



# 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 Otero José Ángel

**Directores:**

Dr. Cesar Hernández Vasquez  
Dr. Mauricio Olguín Carbajal

24 de mayo de 2025

Ciudad de México

## 1 Introducción

## 2 Estado del Arte

## 3 Marco teórico y tecnológico

## 4 Planeación

## 5 Análisis

## 6 Familiarización con Tecnologías

## 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.

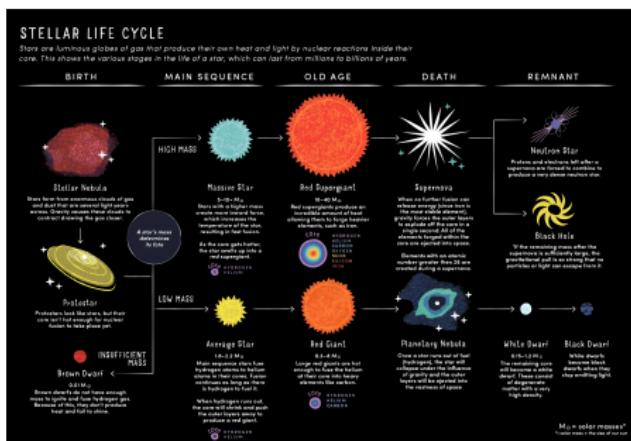


Figura 1: Diagrama del ciclo de evolución de una estrella. Propiedad de: smartyprints\_constellation\_poster\_2025

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

# Antecedentes



Figura 2: Retrato de Sir Isaac Newton pintado por Sir Godfrey Kneller en 1689 Extraído de: [cambridge\\_newton\\_kneller\\_2016](#)

$$F_{ij} = \frac{Gm_i m_j (\mathbf{q}_j - \mathbf{q}_i)}{\|\mathbf{q}_j - \mathbf{q}_i\|^3},$$

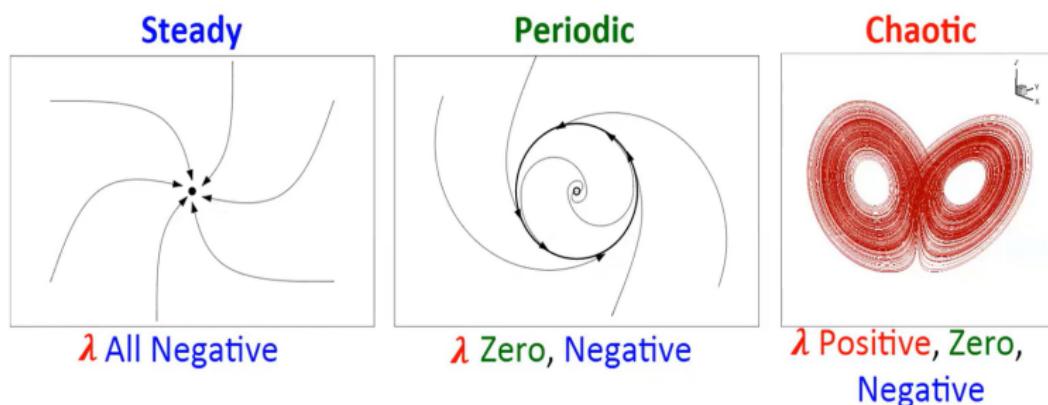
$$m_i \frac{d^2 \mathbf{q}_i}{dt^2} = \sum_{j=1, j \neq i}^N \frac{Gm_i m_j (\mathbf{q}_j - \mathbf{q}_i)}{\|\mathbf{q}_j - \mathbf{q}_i\|^3}$$



Figura 3: Colisión de dos estrellas de neutrones (Simulación). Extraído de: [nasa\\_star\\_collision\\_2018](#)

## 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 4:** Ilustración de tres comportamientos de sistemas dinámicos en relación con sus exponentes de Lyapunov. Extraído de: wang\_numerical\_methods\_2014

# 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**.

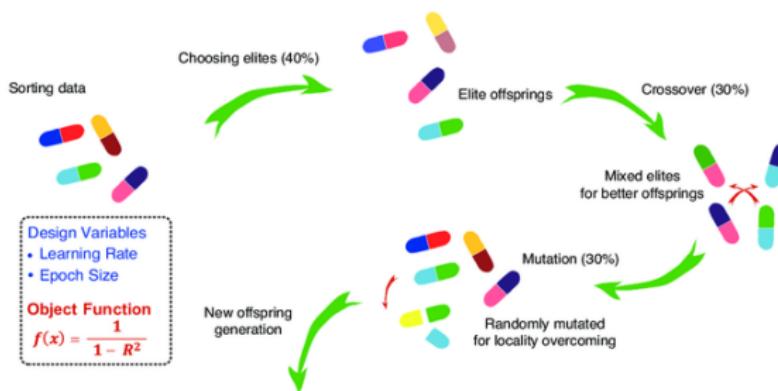
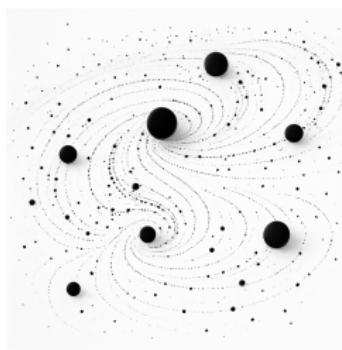


Figura 5: Ejemplo de Aplicación de un Algoritmo Genético (AG) Adaptado de: Kim2021

# Objetivos

## Objetivo General

Desarrollar un modelo de optimización 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 6:** 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.
- **Módulo de Simulación dinámica:** Desarrollar e implementar el modelo computacional de para la simulación dinámica de un sistema de dos cuerpos bajo interacción gravitatoria.

# Objetivos II

## Objetivo Específicos

- **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.
- **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

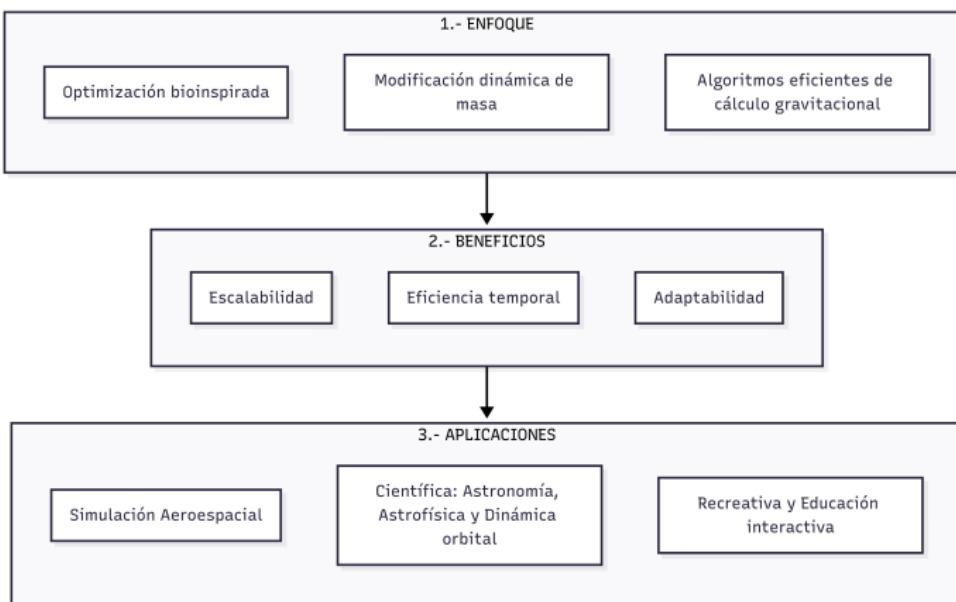


Figura 7: Diagrama representativo de los niveles de Justificación. Autoría Propia

## 1 Introducción

## 2 Estado del Arte

## 3 Marco teórico y tecnológico

## 4 Planeación

## 5 Análisis

## 6 Familiarización con Tecnologías

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Comparativa de Métodos N-cuerpos

Tabla 1: 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. Simplicético	✗	✓	✗	✗
Solver Híbrido TPM	✓	✗	✗	✗
Algoritmo TPM	✓	✗	✗	✗
REBOUND	✓	✓	✗	✗
Est. Planetaria	✓	✗	✗	✗
Solución Propuesta**	✓	✓	✓	✓

\*Dinámico: Capacidad de modificar parámetros clave durante la ejecución

\*\* Basada en REBOUND: REBOUND + masas dinámicas

## 1 Introducción

## 2 Estado del Arte

## 3 Marco teórico y tecnológico

Métodos & Técnicas a utilizar.

Tecnologías: Lenguaje & Librerías.

## 4 Planeación

## 5 Análisis

## 6 Familiarización con Tecnologías

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

## 1 Introducción

## 2 Estado del Arte

## 3 Marco teórico y tecnológico

Métodos & Técnicas a utilizar.

Tecnologías: Lenguaje & Librerías.

## 4 Planeación

## 5 Análisis

## 6 Familiarización con Tecnologías

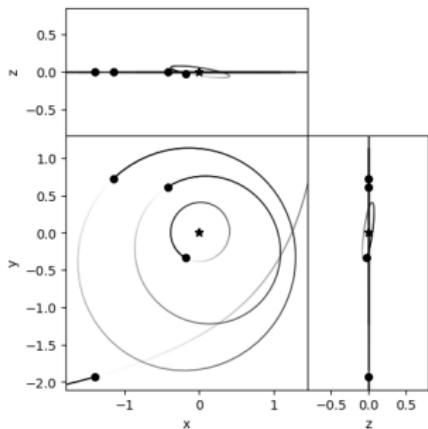
## 7 Conclusiones y Trabajo a Futuro

## 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:  $O(N \cdot N_{\text{SWEPL}})$

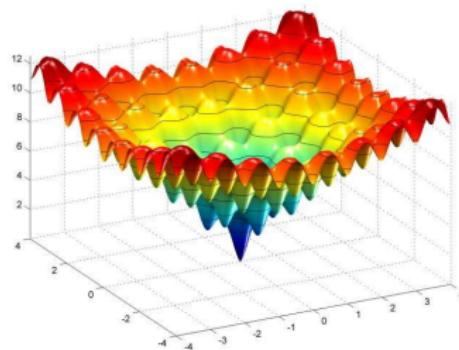


**Figura 8:** Simulación de órbitas usando REBOUND. *Adaptado de: rebound\_hyperbolic\_orbits\_2025*

# Algoritmo de Exploración

## Algoritmo Genético (AG) con pymoo

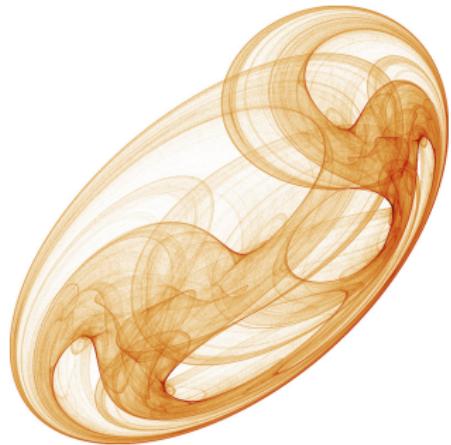
- **Componentes principales:**
  - **Muestreo**
  - **Selección de Padres**
  - **Cruce**
  - **Mutación**
  - **Manejo de Restricciones**



**Figura 9:** Función tipo benchmark,  
para la evaluación de AGs *Adaptado  
de: silva2018*

# Indicador de Estabilidad

## Exponente de Lyapunov

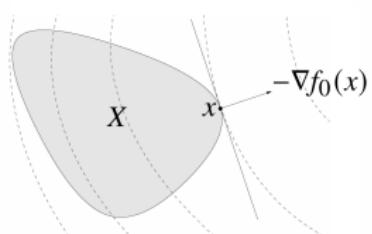


**Figura 10:** Atractor caótico generado mediante exponentes de Lyapunov. *Adaptado de: bourke\_lyapunov\_attractors\_2001*

- 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 11:** 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: BoydVandenberghe Slides 2023

- **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 y tecnológico

Métodos & Técnicas a utilizar.

Tecnologías: Lenguaje & Librerías.

## 4 Planeación

## 5 Análisis

## 6 Familiarización con Tecnologías

## 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 12: Logo de Python  
*Adaptado de: PythonSoftwareFoundation*

# Diagrama de Arquitectura

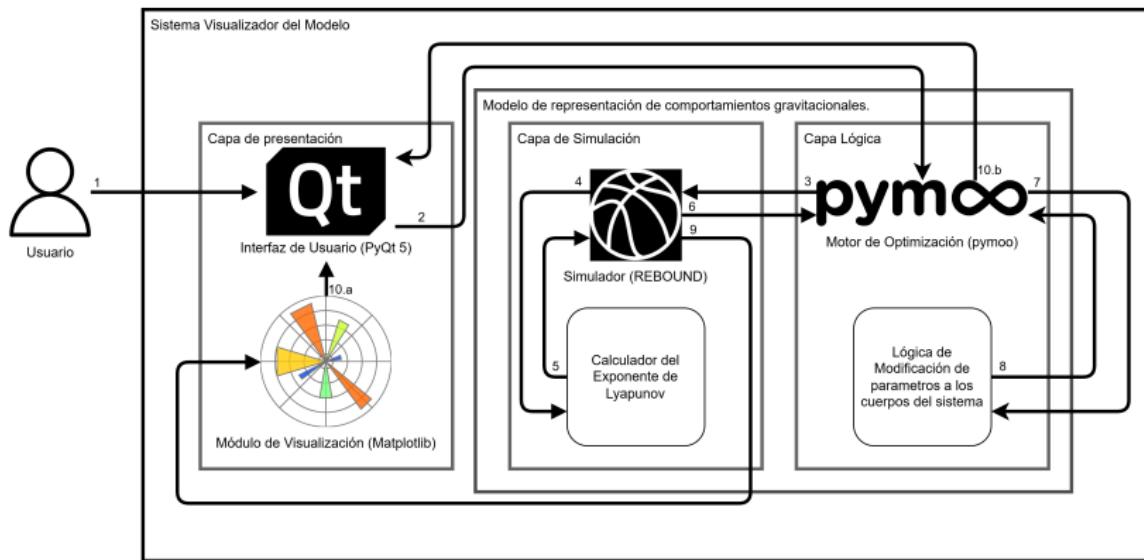


Figura 13: Diagrama de Arquitectura propuesto para el modelo. Autoría Propia

## 1 Introducción

## 2 Estado del Arte

## 3 Marco teórico y tecnológico

## 4 Planeación

## 5 Análisis

## 6 Familiarización con Tecnologías

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Juicio de Expertos

## M.C.José Alberto Torres León

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

## Dr. Daniel Molina Pérez

- WH-fast idóneo para determinar trayectorias.
- El uso de algoritmos bioinspirados está justificado.
- Aplicaciones aeroespaciales y astrofísicas.

# Requerimientos Funcionales

Tabla 2: 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 3: 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\*:

$$m_1 > 0$$

$$m_2 > 0$$

\* Llamamos masas a las variables que busca el AG; como explora todo  $\mathbb{R}$ , se debe restringir su dominio para limitar la búsqueda a valores factibles para el sistema.

## 1 Introducción

## 2 Estado del Arte

## 3 Marco teórico y tecnológico

## 4 Planeación

## 5 Análisis

## 6 Familiarización con Tecnologías

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Diagrama de Actividades

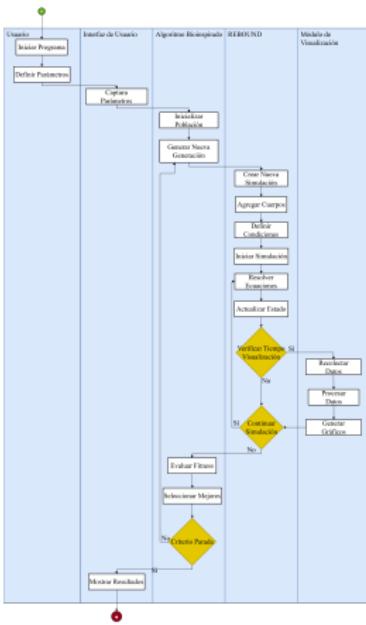


Figura 14: Diagrama de Actividades propuesto para el modelo. Autoría Propia

## 1 Introducción

## 2 Estado del Arte

## 3 Marco teórico y tecnológico

## 4 Planeación

## 5 Análisis

## 6 Familiarización con Tecnologías

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Pasos Iniciales con REBOUND

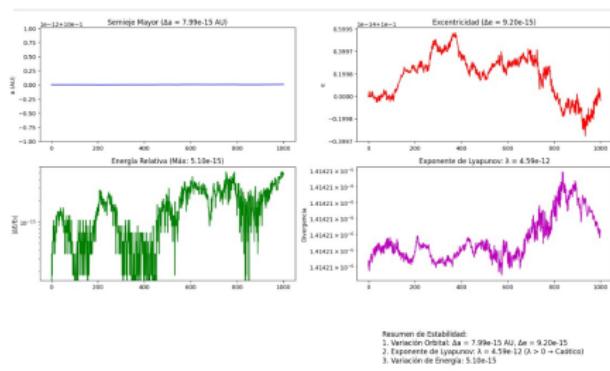


Figura 15: Gráficas de estabilidad orbital.

Tiempo: 1000.0 años

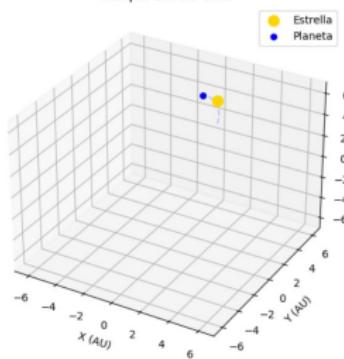


Figura 16: Representación 3D de una órbita.

## 1 Introducción

## 2 Estado del Arte

## 3 Marco teórico y tecnológico

## 4 Planeación

## 5 Análisis

## 6 Familiarización con Tecnologías

## 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 17: Una vista del cielo mirando hacia arriba por la noche. Autoría de: dyer\_cielo\_nocturno\_2021

# Trabajo a futuro



**Figura 18:** Transbordador espacial Challenger se lanza desde el Centro Espacial Kennedy. Autoría de: nasa\_challenger\_unsplash

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 y tecnológico

## 4 Planeación

## 5 Análisis

## 6 Familiarización con Tecnologías

## 7 Conclusiones y Trabajo a Futuro

## 8 Referencias

# Referencias |

*¡Gracias por su atención!*

¿Alguna pregunta o comentario?

# Anexos

## ⑨ Tablas Comparativas del Marco Teórico

## ⑩ Matriz de Procesos y Diccionario de datos

## ⑪ Metodología del Algoritmos Genéticos

## ⑫ Cronograma

# Anexo A : Lenguaje de Programación

Tabla 4: Comparativa de Lenguaje de Programación

Característica	C++	Python
Velocidad de Ejecución	✓	✗
Velocidad de Desarrollo	✗	✓
Disponibilidad de Bibliotecas Científicas	✓	✓
Facilidad de Uso de Bibliotecas	✗	✓
Visualización	✗	✓
Integración con REBOUND	✓	✓
Algoritmos Bioinspirados	✓	✓
Control de Gestión de Memoria	✓	✗
Simplicidad de Gestión de Memoria	✗	✓
Curva de Aprendizaje	✗	✓
Prototipado Rápido	✗	✓
Facilidad de Depuración	✗	✓
Enfoque del Proyecto	✗	✓



# Anexo A : Comparativa de Bibliotecas N-Cuerpos

**Tabla 5: Comparativa de bibliotecas para simulación N-cuerpos.**

Característica	REBOUND	PKDGRAV3	AMUSE	NBody (Python)	PyGaia
Lenguaje Principal	C (Python API)	C++	Python	Python	Python
Manejo de Colisiones	✓	✗	✓	✗	N/A
Enfoque en Cosmología	✗	✓	✗	✗	✗
Hidrodinámica (SPH/Gas)	✗	✓	✓	✗	✗
Paralelización Avanzada*	✓	✓	✓	Parcial	✗
Flexibilidad/Modularidad **	✓	✗	✓	Media	✗
Facilidad de Uso (Python)	✓	✗	✗	✓	✓
Comunidad Activa	✓	✓	✓	Variable	✓
Soporta 2 Cuerpos	✓	✗	Possible	✓	✗
Adecuación al Proyecto	✓	✗	✗	✗	✗

\*Paralelización Avanzada: Soporte para MPI/OpenMP.

\*\*Flexibilidad/Modularidad: Capacidad de personalizar o extender fácilmente.

\*\*\*Soporta 2 Cuerpos: Adecuado para simulaciones simples de pocos cuerpos.

# Anexo A : Algoritmos de Optimización

Tabla 6: Comparativa de Bibliotecas de Optimización

Biblioteca	Enfoque Clave	Ideal Para	Complejidad	Adecuación al Proyecto
pymoo	Multiobjetivo	Problemas complejos	Moderada	✓
DEAP	Flexibilidad máxima	Experimentación profunda	Alta	✗
Platypus	Multiobjetivo fácil	Implementación rápida	Baja-Moderada	~
Inspyred	Versátil	Exploración de algoritmos	Moderada	✗
Nevergrad	Sin derivadas	Funciones costosas/ruidosas	Moderada	~
PyGMO/PaGMO	Alto rendimiento	Simulaciones costosas	Moderada-Alta	~

## Anexo A : Bibliotecas de Visualización

Tabla 7: Comparativa de Bibliotecas de Visualización

Biblioteca	Uso Principal	Animación	Gráficos Estáticos	Interactividad	Trayectorias 2D/3D	Comunidad/Docs	Adecuación al Proyecto
Matplotlib	Científicos	✓	✓	~	✓	✓	✓
Plotly	Web Interactivo	~	~	✓	✓	✗	~
Bokeh	Web Interactivo	~	~	✓	~	~	~
Seaborn	Estadísticas	✗	✗	~	✗	~	✗

## Anexo A : Comparativa de Bibliotecas GUI

Tabla 8: Comparativa de bibliotecas Python para Interfaces Gráficas.

Biblioteca	Estilo	Complejidad	Rendimiento	Integración Gráfica	Multiplataforma	Adecuación al Proyecto
PyQt	Profesional	~	✓	✓	✓	✓
Tkinter	Anticuado	✓	~	✓	✓	~
Streamlit	Moderno	✓	~	✓	~	✗
PySide	Profesional	~	✓	✓	✓	✓
Dear PyGui	Herramientas	~	✓	~	✓	✗

\*Símbolos: ✓ (Bueno), ~ (Regular), ✗ (No adecuado).

\*\*Adecuación al Proyecto: ✓ (Excelente), ~ (Adecuada), ✗ (No ideal).

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

**Tabla 9:** 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.	$O(1)$	<b>newton1687</b>
Integración Numérica	Utiliza métodos como Runge-Kutta para simular movimientos.  Permite aproximaciones numéricas para sistemas complejos.	$O(n^2)$	<b>Orlov2017</b>
Algoritmo de Barnes-Hut	Método de agrupamiento jerárquico para reducir complejidad computacional.  Mejora la eficiencia en simulaciones de múltiples cuerpos.	$O(n \log n)$	<b>Barnes1986</b>

# Anexo A : Selección de Integradores Simplicéticos

Tabla 10: Evaluación de Integradores Simplicéticos 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 Matriz de Procesos y Diccionario de datos

11 Metodología del Algoritmos Genéticos

12 Cronograma

## Anexo B : Matríg de procesos

Tabla 11: Matríg de procesos I

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



## Anexo B : Matríg de procesos II

Tabla 12: Matríg 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 Visualizat ionState con instantánea del sistema.
Generar Gráficos	Dibujar o actualizar la representación visual en la pantalla.	Representación gráfica actualizada en la UI.

## Anexo B : Diccionario de Datos-Cuerpo celeste

Tabla 13: 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	

## Anexo B : Diccionario de Datos-Simulación

Tabla 14: 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,

## Anexo B : Diccionario de Datos-Métricas

Tabla 15: 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

9 Tablas Comparativas del Marco Teórico

10 Matriz de Procesos y Diccionario de datos

11 Metodología del Algoritmos Genéticos

12 Cronograma

## Anexo C : Selección de Padres

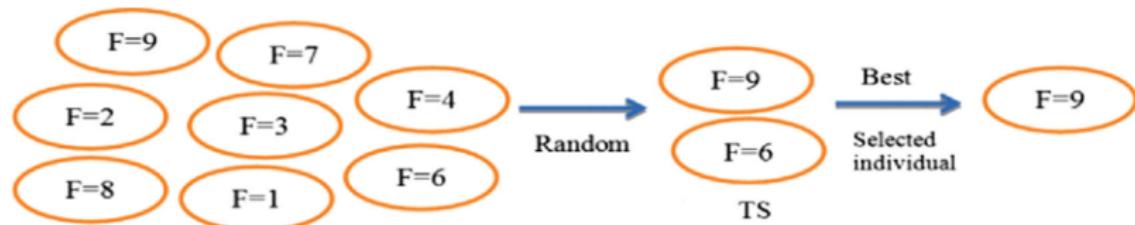
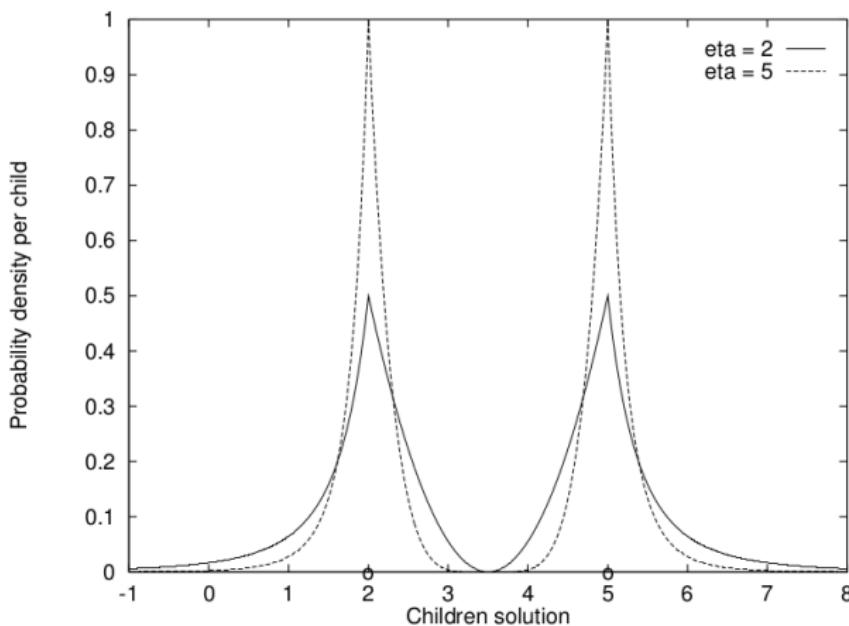


Figura 19: Ilustración que describe el proceso de Selección de padres. *Adaptado de: ayoub2020*

## Anexo C : Cruzamiento



**Figura 20:** Ilustración que describe el proceso de Cruzamiento. *Adaptado de: stackoverflow\_crossover\_index\_2019*

## Anexo C : Mutación

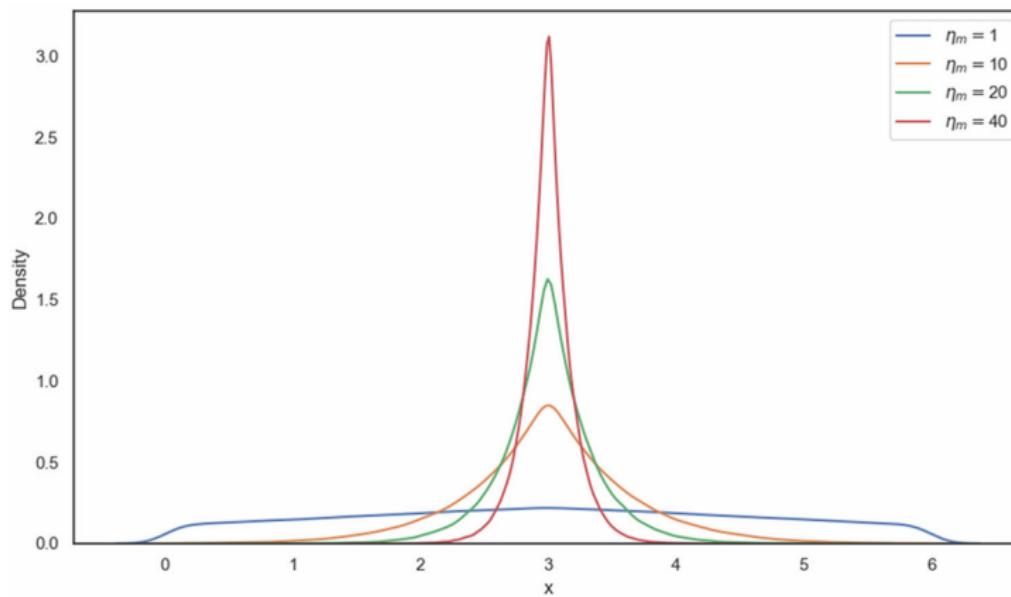


Figura 21: Ilustración que describe el proceso de Mutación. Adaptado de: CarlesBou2023

9 Tablas Comparativas del Marco Teórico

10 Matriz de Procesos y Diccionario de datos

11 Metodología del Algoritmos Genéticos

12 Cronograma

# Anexo D : Cronograma del Proyecto

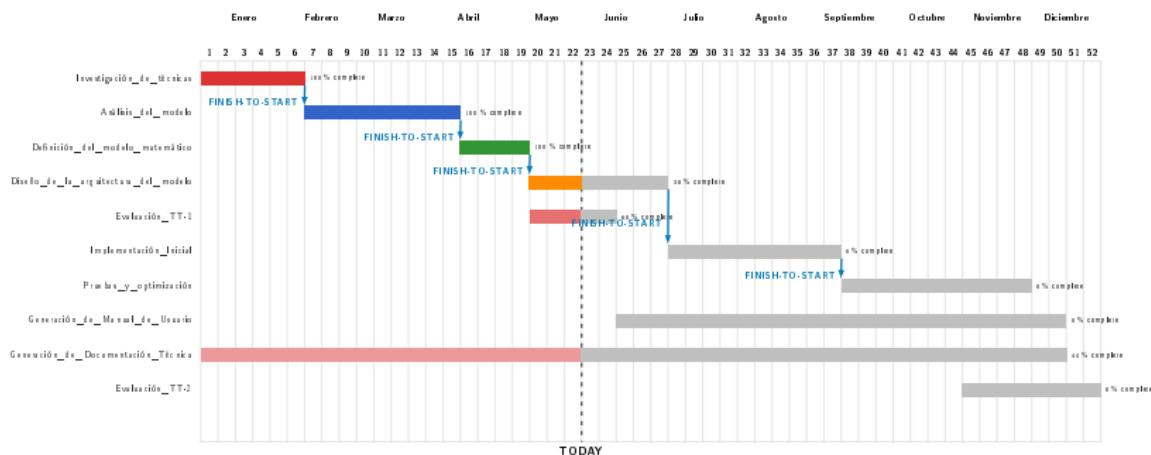


Figura 22: Cronograma de tiempo *Milestone*