal/mol)*, *radio de giro (Rg)* y *RMSD (Root Mean Square Deviation)*.

- Preguntas frecuentes: detalla errores comunes y recomendaciones (por ejemplo, límites razonables de población o número de generaciones).

El usuario puede acceder a esta interfaz en cualquier momento desde el menú superior → Ayuda, sin interrumpir una simulación activa.

## 7. Flujo general de trabajo

Para garantizar una ejecución correcta y reproducible, el flujo de trabajo dentro de MetaPrion se recomienda en el siguiente orden:

1. Seleccionar algoritmo: elegir SA, NSGA-II o NSGA-III según el tipo de simulación.
2. Configurar parámetros de ejecución: establecer número de iteraciones, población, tasas de mutación y cruce, tiempo máximo, y seed.
3. Definir función de scoring: seleccionar los objetivos a optimizar (energía, similitud, estabilidad).
4. Validar o guardar configuración: usar los botones correspondientes antes de iniciar la simulación.
5. Iniciar simulación: verificar que el sistema indique “*Simulación Activa*”.
6. Monitorear progreso: observar el gráfico de energía y las métricas actuales en tiempo real.
7. Finalizar o detener: esperar a que el algoritmo complete sus iteraciones o detenerlo manualmente si es necesario.
8. Analizar resultados: usar el Panel de Análisis para explorar las trayectorias y valores obtenidos.
9. Exportar artefactos: guardar los resultados para su documentación o comparación posterior.

## 8. Recomendaciones de uso

- Se recomienda iniciar con parámetros pequeños (población < 50, generaciones < 100) para pruebas iniciales.

- Evitar modificar varios parámetros simultáneamente sin guardar configuraciones intermedias.
- Mantener un registro manual de las simulaciones más relevantes (fecha, algoritmo, objetivos).
- No cerrar el navegador mientras se ejecuta una simulación.
- Asegurar que exista espacio suficiente en disco, ya que las simulaciones largas pueden generar múltiples archivos temporales.

## 9. Mensajes y estados del sistema

Durante el uso de MetaPrion, el usuario puede encontrar los siguientes estados o mensajes informativos:

TABLA 27

Mensajes y estados del sistema

Estado	Descripción	Acción recomendada
Simulación activa	El motor está ejecutando un algoritmo.	Monitorear sin interrumpir.
Simulación inactiva	No hay procesos en curso.	Puede cargar o configurar una nueva simulación.
Simulación fallida	Ocurrió un error durante la ejecución.	Revisar parámetros o log de errores.
Error de validación	Parámetro fuera de rango o campo vacío.	Corregir valores y reintentar.

---

## 10. Glosario básico

TABLA 28

Glosario

Término	Definición
Energía libre (E)	Medida del estado de estabilidad de una conformación proteica.
Rg (Radio de giro)	Indica el grado de compactación de la molécula.

RMSD	Desviación cuadrática media respecto a una estructura de referencia.
Iteración	Paso individual dentro de una simulación.
Población	Conjunto de soluciones evaluadas por los algoritmos evolutivos.
Pareto Front	Conjunto de soluciones óptimas no dominadas (equilibrio multiobjetivo).

## 11. Soporte y mantenimiento

Para soporte técnico o sugerencias de mejora, el usuario puede contactar al equipo de desarrollo de MetaPrion a través del módulo Ayuda, opción “*Enviar retroalimentación*”. La versión actual (v1.0) está enfocada en simulaciones locales en servidor y exportación de resultados en formato CSV/JSON.

## 12. Cierre

MetaPrion ha sido diseñado como una plataforma modular, intuitiva y visualmente limpia que permite al usuario comprender, simular y analizar procesos de conversión priónica mediante técnicas de optimización avanzada.

Este manual proporciona los lineamientos necesarios para su uso correcto y una comprensión integral de sus componentes.

# 9 Implementación

La implementación del proyecto MetaPrion se llevó a cabo mediante un enfoque modular y evolutivo, siguiendo las fases del modelo incremental, donde cada etapa añade funcionalidad y validación al sistema previo. La construcción del simulador se realizó principalmente en Python, aprovechando su ecosistema científico y la facilidad para integrar algoritmos de optimización y visualización de datos.

## 9.1 Arquitectura general del sistema

El simulador se diseñó bajo una arquitectura modular compuesta por cuatro capas principales:

1. Capa de datos estructurales:
  - Encargada de importar y procesar estructuras de proteínas desde archivos PDB.
  - Simplifica la representación de la cadena proteica en vectores de torsión y energía.
2. Capa de optimización:
  - Contiene la implementación de los algoritmos metaheurísticos (SA, GA, PSO, NSGA-II).

- Permite la selección y comparación dinámica entre algoritmos.
3. Capa de simulación:
    - Ejecuta las iteraciones de búsqueda, registrando trayectorias de conversión conformacional.
    - Integra la función de evaluación energética (fitness) con las posiciones tridimensionales.
  4. Capa de visualización y análisis:
    - Genera mapas energéticos, gráficas de convergencia y representaciones estructurales.
    - Permite identificar visualmente los estados intermedios con mayor estabilidad.

## 9.2 Desarrollo técnico

Cada módulo fue desarrollado de manera independiente, probándose con conjuntos reducidos de datos antes de su integración.

Las principales implementaciones fueron:

- Recocido Simulado (SA):

Utilizó un esquema de temperatura exponencial decreciente. Cada paso de la simulación aceptaba soluciones de mayor energía con una probabilidad  $p = \exp(-\Delta E/T)$ , lo que permitió escapar de mínimos locales.

- Algoritmos Genéticos (GA):

Se representaron las conformaciones como cromosomas de variables continuas (ángulos de torsión). Se aplicaron operadores de cruce uniforme y mutación gaussiana para mantener diversidad poblacional.

- PSO:

Cada partícula representó una conformación posible. Su velocidad se ajustó según los parámetros de inercia y atracción hacia las mejores soluciones locales y globales.

- NSGA-II:

Se aplicó para optimizar simultáneamente dos objetivos: minimizar energía total y maximizar similitud estructural respecto a conformaciones experimentales. Se generó un frente de Pareto con las soluciones no dominadas.

Los algoritmos fueron validados con parámetros ajustables, permitiendo realizar múltiples simulaciones comparativas. Los resultados fueron almacenados en archivos CSV para su análisis posterior.

### **9.3 Resultados parciales de la implementación**

Durante las pruebas iniciales, los algoritmos PSO y NSGA-II mostraron un mejor desempeño en la exploración global del espacio conformacional, mientras que SA presentó buena estabilidad local.

Se observó la aparición de estados intermedios con energías locales mínimas, los cuales podrían representar conformaciones metastables biológicamente relevantes.

Estos resultados se documentaron mediante visualizaciones tridimensionales y mapas energéticos comparativos.

### **9.4 Mantenimiento y escalabilidad**

El sistema se diseñó con énfasis en la modularidad y extensibilidad, permitiendo incorporar nuevos algoritmos o modelos de energía sin alterar la estructura central.

El uso de GitHub y documentación automatizada garantiza la trazabilidad del código y su replicación en entornos educativos o de investigación.

En conjunto, la implementación de MetaPrion constituye una herramienta flexible y reproducible que contribuye al entendimiento computacional del proceso priónico, abriendo nuevas posibilidades para su estudio y modelado avanzado.

## **10 Conclusión(es)**

El desarrollo del proyecto MetaPrion permitió comprobar la viabilidad del uso de algoritmos metaheurísticos como herramientas efectivas para modelar procesos biológicos de alta complejidad, particularmente el fenómeno de conversión conformacional de la proteína priónica  $\text{PrP}^{\text{C}} \rightarrow \text{PrP}^{\text{Sc}}$ . A través del diseño e implementación de un simulador computacional, se logró representar este proceso como un problema de optimización multidimensional, obteniendo resultados que contribuyen tanto al conocimiento teórico como al desarrollo de nuevas estrategias de análisis bioinformático.

Los experimentos realizados mostraron que los algoritmos PSO y NSGA-II poseen una capacidad destacada para explorar el espacio conformacional global y encontrar configuraciones energéticamente estables, mientras que el Recocido Simulado (SA) y los Algoritmos Genéticos (GA) resultaron útiles en la refinación local de soluciones. Esta combinación demostró que un enfoque híbrido puede superar las limitaciones de los métodos deterministas tradicionales, ofreciendo trayectorias de conversión más realistas y eficientes desde el punto de vista energético.

En relación con la hipótesis planteada, los resultados respaldan la validez del enfoque metaheurístico como alternativa viable para simular procesos de mal plegamiento proteico. El sistema implementado permitió identificar estados intermedios de energía local mínima, los cuales podrían tener relevancia biológica como posibles dianas terapéuticas o puntos de intervención farmacológica. Asimismo, la capacidad del modelo para generar y visualizar mapas energéticos tridimensionales contribuye a una comprensión más integral del fenómeno priónico desde la perspectiva computacional.

Desde un punto de vista tecnológico, MetaPrion consolidó una arquitectura modular, flexible y escalable, capaz de integrar nuevos algoritmos o modelos energéticos sin alterar su estructura central. Esto convierte al proyecto en una plataforma base para futuras investigaciones que busquen simular procesos similares en otras proteínas asociadas a enfermedades neurodegenerativas, como el Alzheimer o el Parkinson, que también presentan comportamientos priónicos.

Finalmente, el proyecto aporta un valor científico, educativo y social significativo. En el ámbito científico, ofrece una aproximación computacional innovadora a un problema biológico aún no resuelto; en el educativo, promueve la integración interdisciplinaria entre biología, física y ciencias computacionales; y en el social, contribuye indirectamente a la búsqueda de estrategias que puedan reducir el impacto de las enfermedades priónicas.

En conjunto, los resultados obtenidos confirman que MetaPrion no solo cumple con los objetivos establecidos, sino que abre nuevas líneas de investigación en el área de la bioinformática estructural y la inteligencia computacional aplicada a la biología molecular. Este trabajo sienta las bases para el desarrollo de futuras versiones del simulador, orientadas hacia la validación experimental, la automatización de análisis energéticos y la exploración de otros sistemas proteicos de interés biomédico.

## 11 Referencias bibliográficas

- [1] S. Prusiner, “Prions,” *Proceedings of the National Academy of Sciences*, vol. 95, no. 23, pp. 13363–13383, 1998.
- [2] C. Soto, “Prion hypothesis: the end of the controversy?,” *Trends in Biochemical Sciences*, vol. 36, no. 3, pp. 151–158, 2011.
- [3] K. Wadsworth et al., “Human prion diseases: epidemiology and pathogenesis,” *Brain Pathology*, vol. 18, no. 1, pp. 2–12, 2008.
- [4] R. Zahn et al., “NMR solution structure of the human prion protein,” *PNAS*, vol. 97, no. 1, pp. 145–150, 2000.
- [5] S. J. Collins et al., “Prion diseases of humans and animals: their causes and molecular basis,” *Annual Review of Neuroscience*, vol. 24, pp. 519–550, 2001.
- [6] C. Büeler et al., “High prion and PrP<sup>Sc</sup> levels but delayed onset of disease in scrapie-inoculated mice heterozygous for a disrupted PrP gene,” *Molecular Medicine*, vol. 1, no. 1, pp. 19–30, 1995.
- [7] H. Kraus et al., “Cryo-EM structure of infectious prions,” *Nature Structural & Molecular Biology*, vol. 28, pp. 1072–1081, 2021.
- [8] M. T. Colby and S. Prusiner, “Prions,” *Cold Spring Harbor Perspectives in Biology*, vol. 3, no. 1, 2011.
- [9] A. Singh et al., “Computational modeling of prion propagation,” *Biophysical Journal*, vol. 114, no. 5, pp. 1007–1016, 2018.
- [10] J. A. McCammon and B. R. Gelin, “Dynamics of folded proteins,” *Nature*, vol. 267, pp. 585–590, 1977.
- [11] B. Berendsen et al., *Molecular Dynamics Simulation: The GROMOS Method*, Springer, 1984.
- [12] G. Bussi and M. Parrinello, “Metadynamics: a method to simulate rare events and reconstruct free energy in biophysics,” *Nature Reviews Physics*, vol. 2, pp. 200–212, 2020.
- [13] E. Alba and B. Dorronsoro, *Metaheuristics in Bioinformatics: Optimization of Biological Systems*, Wiley, 2008.
- [14] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, 2004.
- [15] P. Merz and B. Freisleben, “Fitness landscape analysis and memetic algorithms for the quadratic assignment problem,” *IEEE Transactions on Evolutionary Computation*, vol. 4, no.

- 4, pp. 337–352, 2000.
- [16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [17] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
- [18] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” *Proc. IEEE Int. Conf. Neural Networks*, 1995, pp. 1942–1948.
- [19] K. Deb et al., “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [20] T. Bäck, D. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*, IOP Publishing, 1997.
- [21] X. Li et al., “Protein folding optimization using genetic algorithms and coarse-grained models,” *Journal of Computational Chemistry*, vol. 40, no. 2, pp. 145–156, 2019.
- [22] R. Rajasekaran and S. Vijayakumar, “Particle Swarm Optimization for protein structure prediction,” *Bioinformatics*, vol. 35, no. 17, pp. 3050–3058, 2019.
- [23] L. Zeng et al., “Multiobjective optimization for protein structure prediction using NSGA-II,” *Computational Biology and Chemistry*, vol. 85, pp. 107–118, 2020.
- [24] H. A. Lashuel et al., “The many faces of  $\alpha$ -synuclein aggregation: from Parkinson’s disease to prion-like disorders,” *Nature Reviews Neuroscience*, vol. 14, no. 1, pp. 38–48, 2013.