



# Instituto Politécnico Nacional

## Escuela Superior de Cómputo



### Trabajo Terminal No. 2025-B065

#### Modelo para representar la interacción gravitacional de dos cuerpos

Programa de Ingeniería en Inteligencia Artificial (2020)

**Alumnos:**

Carrillo Barreiro José Emiliano  
Robles Ortero José Ángel

**Directores:**

Dr. Cesar Hernández Vasquez  
Dr. Mauricio Olguín Carbajal

20 de mayo de 2025

Ciudad de México

## 1 Introducción

## 2 Estado del Arte

## 3 Marco Teórico

## 4 Planeación

## 5 Análisis

## 6 Diseño

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Introducción

Las simulaciones de  $n^1$ -cuerpos, usadas en astrofísica para modelar interacciones galácticas y sistemas planetarios, tienen una restricción: **masas estáticas**. Esto limita la precisión al simular procesos de masa variable, un ejemplo de estos procesos es: la evolución estelar.

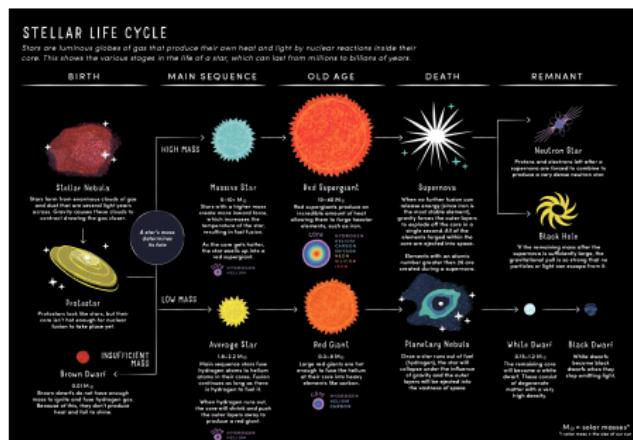
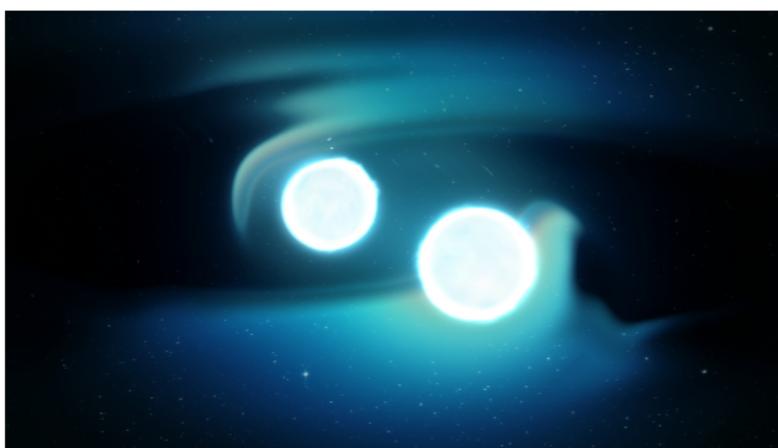


Figura 1: Diagrama del ciclo de evolución de una estrella. Propiedad de: [1]

<sup>1</sup> Durante toda la presentación se usará  $n = 2$ , dado el acotamiento que se tiene para el presente Trabajo Terminal.

## Planteamiento Del Problema

Un problema fundamental en las simulaciones de  $n$ -cuerpos es la masa invariable, lo que afecta la estabilidad de los cuerpos involucrados en fenómenos dinámicos como fusiones estelares o acreción. Esta rigidez limita tanto el estado del sistema de  $n$ -cuerpos como el potencial para aplicaciones interactivas, requiriendo métodos más flexibles.



**Figura 2:** Colisión de dos estrellas de neutrones (Simulación). Extraido de: [2]

# Propuesta de Solución

Proponemos un modelo de simulación de  $n$ -cuerpos con **modificación dinámica de masas**, de los cuerpos del sistema, en tiempo de ejecución. Se emplearán diversos enfoques de solución al problema de  $n$ -cuerpos para **eficiencia computacional**, y algoritmos bio-inspirados para el **ajuste paramétrico adaptativo**. El modelo, **escalable**, se optimizará para hardware accesible, buscando **superar las limitaciones de los modelos actuales**.

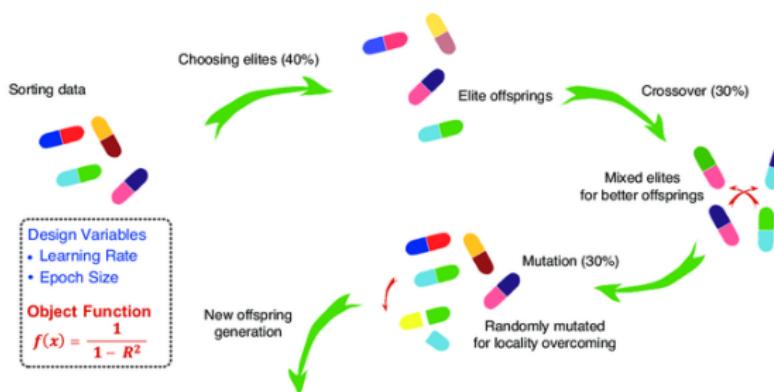
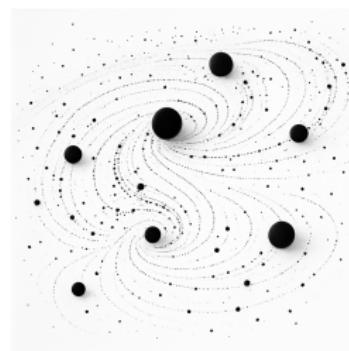


Figura 3: Ejemplo de Aplicación de un Algoritmo Genético (AG) Adaptado de: [3]

# Objetivos

## Objetivo General

Desarrollar un modelo teórico para la simulación del problema de dos cuerpos que permita la modificación dinámica de la masa, mejorando la estabilidad local del sistema visible en la representación de sus interacciones gravitacionales y eventos asociados.



**Figura 4:** Ejemplo de representación de un sistema de  $n$ -cuerpos. *Autoría Propia*

# Objetivos I

## Objetivo Específicos

### Módulo de simulación

Implementar el módulo de simulación encargado de integrar los distintos procedimientos algorítmicos requeridos para obtener la descripción numérica del sistema de interacción de  $n$  cuerpos, incluyendo la aplicación de métodos de integración numérica, el cálculo de fuerzas gravitatorias y la detección de colisiones.

### Módulo de optimización

Diseñar e implementar el módulo de optimización orientado al ajuste dinámico de las masas de los cuerpos que conforman el sistema de  $n$ -cuerpos, mediante el uso de algoritmos bioinspirados, con el fin de identificar el primer conjunto de valores que satisfaga las restricciones impuestas en cuanto a estabilidad y viabilidad del sistema.

# Objetivos II

## Objetivo Específicos

### Módulo de Simulación dinámica

Desarrollar e implementar el modelo computacional de para la simulación dinámica de un sistema de dos cuerpos bajo interacción gravitatoria.

### Módulo de visualización

Implementar el módulo de visualización gráfica para la representación dinámica del sistema simulado, mostrando su evolución temporal a lo largo de un conjunto limitado de iteraciones, a fin de apoyar la interpretación de los resultados del modelo

# Objetivos III

## Objetivo Específicos

### Interfaz básica (UI)

Diseñar e implementar una interfaz básica que permita el ingreso estructurado de parámetros asociados a los cuerpos del sistema, diferenciando entre atributos fijos (e.g., radio) y variables susceptibles de optimización (e.g., masa), así como la definición de restricciones obligatorias y optionales que condicionan el espacio de soluciones. Además de permitir visualizar los resultados generados por los módulos de simulación y optimización

# Justificación y Relevancia Científico-Tecnológica

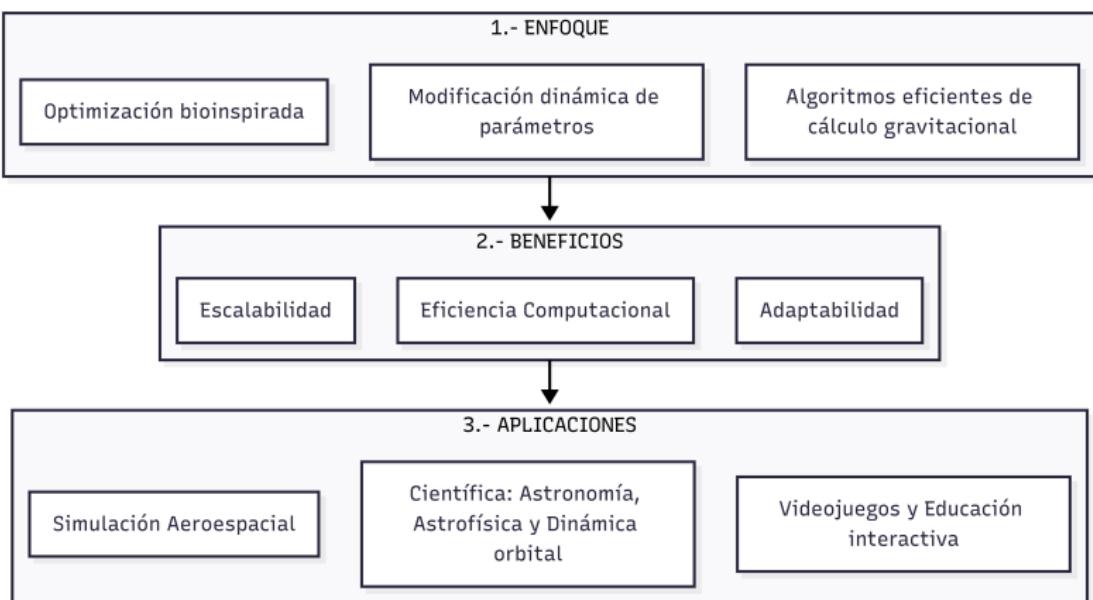


Figura 5: Diagrama representativo de los niveles de Justificación. Autoría Propia

## 1 Introducción

## 2 Estado del Arte

## 3 Marco Teórico

## 4 Planeación

## 5 Análisis

## 6 Diseño

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Estado del arte

Tabla 1: Comparativa contra soluciones disponibles

Producto o metodo	Características	Escalabilidad	Usa IA	Cambios dinamicos
ode_num_int	Framework C++11 modular para EDOs; orientado a pruebas de integradores.	Media	No	No
Representación Geométrica	Quadtrees/Octrees para geometría eficiente, no simula dinámica.	Alta	No	No
Método n-NNN	Simulación con n-vecinos y cirugía Hamiltoniana; inspirado en IA.	Alta	Sí	No
PKDGRAV3	Hidrodinámica sin malla (MFM/MFV); vecinos adaptativos.	Alta	No	No

# Estado del arte

Tabla 2: Comparativa contra soluciones disponibles

Producto o metodo	Características	Escalabilidad	Usa IA	Cambios dinamicos
<b>SPH/N-body Híbrido</b>	Interacciones gas-estrella con árbol Barnes-Hut y pasos bloque.	Alta	No	No
<b>Integrador Simpléctico</b>	Orden 2+, reversible; ideal para colisiones.	Media	No	No
<b>Solver TPM</b>	Combina PM y Tree según densidad local; altamente paralelizado.	Alta	No	No
<b>Algoritmo TPM</b>	Descomposición por densidad; integración multi-escala.	Alta	No	No

# Estado del arte

Tabla 3: Comparativa contra soluciones disponibles

Producto o metodo	Características	Escalabilidad	Usa IA	Cambios dinamicos
REBOUND	Modular; incluye varios integradores, colisiones y condiciones frontera.	Alta	No	No
Estabilidad Planetaria	Estudio con REBOUND; Gini vs. inestabilidad. No eventos internos.	Alta	No	No
Solución Propuesta	Combina FMM/Barnes-Hut para cálculo gravitacional eficiente con Algoritmos Bioinspirados para la <i>optimización y ajuste dinámico</i> de parámetros (masa) durante la simulación.	Alta	Sí	Sí

# Comparativa de Métodos N-cuerpos

Tabla 4: Comparativa de Métodos N-cuerpos

Método	Alta Escala	Modular	Usa IA	Dinámico*
Framework ode_num_int	✗	✓	✗	✗
Repr. Geométrica	✓	✗	✗	✗
Método n-NNN	✓	✗	✓	✗
PKDGRAV3	✓	✗	✗	✗
Híbrido SPH/N-body	✓	✓	✗	✗
Int. Simplicio	✗	✓	✗	✗
Solver Híbrido TPM	✓	✗	✗	✗
Algoritmo TPM	✓	✗	✗	✗
REBOUND	✓	✓	✗	✗
Est. Planetaria	✓	✗	✗	✗
<b>Solución Propuesta</b>	✓	✓	✓	✓



## 1 Introducción

## 2 Estado del Arte

## 3 Marco Teórico

Tecnologías: Lenguaje & Librerías.  
Métodos & Técnicas a utilizar.

## 4 Planeación

## 5 Análisis

## 6 Diseño

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

## 1 Introducción

## 2 Estado del Arte

## 3 Marco Teórico

Tecnologías: Lenguaje & Librerías.  
Métodos & Técnicas a utilizar.

## 4 Planeación

## 5 Análisis

## 6 Diseño

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Lenguaje de Programación

## Python



- Alto nivel y propósito general
- Sintaxis clara
- Multiparadigma
- Tipado dinámico
- Extensa biblioteca

Figura 6: Logo de Python *Adaptado de: [4]*

# Biblioteca de Simulación N-Cuerpos

## REBOUND

- Implementado en C
- Dinámica colisional y no colisional
- Integradores avanzados
- Algoritmos de gravedad
- Detección y resolución de colisiones



Figura 7: Logo de REBOUND  
Adaptado de: [5]

# Biblioteca de Optimización

## pymoo



Figura 8: Logo de pymoo *Adaptado de: [6]*

- Framework para optimización
- Amplia gama de algoritmos
- Alta flexibilidad
- Uso de herramientas específicas
- Soporte para paralelización

# Biblioteca de Visualización

## Matplotlib

- Biblioteca de visualización
- Gran flexibilidad y personalización
- Gráficos científicos
- Amplia documentación

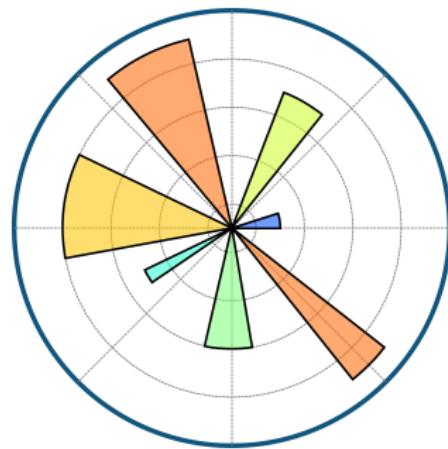


Figura 9: Logo de Matplotlib  
Adaptado de: [7]

# Biblioteca de GUI para Python

## PyQt



- Basado en Qt (C++)
- Widgets y *layouts*
- Integración con Matplotlib
- Soporte para multihilo
- Qt Designer

Figura 10: Logo de PyQt. *Adaptado de: [8]*

## 1 Introducción

## 2 Estado del Arte

## 3 Marco Teórico

Tecnologías: Lenguaje & Librerías.  
Métodos & Técnicas a utilizar.

## 4 Planeación

## 5 Análisis

## 6 Diseño

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Cálculo de Gravedad y Colisiones

Mediante módulos de REBOUND

- **Cálculo de Gravedad:**

- **Suma Directa:**  $O(N \cdot N_{\text{active}})$
- **Octree (Barnes-Hut):**  
 $O(N \log N)$

- **Detección de Colisiones:**

- **Búsqueda Directa:**  $O(N^2)$
- **Octree:**  $O(N \log N)$
- **Barrido Plano**

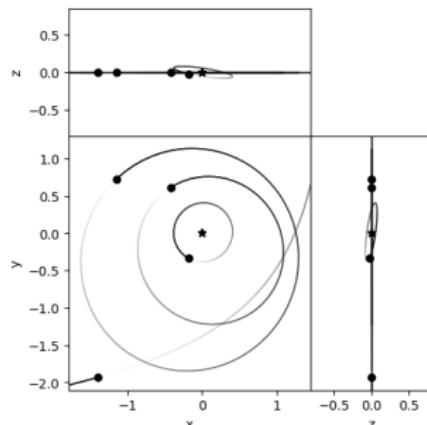


Figura 11: Simulación de órbitas usando REBOUND Adaptado de: [9]

# Algoritmo de Exploración

## Algoritmo Genético (AG) con pymoo

- Componentes principales:
  - Muestreo
  - Selección de Padres
  - Cruce
  - Mutación
  - Manejo de Restricciones

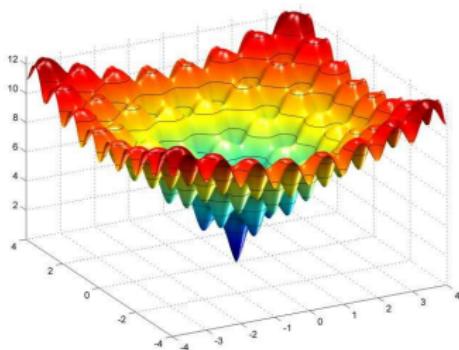


Figura 12: Función tipo benchmark, para la evaluación de AGs Adaptado de: [10]

# Indicador de Estabilidad

## Exponente de Lyapunov

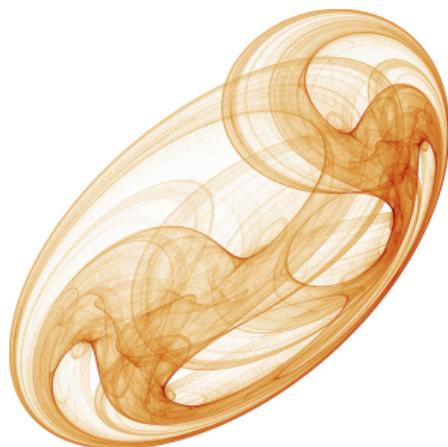
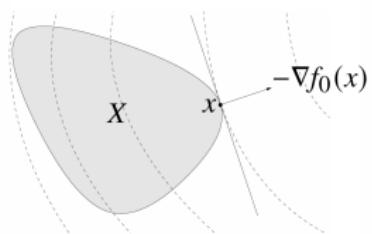


Figura 13: Atractor caótico generado mediante exponentes de Lyapunov. Adaptado de: [11]

- Cuantifica sensibilidad
- Indicador primario de estabilidad/caos
- Medida objetiva y cuantitativa
- Predice comportamiento a largo plazo
- Base matemática rigurosa

# Enfoque del Proyecto

## Factibilidad y Optimización



**Figura 14:** Gráfica 2D que muestre contornos de una función objetivo  $f_0(x)$ , una región factible sombreada definida por restricciones, y el punto óptimo  $x^*$ .  
Adaptado de: [12]

### • Problema de Factibilidad:

- Determinar si existen configuraciones estables
- Criterio clave: Exponente de Lyapunov  $\lambda_1 \leq$  umbral
- Restricciones físicas.

### • Marco de Optimización:

- Función objetivo: minimizar  $\lambda_1$  directamente
- Algoritmo Genético como herramienta exploratoria
- Permite búsqueda eficiente en el espacio de parámetros

## 1 Introducción

## 2 Estado del Arte

## 3 Marco Teórico

## 4 Planeación

## 5 Análisis

## 6 Diseño

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Juicio de Expertos: M.C. José Alberto Torres León

**Perfil del Experto** Egresado del Centro de Investigación en Cómputo (CIC) del Instituto Politécnico Nacional, sus áreas de especialización incluyen agentes inteligentes para videojuegos (jugadores y dinámicos), optimización evolutiva entre otros.

**Resumen de Opiniones y Observaciones Obtenidas** Tras la revisión del material y la discusión, el M.C. Torres León aportó las siguientes observaciones y valoraciones:

- Modificación de parámetros en tiempo de ejecución
- Simulaciones apegadas a la realidad
- Interacciones dinámicas

# Requerimientos Funcionales

Tabla 5: Requerimientos Funcionales

ID	Descripción	Prioridad	Actor	Criterios de Aceptación
RF-001	Modificación dinámica de la masa durante la simulación.	Alta	Usuario/Modelo	El usuario cambia la masa y la simulación responde adecuadamente.
RF-002	Simulación de interacciones gravitacionales entre dos cuerpos.	Alta	Modelo	Se calculan y representan correctamente las fuerzas gravitacionales.
RF-003	Uso de FMM y Barnes-Hut para optimización.	Media	Modelo	Se implementan y mejoran el rendimiento frente a métodos directos.
RF-004	Ajuste dinámico de parámetros mediante algoritmos bioinspirados.	Alta	Modelo	Mantienen la estabilidad del sistema durante cambios.
RF-005	Visualización gráfica de la evolución de los cuerpos.	Media	Usuario	Muestra trayectorias y estados claramente durante la simulación.
RF-006	Interfaz básica para modificar parámetros y ver resultados.	Media	Usuario	El usuario ajusta parámetros y observa los efectos.

# Requerimientos No Funcionales

Tabla 6: Requerimientos No Funcionales

ID	Descripción	Prioridad	Actor	Criterios de Aceptación
RNF-001	Optimización de la simulación para eficiencia computacional.	Alta	Modelo	Ejecuta en tiempo razonable y sin alto consumo de recursos.
RNF-002	Soporte a futuro para más de dos cuerpos.	Media	Modelo	Escalable sin cambios estructurales mayores.
RNF-003	Respeto a leyes físicas en interacciones gravitacionales.	Alta	Modelo	Resultados coherentes con la física.
RNF-004	Estabilidad ante modificaciones dinámicas de parámetros.	Alta	Modelo	No colapsa ni muestra comportamientos erráticos.
RNF-005	Interfaz intuitiva y accesible para usuarios no técnicos.	Media	Usuario	Puede usarse sin conocimientos avanzados.
RNF-006	Ejecución en hardware de gama media.	Media	Modelo	Funciona en equipos con especificaciones mínimas.
RNF-007*	Integración con entornos como Unreal Engine.	Baja	Modelo	Integración funcional con motores externos.
RNF-008	Arquitectura modular para facilitar mantenimiento.	Media	Desarrollador	Componentes separados y modificables.
RNF-009	Documentación clara para usuarios y desarrolladores.	Media	Desarrollador	Completa y comprensible para todos los perfiles.

\* El requerimiento no funcional No. 07 (RNF-007), solo hace mención a la idea conceptual del requerimiento, no se plantea su elaboración.



# Formulación del Problema de Optimización

## Función Objetivo

Minimizar el valor absoluto del exponente de Lyapunov ( $\lambda$ ) en función de las masas  $m_1$  y  $m_2$ :

$$\text{Minimizar} \quad |\lambda(m_1, m_2)|$$

## Variables de Decisión

- $m_1$ : Masa del cuerpo 1
- $m_2$ : Masa del cuerpo 2

# Formulación del Problema de Optimización

## Restricciones

- El exponente de Lyapunov debe ser menor a  $\alpha$ :

$$\lambda(m_1, m_2) < \alpha$$

- La masa del cuerpo 2 debe ser menor a  $\mu$  veces la masa del cuerpo 1:

$$m_2 < \mu m_1$$

- Las masas deben ser positivas:

$$m_1 > 0$$

$$m_2 > 0$$

## 1 Introducción

## 2 Estado del Arte

## 3 Marco Teórico

## 4 Planeación

## 5 Análisis

## 6 Diseño

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Matriz de procesos

Tabla 7: Matriz de procesos I

Nombre del proceso	Objetivo	Salidas
Captura Parámetros	Recopilar, validar y almacenar parámetros de configuración del usuario.	Estructura ConfigurationData validada, estado de UI actualizado.
Mostrar Resultados	Presentar solución óptima y visualización final al usuario.	Actualización visual de la UI con resultados finales.
Evaluar Fitness	Calcular fitness penalizado ( $F_p$ ) combinando LE y violaciones.	Valor numérico de $F_p(x)$ .
Crear Nueva Simulación	Instanciar un nuevo entorno de simulación vacío en REBOUND.	Referencia a nuevo objeto Simulation.

## Matriz de procesos II

Tabla 8: Matriz de procesos

Nombre del proceso	Objetivo	Salidas
Agregar Cuerpos	Añadir una partícula con propiedades físicas a la simulación.	Instancia sim modificada con nueva partícula.
Iniciar/Ejecutar simulación	Si- Ejecutar la integración numérica paso a paso hasta $T_{\text{máx.}}$ .	Estructura Simulation Result con trayectoria completa.
Recolectar Datos	Extraer estado actual del sistema en instantes de visualización.	Estructura VisualizationState con instantánea del sistema.
Generar Gráficos	Dibujar o actualizar la representación visual en la pantalla.	Representación gráfica actualizada en la UI.

# Diccionario de Datos: Cuerpo celeste

Tabla 9: Cuerpo celeste.

Nombre del atributo	Descripción	Tipo	Rango	Ejemplo
masa	Masa del cuerpo celeste...	float	> 0 (positivos)...	1.0
a	Semieje mayor de la órbita...	float	> 0 (positivos)	1.0
e	Excentricidad orbital...	float	[0, 1)	0.1
inc_deg	Inclinación orbital...	float	[0°, 180°]	30.0
perturba	Indica si se aplica...	bool	True o False	

# Diccionario de Datos: Simulación

Tabla 10: Simulación.

Nombre del atributo	Descripción	Tipo	Rango	Ejemplo
t_max	Tiempo total de simulación...	float	> 0 (positivos)	100.0
N_steps	Número de pasos a almacenar...	entero	> 0 (positivos)	1000
sim.units	Unidades de la simulación...	texto	AU, yr, Msun	AU, yr, Msun
sim.integrator	Indica el integrador numérico...	texto	ias15, whfast, BS, mercurius	ias15
x, y, z	Guarda las posiciones...	array(float) 0 (positivos)	[5.0, 20.0]	230.0,

# Diccionario de Datos: Métricas

Tabla 11: Métricas.

Nombre del atributo	Descripción	Tipo	Rango	Ejemplo
times	Array que guarda los tiempos...	array(float)	> 0 (positivos)	[0.0, 200.0]
energy	Energía total del sistema...	float	Valor real	-0.5
a_arr, a_pert	Array que guarda el semi-eje...	array(float)	> 0 (positivos)	[1.0, 1.5, 2.0]
e_arr, e_pert	Array que guarda la excentricidad...	array(float)	[0, 1)	[0.1, 0.2, 0.3]
Exponente de Lyapunov ( $\lambda$ )	Indica la tasa de crecimiento...	float	Valor real	0.01

## 1 Introducción

## 2 Estado del Arte

## 3 Marco Teórico

## 4 Planeación

## 5 Análisis

## 6 Diseño

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Diagrama de Arquitectura

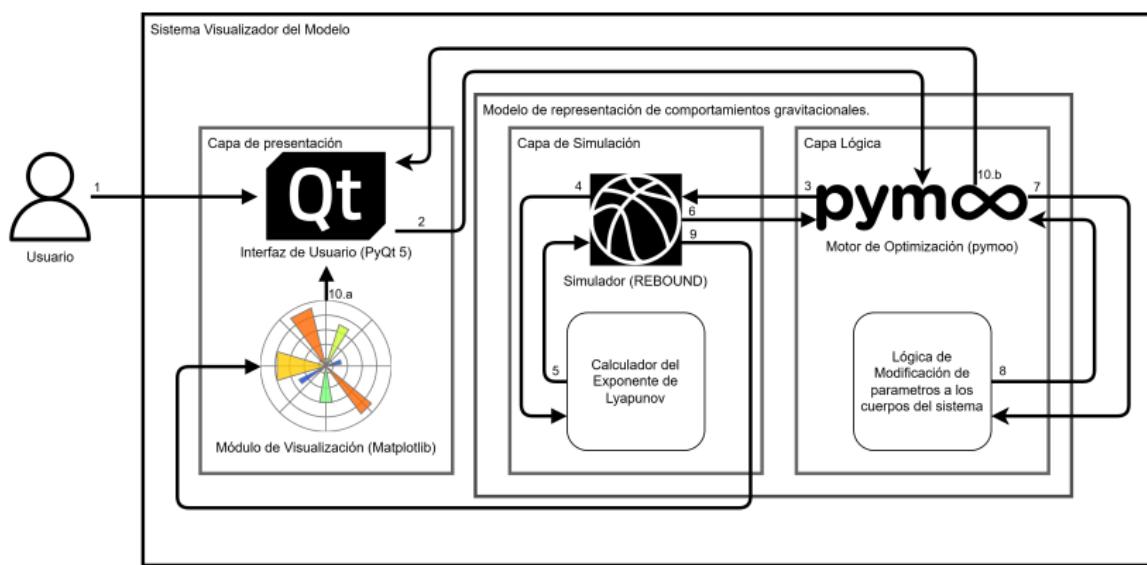


Figura 15: Diagrama de Arquitectura propuesto para el modelo. Autoría Propia

## 1 Introducción

## 2 Estado del Arte

## 3 Marco Teórico

## 4 Planeación

## 5 Análisis

## 6 Diseño

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Conclusión

**Durante la primera fase del Trabajo Terminal, se logró:**

- Establecer el planteamiento y la metodología que guiarán el desarrollo de la solución propuesta.
- Identificar diversos métodos y técnicas para abordar eficazmente la problemática planteada.
- Consultar a expertos para validar el enfoque y alinearlos con los objetivos del proyecto.



**Figura 16:** Una vista del cielo mirando hacia arriba por la noche *Autoría de: [13]*

# Trabajo a futuro



**Figura 17:** Transbordador espacial Challenger se lanza desde el Centro Espacial Kennedy. Autoría de: [14]

Para la siguiente fase del proyecto terminal, se realizarán las siguientes tareas:

- Desarrollo de pseudocódigos
- Implementación de módulos.
- Pruebas y refinamiento.
- Consulta con expertos y miembros del comité sinodal sobre los avances.
- Elaboración de un manual de usuario.

## 1 Introducción

## 2 Estado del Arte

## 3 Marco Teórico

## 4 Planeación

## 5 Análisis

## 6 Diseño

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

## Referencias I

- [1] Smarty Prints, "Constellation Star Poster: Northern Hemisphere, Stellar Life Cycle, Night Sky Stargazing", Póster educativo que muestra que ilustra el ciclo de vida de una estrella., Smarty Prints. (14 de mayo de 2025), dirección: <https://smartypointsshop.com/products/constellation-star-poster-northern-hemisphere-stellar-life-cycle-night-sky-stargazing-print-16x20-18x24-24x36> (visitado 14-05-2025).

## Referencias II

- [2] NASA's Scientific Visualization Studio, "Star Collision", Visualización científica producida por Scott Wiessinger. Visualización por Brian Monroe. Redacción por Francis Reddy. Consultoría científica por Julie McEnery, Brad Cenko y Eleonora Troja., NASA Goddard Space Flight Center. (2 de jul. de 2018), dirección: <https://svs.gsfc.nasa.gov/12949/> (visitado 14-05-2025).
- [3] J. Kim y col., "Prediction of engine NOx for virtual sensor using deep neural network and genetic algorithm", *Oil & Gas Science and Technology - Revue de l' IFP*, vol. 76, pág. 72, nov. de 2021. DOI: 10.2516/ogst/2021054.

## Referencias III

- [4] Python Software Foundation, “Python Software Foundation Landing Page”, Python Software Foundation. (14 de mayo de 2025), dirección: <https://www.python.org/psf-landing/> (visitado 14-05-2025).
- [5] H. Rein y col., “REBOUND: an open-source multi-purpose N-body code for collisional dynamics”, *Astronomy & Astrophysics*, vol. 537, pág. 10, 2012, Published online 20 January 2012. DOI: 10.1051/0004-6361/201118085. dirección: <https://doi.org/10.1051/0004-6361/201118085>.
- [6] J. Blank y col., “pymoo: Multi-Objective Optimization in Python”, *IEEE Access*, vol. 8, págs. 89 497-89 509, 2020, Introduces pymoo, a Python framework for multi-objective optimization. DOI: 10.1109/ACCESS.2020.2990567.

## Referencias IV

- [7] J. D. Hunter, "Matplotlib: A 2D graphics environment", *Computing in Science & Engineering*, vol. 9, n.º 3, págs. 90-95, 2007. DOI: 10.1109/MCSE.2007.55.
- [8] Qt Group, *Qt Group Frequently Asked Questions*, Qt Group page, last edited May 1, 2025, 2025. dirección: <https://www.qt.io/faq>.
- [9] H. Rein, "Hyperbolic Orbits", Notebook de Jupyter que demuestra la simulación de órbitas hiperbólicas utilizando el código N-body REBOUND., GitHub. (14 de mayo de 2025), dirección: [https://github.com/hannorein/rebound/blob/main/ipython\\_examples/HyperbolicOrbits.ipynb](https://github.com/hannorein/rebound/blob/main/ipython_examples/HyperbolicOrbits.ipynb) (visitado 14-05-2025).

## Referencias V

- [10] K. Gonçalves-e Silva y col., "Less is more: Simplified Nelder-Mead method for large unconstrained optimization", *Yugoslav Journal of Operations Research*, vol. 28, págs. 14-14, jun. de 2018. DOI: 10.2298/YJOR180120014G.
- [11] P. Bourke, "Random Attractors – Found using Lyapunov Exponents", Página que explora atractores caóticos generados mediante exponentes de Lyapunov en sistemas dinámicos bidimensionales. (oct. de 2001), dirección: <https://paulbourke.net/fractals/lyapunov/> (visitado 14-05-2025).
- [12] S. Boyd y col., *Convex Optimization – Slides*, Presentation slides, Presentation slides accompanying the book "Convex Optimization". Revised by Stephen Boyd, Lieven Vandenberghe, and Parth Nobel. Version date on title page of PDF: 2023., 2023.

## Referencias VI

- [13] J. Dyer, "Una vista del cielo mirando hacia arriba por la noche", Fotografía de dominio público disponible en Unsplash. (28 de dic. de 2021), dirección: <https://unsplash.com/es/fotos/una-vista-del-cielo-mirando-hacia-arriba-por-la-noche-nkGeS7jEBNg> (visitado 19-05-2025).
- [14] NASA, "El transbordador espacial Challenger se lanza desde el Centro Espacial Kennedy", Fotografía de dominio público proporcionada por NASA a través de Unsplash. (14 de mar. de 2020), dirección: <https://unsplash.com/es/fotos/el-transbordador-espacial-challenger-se-lanza-desde-el-centro-espacial-kennedy-dCgbRAQmTQA> (visitado 19-05-2025).

## Referencias VII

- [15] S. I. Newton, *Philosophiae Naturalis Principia Mathematica*, Revised and edited by Florian Cajori, trad. por A. Motte. The Royal Society, 1687, University of California Press, 1934.
- [16] S. Orlov, “C++ Playground for Numerical Integration Method Developers”, en *Supercomputing*, V. Voevodin y col., eds., Cham: Springer International Publishing, 2017, págs. 418-429. DOI: 10.1007/978-3-319-71255-0\_34.
- [17] J. Barnes y col., “A hierarchical  $O(N \log N)$  force-calculation algorithm”, *Nature*, vol. 324, n.º 6096, págs. 446-449, 1986, ISSN: 1476-4687. DOI: 10.1038/324446a0. dirección: <https://doi.org/10.1038/324446a0>.

## Referencias VIII

- [18] A. Ayoub y col., "Cell Blood Image Segmentation Based on Genetic Algorithm", en mar. de 2020, págs. 564-573, ISBN: 978-3-030-44288-0. DOI: 10.1007/978-3-030-44289-7\_53.
- [19] Stack Overflow, "What does crossover index of 0.25 mean in Genetic algorithm for real encoding?" Discusión sobre el parámetro de índice de cruce en el operador Simulated Binary Crossover (SBX). (22 de mayo de 2019), dirección: <https://stackoverflow.com/questions/56263132/what-does-crossover-index-of-0-25-means-in-genetic-algorithm-for-real-encoding> (visitado 19-05-2025).

## Referencias IX

- [20] J. L. Carles-Bou y col., "Self-adaptive polynomial mutation in NSGA-II", *Soft Computing*, vol. 27, n.º 23, págs. 17711-17727, 2023, ISSN: 1433-7479. DOI: 10.1007/s00500-023-09049-0. dirección: <https://doi.org/10.1007/s00500-023-09049-0>.

*¡Gracias por su atención!*

¿Alguna pregunta o comentario?

# Anexos

## 9 Tablas Comparativas del Marco Teórico

## 10 Diagramas de procesos y actividades

## 11 Metodología del Algoritmos Genéticos

## Anexo A : Lenguaje de Programación

Tabla 12: Comparativa de Lenguaje de Programación

Característica	C++	Python
Velocidad de Ejecución	Máxima. Ideal para cálculos intensivos.	Menor.
Velocidad de Desarrollo	Lento y verboso.	Rápido, claro y flexible para iterar.
Bibliotecas Científicas	Potentes, pero más complejas de integrar.	Vastas y accesibles.
Visualización	Requiere herramientas externas.	Fácil con Matplotlib, Plotly o Bokeh.
Integración con REBOUND	Directa con linking C.	Interfaz oficial en Python lista para usar.
Algoritmos Bioinspirados	Óptimo si se implementan desde cero.	Fáciles de coordinar o conectar con código externo.
Gestión de Memoria	Manual. Mayor control.	Automática. Simplifica el desarrollo.
Curva de Aprendizaje	Alta..	Baja a moderada..
Prototipado / Experimentación	Lento.	Ágil.
Depuración	Complejo.	Sencillo.
Enfoque del Proyecto	Control total, pero puede complicar.	Permite centrarse en la solución y resultados.

# Anexo A : Comparativa de Bibliotecas N-Cuerpos

Tabla 13: Comparativa extensa de bibliotecas para simulación N-cuerpos.

Característica	REBOUND	PKDGRAV3	AMUSE	NBody (Python)	PyGaia
Lenguaje Principal / Interfaz	C (Python API)	C++	Python	Python	Python
Enfoque Principal	Colisional/no colisional, sistemas N-cuerpos	Cosmología a gran escala	Framework multipropósito	Simulaciones educativas	Ánálisis Gaia
Tipos de Problemas	Planetas, cúmulos, anillos	Galaxias, materia oscura	Multifísica astrofísica	Sistemas pequeños	Dinámica galáctica
Algoritmos de Gravedad	Barnes-Hut/Suma directa	TreePM	Hermite/Tree/SPH	Suma directa	Potenciales analíticos
Manejo de Colisiones	Sí (esferas duras)	No	Depende del backend	No	N/A
Integradores Numéricos	WHFast/ IAS15/ Leap-frog	Leapfrog KDK	Hermite/ Symplectic	RK/ Leapfrog	SciPy ODE
Hidrodinámica (SPH/Gas)	No	Sí	Sí	No	No
Paralelización	MPI/OpenMP	MPI	MPI frameworks	Multiprocessing	CPU básica
Flexibilidad/Modularidad	Módulos intercambiables	Enfoque cosmología	Interoperabilidad	Implementación-dependiente	Centrado en Gaia
Facilidad de Uso (Python)	Excelente docs/API	N/A (C++)	Compleja (múltiples backends)	Variable	Astronomer-friendly
Comunidad/Mantenimiento	Activo desarrollo	Cosmología activa	Colaborativo	Individual	Soporte Gaia
Idoneidad para el Proyecto	Óptimo: • Soporta 2 cuerpos • Python • Modular	Inadecuado: Escala/física diferente	Complejidad excesiva para necesidades simples	Muy básico para requisitos	Enfoque observacional no aplicable

## Anexo A : Algoritmos de Optimización

Tabla 14: Comparativa de Bibliotecas de Optimización

Biblioteca	Enfoque Principal	Clave / Fortaleza	Ideal Para (Contexto del Proyecto)	Complejidad / Flexibilidad
pymoo	Framework moderno y completo para optimización <b>multiobjetivo</b> (y mono). Amplia gama de algoritmos.		Problemas complejos, si se requieren múltiples objetivos o algoritmos robustos mono-objetivo.	Moderada / Alta
DEAP	Máxima <b>flexibilidad</b> para construir algoritmos evolutivos desde cero.		Experimentación profunda con la estructura interna de los algoritmos (ej. GA personalizado).	Alta / Muy Alta
Platypus	<b>Optimización multiobjetivo fácil de usar</b> con algoritmos estándar (NSGA-II, SPEA2).		Implementación rápida de MOEAs conocidos, buen punto de partida para multiobjetivo.	Baja-Moderada / Moderada
Inspyred	Framework versátil para varios algoritmos evolutivos y metaheurísticas.		Exploración de diferentes tipos de algoritmos evolutivos si las opciones más nuevas no son prioritarias.	Moderada / Alta
Nevergrad	<b>Optimización sin derivadas (caja negra)</b> ; ideal para funciones costosas/ruidosas.		Cuando la función objetivo (simulación + LE) es compleja y sin gradientes fáciles.	Moderada / Alta
PyGMO/PaGMO	<b>Alto rendimiento y paralelización</b> (backend C++) para problemas globales complejos.		Simulaciones muy costosas donde la paralelización es crítica para la eficiencia.	Moderada-Alta / Alta

## Anexo A : Bibliotecas de Visualización

Tabla 15: Comparativa de Bibliotecas de Visualización

Característica	Matplotlib	Plotly	Bokeh	Seaborn
Uso Principal/Fortaleza	Gráficos científicos.	Interactividad web.	Interactividad web.	Estadísticas.
Animación	Muy capaz. Ideal para simulaciones.	Buena para animaciones web.	Buena para datos en streaming.	Limitada.
Gráficos Estáticos	Excelente.	Buena.	Buena.	Regular.
Interactividad	Básica.	Alta.	Alta.	Básica.
Interfaz	Simple.	Web	Web.	N/A.
Calidad Publicación	Muy alta.	Alta.	Alta.	Alta.
Trayectorias 2D/3D	Directa.	Clara y efectiva.	Buena.	N/A.
Curva Aprendizaje (anim.)	Moderada.	Moderada.	Moderada.	Moderada.
Estética por Defecto	Funcional.	Moderna.	Flexible.	Mejorada estadísticas.
Comunidad/Docs	Muy extensa.	En crecimiento.	Buena.	Buena.
Adecuación Proyecto	Excelente: animación, análisis.	Interactividad web.	Prioridad web.	Ánálisis complementario.

## Anexo A : Comparativa de Bibliotecas GUI

Tabla 16: Comparativa de bibliotecas Python para Interfaces Gráficas.

Característica	PyQt	Tkinter	Streamlit	PySide	Dear PyGui
Toolkit subyacente	Qt (C++)	Tcl/Tk	React (Web)	Qt (C++)	GPU (ImGui-like)
Estilo	Profesional, personalizable	Anticuado, editable	Moderno, limpio	Igual a PyQt	Herramientas/juegos
Complejidad de desarrollo	Media-Alta	Baja-Media	Muy baja	Igual a PyQt	Media
Curva de aprendizaje	Moderada	Baja	Muy baja	Igual a PyQt	Requiere nuevo concepto
Rendimiento	Muy bueno, optimizado	Bueno en GUIs simples	Bueno, depende del navegador	Muy bueno, igual a PyQt	Excelente, acelerado GPU
Widgets disponibles	Extensa colección madura	Básica, útil	Limitada, centrada en datos	Igual a PyQt	Buena para herramientas
Layouts	Extensos, Qt Designer	Pack/grid/place	Automáticos, poco control	Igual a PyQt	Control total programático
Integración gráfica	Excelente con Matplotlib, etc.	Buena con Matplotlib	Excelente para Plotly y Altair	Igual a PyQt	Posible, requiere ajustes
Multihilo	Señales/slots robustos	Limitado, no thread-safe nativo	Abstracción interna	Igual a PyQt	Usuario maneja concurrencia
Multiplataforma	Win/macOS/Linux	Win/macOS/Linux	Navegador (web)	Win/macOS/Linux	Win/macOS/Linux
Licencia	GPL/comercial (LGPL en PyQt5)	Lib. estándar (libre)	Apache 2.0 (libre)	LGPL (comercial viable)	MIT (libre)
Comunidad y documentación	Amplia y activa	Amplia	Activa	Activa y sólida	Creciente y entusiasta
Idoneidad para el proyecto	Excelente: robusto, flexible, GUI + Matplotlib embebido	Adeuada para GUI simple	Menos ideal: mejor para dashboards web	Excelente alternativa LGPL	Paradigma distinto, menos directo

## Anexo A : Enfoques principales en problemas de N-Cuerpos

**Tabla 17:** Comparativa de enfoques principales a la hora de resolver problemas de  $n$  cuerpos

Enfoque	Descripción	Complejidad	Referencia
Solución Analítica (n=2)	Método que resuelve el problema de Kepler para dos cuerpos. Proporciona soluciones exactas para sistemas de dos objetos.	$O(1)$	[15]
Integración Numérica	Utiliza métodos como Runge-Kutta para simular movimientos. Permite aproximaciones numéricas para sistemas complejos.	$O(n^2)$	[16]
Algoritmo Barnes-Hut	Método de agrupamiento jerárquico para reducir complejidad computacional. Mejora la eficiencia en simulaciones de múltiples cuerpos.	$O(n \log n)$	[17]

## Anexo A : Selección de Integradores Simplécticos

Tabla 18: Evaluación de Integradores Simplécticos Relevantes

Integrador	Eficiencia en Sistemas Keplerianos Dominados	Manejo de Encuentros Cercanos	Complejidad del Método Base
Leapfrog/Verlet	Moderada	X (Requiere regularización)	Baja
Wisdom-Holman (WHFast)	✓ ✓ (Muy Alta)	? (Requiere hibridación)	Moderada
Orden Superior (Yoshida)	Variable	X (Puede ser inestable)	Alta
Híbridos (con WHFast)	Alta (base WHFast)	✓ (Propósito principal)	Alta (Combinación)

Leyenda: ✓ ✓ Muy Bueno/Alto, ✓ Bueno/Moderado, ? Depende/Requiere Adicional, X Bajo/No Directo

9 Tablas Comparativas del Marco Teórico

10 Diagramas de procesos y actividades

11 Metodología del Algoritmos Genéticos

## Anexo B : Diagrama de Arquitectura

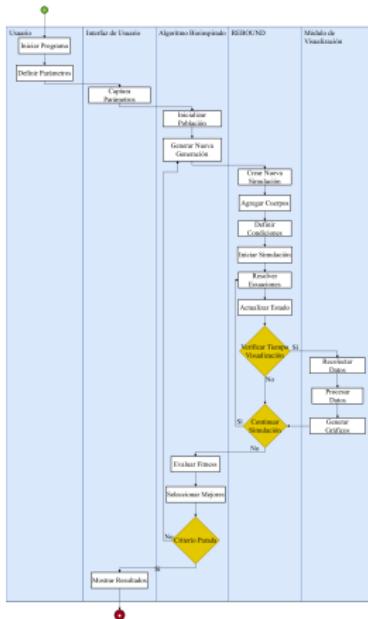


Figura 18: Diagrama de Actividades propuesto para el modelo. Autoría Propia

⑨ Tablas Comparativas del Marco Teórico

⑩ Diagramas de procesos y actividades

⑪ Metodología del Algoritmos Genéticos

## Anexo C : Selección de Padres

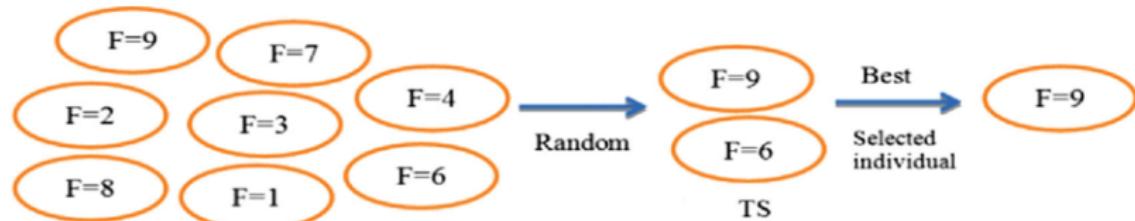


Figura 19: Ilustración que describe el proceso de Selección de padres. Adaptado de: [18]

## Anexo C : Cruzamiento

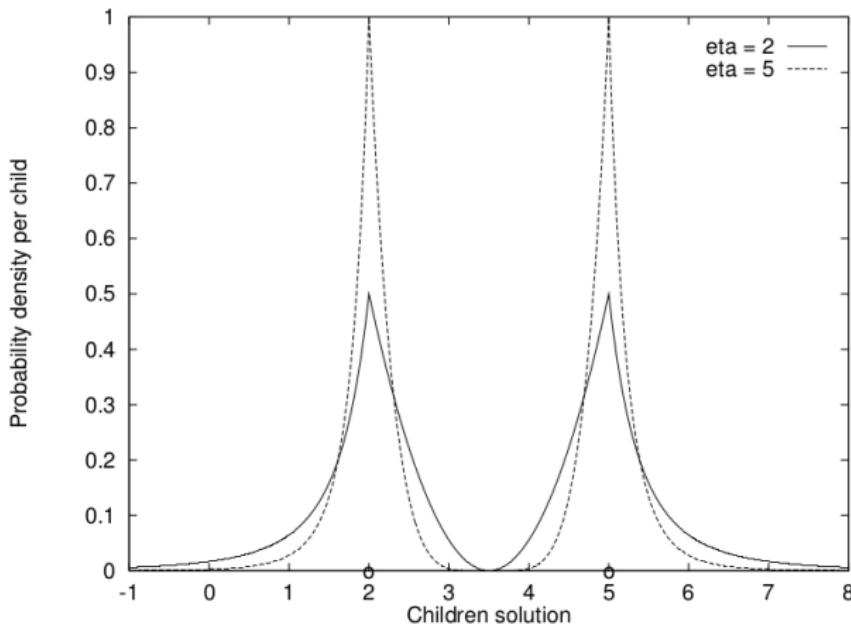
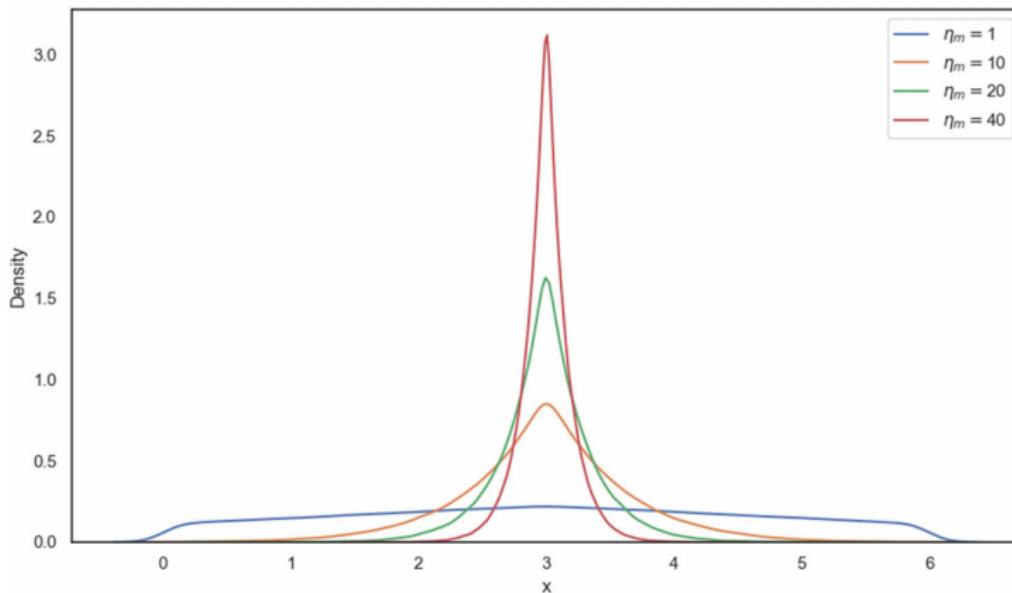


Figura 20: Ilustración que describe el proceso de Cruzamiento. Adaptado de: [19]

## Anexo C : Mutación

Figura 21: Ilustración que describe el proceso de Mutación. *Adaptado de: [20]*