Universidad
Nacional de
Ingeniería

Área del Conocimiento de Agricultura

Desarrollo de un Software de Código Abierto Para la Simulación de Reactores Batch Ideales Homogéneos Isotérmicos y no Isotérmicos

Trabajo Monográfico para optar al título de
Ingeniero Químico

Elaborado por:

Br. Carlos Adrián Espinosa Luna
2021-0663U

Br. Blanca Gisell Rivas Medal
2021-0740U

Tutor:

MSc. Juan Manuel Alonso Santos

19 de junio del 2026
Managua, Nicaragua

Dedicatorias

A Dios, por haberme guiado a lo largo de este camino, por darme fortaleza en los momentos difíciles, sabiduría en cada etapa y la oportunidad de culminar este importante logro en mi vida.

A mi madre, **Ana Luna**, con todo mi amor, respeto y gratitud. Este trabajo representa la culminación de una de las etapas más importantes de mi formación académica y, con ella, el alcance del nivel profesional que tanto deseaba para mí. Gracias por su apoyo incondicional, sus sacrificios, sus consejos y por creer en mí incluso en los momentos en que el camino parecía más difícil. Este logro también es suyo.

A mi abuela, **Sonia Silva**, por su cariño, sus enseñanzas y su presencia invaluable a lo largo de mi vida. Su ejemplo de esfuerzo y fortaleza ha sido una inspiración constante en mi formación personal y académica.

A mi tío, **Óscar Pérez**, por su apoyo, sus palabras de aliento y por acompañarme de una u otra manera en este proceso. Su respaldo ha sido significativo en este recorrido.

Dedico este trabajo a cada uno de ustedes, porque han sido parte fundamental de mi vida y de este logro que hoy alcanzo con profundo agradecimiento.

Espinosa Luna, Carlos Adrián

Primeramente, agradezco a Dios por permitirme alcanzar este momento tan significativo en mi formación profesional, por brindarme salud, sabiduría y fortaleza para cumplir mis metas, y por ser siempre mi guía y protección a lo largo de este camino.

A mis padres, **Blanca Medal** y **Pedro Rivas**, por su apoyo incondicional en todo momento, por ser los pilares fundamentales de mi vida, por sus valiosos consejos, por enseñarme el valor del esfuerzo y por su amor constante, que han sido esenciales en mi formación como persona.

A mis hermanos y sobrinos, quienes han contribuido a mi crecimiento personal, por su ejemplo de dedicación y perseverancia, por sus consejos y por cada muestra de cariño que me impulsó a seguir adelante.

A mi tía **Antonia Gutiérrez**, quien durante estos cinco años me brindó su hogar con tanto amor, por su apoyo, sus consejos y su constante motivación; así como a mis primos, por acogerme con tanto cariño.

Finalmente, a todas aquellas personas que, de una u otra manera, me brindaron su apoyo y contribuyeron a la realización de este logro.

Rivas Medal, Blanca Gisell

Agradecimientos

Agradezco profundamente a Dios por concederme la vida, la fortaleza y la sabiduría necesarias para culminar esta importante etapa de mi formación profesional.

Expreso mi más sincero agradecimiento a mi familia, especialmente a mi madre, **Ana Luna**; a mi abuela, **Sonia Silva**; y a mi tío, **Óscar Pérez**, quienes han sido un pilar fundamental en mi vida. Gracias por su apoyo incondicional, su confianza, sus consejos y por acompañarme durante este proceso. Su respaldo ha sido esencial para alcanzar esta meta.

A mis padrinos, **Ramón Ramírez** y **Joy Espinoza**, les expreso mi profundo agradecimiento por formar parte de mi progreso durante la elaboración de este trabajo. Su apoyo, acompañamiento y confianza representaron un aporte significativo en este proceso académico y personal.

A mis amigos, quienes me acompañaron a lo largo de la carrera, les agradezco por compartir experiencias, aprendizajes y momentos significativos que contribuyeron a mi crecimiento académico y personal.

A mis docentes, especialmente al MSc. Juan Alonso, por su motivación, dedicación y valioso aporte a mi formación como ingeniero.

Finalmente, a todas las personas que, de una u otra manera, formaron parte de este camino, les expreso mi más profundo agradecimiento.

Espinosa Luna, Carlos Adrián

Agradezco, en primer lugar, a Dios, ya que sin su presencia en mi vida no habría sido posible llegar hasta donde estoy hoy. Por su infinita misericordia y amor, que me han permitido alcanzar esta etapa tan importante en mi desarrollo profesional.

A mi familia, mi más profundo agradecimiento por su apoyo constante, por acompañarme en este camino y hacerlo más llevadero. Gracias por su educación y amor incondicional, que han sido mi fortaleza a lo largo de cada etapa de mi vida.

Al **MSc. Juan Alonso**, por su valioso tiempo, dedicación y conocimientos compartidos, los cuales han sido fundamentales para culminar esta etapa. Gracias por su guía, compromiso y apoyo, sin los cuales este logro no habría sido posible.

A mis amigos, por ser un hogar y un refugio en los momentos necesarios; gracias por los recuerdos compartidos y por todas las enseñanzas que me han brindado.

Finalmente, agradezco a todas aquellas personas, tanto a las que están presentes como a las que ya no, que han dejado huella en mi vida, contribuyendo a mi crecimiento.

Rivas Medal, Blanca Gisell

Opinión del Catedrático Guía

La Ingeniería de las Reacciones Químicas constituye una de las áreas fundamentales de la Ingeniería Química, debido a que integra principios de cinética química, balances de materia y energía, modelado matemático y análisis dinámico de sistemas reaccionantes. En la actualidad, el uso de herramientas computacionales para la simulación de reactores representa un recurso de gran importancia tanto en el ámbito académico como técnico, ya que permite estudiar el comportamiento de procesos químicos complejos de manera segura, reproducible y eficiente.

El presente trabajo monográfico se orientó al desarrollo de un software de código abierto para la simulación de reactores batch ideales homogéneos isotérmicos y no isotérmicos, denominado ReactSim Studio. La investigación se fundamentó en modelos clásicos de Ingeniería de las Reacciones Químicas, incorporando balances de materia y energía, cinética química dependiente de la temperatura y métodos numéricos para la resolución de sistemas de ecuaciones diferenciales ordinarias.

Los bachilleres Carlos Adrián Espinosa Luna y Blanca Gisell Rivas Medal desarrollaron un trabajo riguroso y técnicamente consistente, en el cual implementaron modelos matemáticos de reactores batch en un entorno computacional basado en Python, integrando herramientas numéricas modernas y una interfaz gráfica funcional para la configuración de casos de simulación, visualización de resultados y gestión de parámetros operativos. Asimismo, realizaron procesos de validación mediante problemas de referencia de la literatura clásica de Ingeniería de las Reacciones Químicas, particularmente del texto de H. Scott Fogler, complementados con análisis de estabilidad, convergencia y desempeño numérico.

Uno de los principales aportes de este trabajo radica en el desarrollo de una herramienta abierta, trazable y extensible, capaz de relacionar de forma explícita los fundamentos teóricos del modelado de reactores con su implementación computacional y su interpretación técnica. Esto representa un aporte valioso para el fortalecimiento de herramientas académicas y de simulación dentro del ámbito de la Ingeniería Química.

El trabajo refleja dedicación, disciplina, iniciativa y una adecuada aplicación de conocimientos científicos, matemáticos y computacionales por parte de los tesistas, quienes desarrollaron una investigación alineada con los estándares académicos y técnicos de nuestra institución. Por ello, solicito respetuosamente a los honorables miembros del jurado valorar el presente trabajo como una investigación de excelente nivel académico y técnico, en correspondencia con el rigor, dedicación y calidad demostrados por los tesistas durante su desarrollo, y concedan a Carlos Adrián Espinosa Luna y Blanca Gisell Rivas Medal el título de Ingenieros Químicos.

_____

MSc. Juan Manuel Alonso Santos

Tutor

Resumen

La presente investigación tuvo como objetivo desarrollar ReactSim Studio, un software de código abierto para simular reactores batch ideales homogéneos, isotérmicos y no isotérmicos, en fase líquida, de volumen constante y con una sola reacción irreversible. El estudio se fundamentó en balances de materia y energía, leyes cinéticas de tipo potencia y la ecuación de Arrhenius.

La investigación siguió un enfoque aplicado, cuantitativo, determinístico y no experimental. Se formuló un modelo matemático general para el reactor batch no isotérmico, reduciendo el caso isotérmico como una condición particular. Las ecuaciones diferenciales fueron resueltas mediante solve_ivp de SciPy, utilizando los métodos RK45, Radau y BDF. El software fue desarrollado en Python, con interfaz gráfica en PySide6.

ReactSim Studio permite definir casos de simulación, normalizar datos al Sistema Internacional, visualizar perfiles de concentración, conversión y temperatura, gestionar archivos .reactsim y exportar resultados. La validación del simulador se realizó mediante casos de referencia, soluciones analíticas y semianalíticas, pruebas de estabilidad y convergencia numérica, así como una contrastación externa con Cantera 3.2.0 mediante casos equivalentes. Los resultados mostraron coherencia con las tendencias dinámicas esperadas y concordancia numérica frente a la herramienta externa, por lo que ReactSim Studio constituye una herramienta trazable, extensible y técnicamente consistente para el análisis de reactores batch ideales.

Abstract

This research aimed to develop ReactSim Studio, an open-source software tool for simulating ideal homogeneous batch reactors, both isothermal and non-isothermal, in liquid phase, at constant volume, and with a single irreversible reaction. The study was based on material and energy balances, power-law kinetics, and the Arrhenius equation.

The research followed an applied, quantitative, deterministic, and non-experimental approach. A general mathematical model was formulated for the non-isothermal batch reactor, with the isothermal case obtained as a particular condition. The differential equations were solved using SciPy's solve_ivp function with the RK45, Radau, and BDF methods. The software was developed in Python, with a graphical interface built in PySide6.

ReactSim Studio allows users to define simulation cases, normalize data to the International System of Units, visualize concentration, conversion, and temperature profiles, manage .reactsim files, and export results. The simulator was validated using reference cases, analytical and semi-analytical solutions, numerical stability and convergence tests, as well as an external comparison with Cantera 3.2.0 through equivalent simulation cases. The results showed consistency with the expected dynamic trends and numerical agreement with the external tool, making ReactSim Studio a traceable, extensible, and technically consistent tool for analyzing ideal batch reactors.

Índice de Contenido

I. Introducción 1
II. Objetivos 4
2.1. Objetivo General 4
2.2. Objetivos Específicos 4
III. Marco Teórico 5
3.1. Ingeniería de las Reacciones Químicas 5
3.2. Clasificación de los Reactores Químicos 6
3.3. Reactor Batch Ideal Homogéneo 7
3.3.1. Definición y Características 8
3.3.2. Descripción Física y Operación del Reactor 8
3.3.3. Aplicaciones del Reactor Batch en Procesos Químicos 10
3.3.4. Supuestos del Reactor Batch 10
3.4. Balance de Materia en Reactores Batch 12
3.4.1. Formulación del Balance de Materia 12
3.4.2. Aplicación a una Reacción Química Irreversible 13
3.4.3. Relación entre Estequiometría y Velocidad de Reacción 13
3.4.4. Interpretación Física del Balance de Materia 14
3.5. Ecuación de Diseño del Reactor Batch Ideal 15
3.5.1. Forma General de la Ecuación de Diseño 15
3.5.2. Relación entre la Ecuación de Diseño y la Conversión 16
3.5.3. Importancia de la Ecuación de Diseño en la Simulación 17
3.6. Balance de Energía en Reactores Batch 18
3.6.1. Formulación del Balance de Energía 18
3.6.2. Calor Absorbido o Generado por la Reacción 19
3.6.3. Transferencia de Calor con el Entorno 20
3.6.4. Ecuación Final del Balance de Energía 20
3.6.5. Interpretación Física del Balance de Energía 20
3.6.6. Modos de Operación Térmica de un Reactor 21
3.7. Acoplamiento entre Balances de Materia y Energía 22
3.7.1. Naturaleza del Acoplamiento Materia-Energía 23
3.7.2. Sistema de Ecuaciones Diferenciales Acopladas 23
3.7.3. Implicaciones Matemáticas del Acoplamiento 24

3.7.4. Interpretación Física del Acoplamiento 24
3.8. Cinética Química y Parámetros Cinéticos 25
3.8.1. Velocidad de Reacción y Ley Cinética 25
3.8.2. Orden de Reacción 25
3.8.3. Ecuación de Arrhenius 26
3.8.4. Significado Físico de los Parámetros Cinéticos 26
3.8.5. Unidades de la Constante Cinética Según el Orden de Reacción ... 27
3.8.6. Relación entre Cinética y Comportamiento Térmico del Reactor 27
3.8.7. Consideraciones Prácticas para la Simulación 28
3.9. Métodos Numéricos para la Solución del Modelo 28
3.9.1. Naturaleza del Sistema de Ecuaciones Diferenciales Ordinarias 28
3.9.2. Necesidad de Métodos Numéricos en Reactores Batch Ideales Isotérmicos y no Isotérmicos 29
3.9.3. Métodos Explícitos e Implícitos 29
3.9.4. Método de Runge-Kutta de Orden Adaptable (RK45) 30
3.9.5. Métodos Implícitos Radau y BDF 30
3.9.6. Rigidez del Sistema y Criterio de Selección del Método Numérico .. 31
3.9.7. Tolerancias, Estabilidad y Convergencia Numérica 31
3.10. Variables de Desempeño y Criterios de Evaluación del Reactor 32
3.10.1. Concentración y Perfiles Temporales de Especie 32
3.10.2. Conversión y Especie de Referencia 32
3.10.3. Reactivo Limitante 33
3.10.4. Evolución de la Temperatura del Reactor 33
3.10.5. Temperatura Máxima Permisible 34
3.10.6. Tiempo Recomendado de Operación o Parada 34
3.10.7. Interpretación de los Resultados de Simulación 34
3.11. Simulación Computacional de Reactores Batch 35
3.11.1. Papel de la Simulación en el Análisis Técnico de Reactores 35
3.11.2. Relación entre el Modelo Matemático, Método Numérico y Software 36
3.11.3. Ventajas de las Herramientas Abiertas y Trazables en Ingeniería Química 36
3.11.4. Cantera como Herramienta Abierta de Simulación Química 37
3.12. Síntesis del Modelo Teórico y Transición a la Metodología 38
IV. Metodología 39

4.1. Enfoque Metodológico de la Investigación 39
4.2. Modelado Matemático del Reactor Batch Ideal (Modelo Isotérmico y no Isotérmico) 40
4.2.1. Definición del Sistema y Supuestos del Modelo 40
4.2.2. Variables del Modelo 41
4.2.3. Balance de Materia para el Reactor Batch 42
4.2.4. Dependencia de la Cinética con la Temperatura 42
4.2.5. Balance de Energía del Reactor Batch (Formulación no Isotérmica) 42
4.2.6. Condiciones Iniciales 43
4.3. Estrategia de Resolución Numérica 43
4.3.1. Planteamiento del Problema Numérico (IVP) 44
4.3.2. Selección del Integrador y Criterio de Rigidez 45
4.3.3. Control del Error y Tolerancias Numéricas 45
4.3.4. Discretización Temporal y Gestión de Salidas 46
4.3.5. Consistencia Física y Validaciones Internas 47
4.4. Diseño Algorítmico del Simulador, Casos de Uso, Parámetros de Entrada y Sistema de Unidades 47
4.4.1. Taxonomía de Casos Soportados por el Simulador (Alcance Funcional) 48
4.4.2. Sistema Fijo de Unidades y Normalización de Entradas 48
4.4.3. Parámetros de Entrada por Tipo de Caso 50
4.4.4. Algoritmo General del Simulador (Diseño y Proceso) 51
4.4.5. Biblioteca de Problemas con Resultados Esperados para la Validación del Simulador 52
4.4.6. Métodos Numéricos Aplicados (Criterio Metodológico del Software) 52
4.5. Interfaz de Entrada, Gestión de Casos y Criterios de Aceptación del Simulador 53
4.5.1. Estructura de la Interfaz de Entrada por Tipo de Caso 53
4.5.2. Gestión de Unidades y Consistencia Dimensional 54
4.5.3. Gestión de Casos y Persistencia de Configuración 55
4.5.4. Resultados Generados por el Simulador 55
4.5.5. Criterios de Aceptación y Validación Técnica del Simulador 56
4.5.6. Validación del Simulador 56
4.5.7. Contrastación Externa con Cantera 57
4.5.8. Relación con los Objetivos de Trabajo 58

4.6. Arquitectura Conceptual del Simulador y Criterios de Diseño 59
4.6.1. Nivel Físico-Matemático del Modelo del Reactor 59
4.6.2. Nivel Numérico-computacional (Resolución del Modelo) 60
4.6.3. Nivel de Interfaz, Gestión de Casos e Interpretación Visual 61
4.6.4. Relación entre la Arquitectura y Metodología 61
4.7. Síntesis y Alcance Metodológico del Trabajo 62
4.8. Entregables del Proyecto 63
V. Resultados, Implementación y Discusión 65
5.1. Implementación del Algoritmo Interno de ReactSim Studio 65
5.2. Descripción del Algoritmo de Simulación Implementado 66
5.2.1. Construcción del Caso de Simulación 67
5.2.2. Validación y Normalización del Caso 67
5.2.3. Construcción de la Función Cinética 68
5.2.4. Resolución Numérica del Sistema Dinámico 68
5.2.5. Postproceso y Generación de Resultado Final 69
5.2.6. Presentación y Trazabilidad de Resultados en la Interfaz 69
5.3. Flujograma Interno del Simulador 70
5.3.1. Flujograma General de Operación 71
5.3.2. Flujograma de Validación y Normalización del Caso 72
5.3.3. Flujograma de Integración Numérica y Postproceso 72
5.3.4. Consideraciones Sobre la Inclusión del Flujo Completo 73
5.4. Estructura Modular del Software 74
5.4.1. Módulo de Simulación 74
5.4.2. Módulo de Persistencia y Casos de Referencia 75
5.4.3. Módulo de Interfaz Gráfica 76
5.4.4. Integración entre Módulos 76
5.5. Interfaz Desarrollada y Funcionalidades Implementadas 78
5.5.1. Organización General de la Interfaz 78
5.5.2. Configuración del Caso y Control de Simulación 79
5.5.3. Visualización e Interpretación de Resultados 80
5.5.4. Gestión de Casos, Benchmarks y Exportación 80
5.6. Validación Mediante Soluciones Analíticas y Resultados de ReactSim Studio 81
5.6.1. Casos Isotérmicos con Solución Analítica Cerrada 81

5.6.2. Casos no Isotérmicos con Solución Analítica Adaptada a Partir de Fogler 83
5.6.3. Caso de Referencia Bibliográfica: Ejemplo 9-1 Fogler 84
5.6.4. Contrastación Externa de Resultados con Cantera..... 85
5.7. Desempeño Numérico del Simulador 88
5.7.1. Comparación Entre los Métodos de Integración 88
5.7.2. Comportamiento de la Selección Automática del Solver..... 89
5.7.3. Sensibilidad a Tolerancias Numéricas..... 90
5.7.4. Discusión del Desempeño Numérico 92
5.8. Discusión General de Resultados y Limitaciones Actuales..... 92
VI. Conclusiones 95
VII. Recomendaciones..... 97
VIII. Bibliografía..... 98
IX. Anexos 100

Índice de Figuras

Figura 1. Representación esquemática de la operación de los reactores Batch, CSTR y PFR 6

Figura 2. Esquema representativo de un reactor batch con chaqueta de transferencia de calor 9

Figura 3. Representación conceptual de la evolución temporal de la concentración en un reactor batch ideal 14

Figura 4. Esquema del balance de energía en un reactor batch con chaqueta de transferencia de calor 21

Figura 5. Flujo general de operación de ReactSim Studio 66

Figura 6. Flujograma general de operación de ReactSim Studio 71

Figura 7. Flujograma de validación y normalización del caso de simulación . 72

Figura 8. Flujograma de integración numérica y postproceso en ReactSim Studio 73

Figura 9. Organización modular de ReactSim Studio y relación funcional entre sus componentes 77

Figura 10. Interfaz principal de ReactSim Studio 79

Figura 11. Vista de resultados de simulación en ReactSim Studio 80

Figura 12. Resultado del caso Fogler Ejemplo 9-1 en ReactSim Studio 85

Figura 13. Comparación del costo numérico de RK45, Radau y BDF en casos representativos de ReactSim Studio 89

Figura 14. Efecto del ajuste de rtol y atol sobre el costo numérico y la estabilidad del resultado. 91

Índice de Tablas

Tabla 1. Unidades de kT y $k0$ según el orden de reacción 27

Tabla 2. Variables del modelo matemático del reactor batch 41

Tabla 3. Criterio práctico de selección del solver y tolerancias 46

Tabla 4. Sistema interno de unidades adoptado por el simulador 48

Tabla 5. Organización modular del software y función principal de sus componentes 77

Tabla 6. Comparación entre soluciones analíticas isotérmicas y resultados de ReactSim Studio 82

Tabla 7. Comparación entre soluciones analíticas no isotérmicas adaptadas y resultados en ReactSim Studio 83

Tabla 8. Comparación entre la solución de Fogler y el resultado de ReactSim Studio para el Ejemplo 9-1. 85

Tabla 9. Comparación Externa entre ReactSim Studio y Cantera para el Caso Isotérmico de Segundo Orden 86

Nota. Elaboración propia. 86

Tabla 10. Comparación Externa entre ReactSim Studio y Cantera para el Caso Adiabático Basado en Fogler 87

Tabla 11. Comparación de métodos numéricos implementados en ReactSim Studio 89

Tabla 12. Comportamiento de la selección automática del solver en ReactSim Studio 90

Tabla 13. Sensibilidad del caso no isotérmico adiabático frente a distintas tolerancias numéricas 91

Índice de Anexos

Anexo A. Código DOT del Flujograma Interno de ReactSim Studio 100

Anexo B. Soluciones Analíticas y Semianalíticas de Referencia para la Validación de ReactSim Studio 103

Anexo C. Evidencia del Repositorio Público y Licencia de ReactSim Studio .115

Anexo D. Recursos de Acceso, Instalación y Ejecución de ReactSim Studio 121

Anexo E. Identidad Visual Básica de ReactSim Studio 125

Anexo F. Guía de Usuario de ReactSim Studio 127

Anexo G. Códigos y Configuración de Casos Equivalentes en Cantera 131

I. Introducción

La Ingeniería de las Reacciones Químicas constituye una de las áreas fundamentales de la Ingeniería Química, debido a que permite analizar, diseñar y evaluar sistemas en los que ocurren transformaciones químicas bajo condiciones controladas. Dentro de este campo, el reactor batch ideal homogéneo representa un modelo de gran importancia para el estudio del comportamiento dinámico de las reacciones, ya que permite relacionar de manera directa la cinética química con la evolución temporal de variables como la concentración, la conversión y la temperatura. Su relevancia no se limita al ámbito académico, sino que también se vincula con procesos reales de fase líquida donde se requiere flexibilidad operativa, control del proceso y análisis de la respuesta del sistema en función del tiempo.

El análisis de reactores batch isotérmicos y no isotérmicos se fundamenta en balances de materia, balances de energía y leyes cinéticas. Sin embargo, aún bajo condiciones idealizadas, la incorporación de efectos térmicos puede conducir a sistemas de ecuaciones diferenciales ordinarias acopladas, especialmente cuando la constante cinética depende de la temperatura mediante la ecuación de Arrhenius. En estos casos, la evolución de la concentración, la velocidad de reacción y la temperatura quedan interrelacionadas, por lo que la obtención de soluciones analíticas suele ser limitada o impracticable. En consecuencia, el uso de métodos numéricos resulta necesario para describir de manera consistente la respuesta dinámica del reactor frente a distintas condiciones cinéticas, térmicas y operativas.

Autores clásicos como Smith (1981), Levenspiel (1999) y Fogler (2016) han destacado la importancia del modelado matemático, la interpretación física de los resultados y el uso de herramientas computacionales en el análisis de reactores químicos. Smith desarrolló la formulación de balances de materia y energía en sistemas reaccionantes; Levenspiel resaltó el valor de los modelos ideales como base para el desarrollo del criterio ingenieril; y Fogler incorporó de forma sistemática el uso de métodos numéricos para estudiar problemas transitorios, sensibilidad paramétrica y casos cuya solución analítica resulta restringida. Estos aportes consolidan la simulación como un recurso técnico relevante para complementar el análisis teórico y comprender el desempeño de los reactores.

Con el avance de la computación científica, han surgido plataformas comerciales y herramientas abiertas para la simulación de procesos y sistemas reaccionantes. Programas comerciales como Aspen Plus y gPROMS ofrecen capacidades avanzadas, pero suelen operar mediante modelos cerrados y estructuras altamente automatizadas, lo que puede dificultar la trazabilidad entre las hipótesis del modelo, las ecuaciones implementadas, el método numérico utilizado y los resultados obtenidos. En contraste, herramientas de código abierto como DWSIM, Cantera y bibliotecas científicas como SciPy han demostrado la utilidad de los entornos abiertos para el modelado, la simulación y el análisis numérico en Ingeniería Química. Particularmente, Cantera constituye una referencia relevante por tratarse de una herramienta abierta orientada a problemas de cinética química, termodinámica, transporte y simulación de reactores, con soporte para su uso desde

Python y otros lenguajes de programación. Esta característica permite emplearla como un entorno externo de comparación para casos dinámicos equivalentes, sin perder de vista que su alcance general difiere del enfoque especializado adoptado por ReactSim Studio.

A pesar de estos avances, muchas herramientas abiertas se orientan a simulación general de procesos, mecanismos cinéticos complejos, combustión, redes de reactores o aplicaciones particulares de investigación. Por ello, resulta pertinente desarrollar una herramienta especializada, clara y trazable para el análisis de reactores batch ideales homogéneos, que permita integrar los fundamentos de la Ingeniería de las Reacciones Químicas con su implementación computacional. Este enfoque resulta especialmente útil en contextos académicos y técnicos, donde no solo interesa obtener resultados numéricos, sino también comprender su relación con el modelo físico-matemático y el método de solución empleado.

En respuesta a esta necesidad, el presente trabajo desarrolla ReactSim Studio, un software de código abierto orientado a la simulación de reactores batch ideales homogéneos, en fase líquida, de volumen constante y con una sola reacción química irreversible. La herramienta permite analizar casos isotérmicos y no isotérmicos mediante modelos basados en balances de materia y energía, leyes cinéticas de tipo potencia, dependencia térmica mediante la ecuación de Arrhenius y métodos numéricos para la resolución de problemas de valor inicial. Su propósito es ofrecer una plataforma técnica, reproducible y extensible que vincule teoría, modelado, simulación e interpretación de resultados.

El desarrollo del simulador se realizó en Python, empleando la función solve_ivp de SciPy para la integración de ecuaciones diferenciales ordinarias mediante los métodos RK45, Radau y BDF, con el fin de abordar tanto problemas no rígidos como escenarios potencialmente rígidos. Además, se incorporó una interfaz gráfica desarrollada con PySide6, orientada a facilitar la definición de casos, el ingreso de parámetros cinéticos, térmicos y operativos, la ejecución de simulaciones y la visualización estructurada de resultados.

La validación de ReactSim Studio se realizó mediante casos de referencia y problemas representativos de la literatura clásica, especialmente del texto *Elements of Chemical Reaction Engineering* de H. Scott Fogler. Este proceso se complementó con pruebas de estabilidad, convergencia y consistencia numérica, verificando que los resultados obtenidos fueran coherentes con soluciones analíticas disponibles y con las tendencias esperadas del modelo. Adicionalmente, se incorporó una comparación externa con Cantera mediante casos equivalentes configurados con pseudoespecies y cinéticas simplificadas, con el propósito de contrastar los resultados de ReactSim Studio frente a una herramienta abierta reconocida para simulación química. De esta manera, la investigación integra fundamentos teóricos, formulación matemática, desarrollo de software, validación numérica y comparación externa dentro de una herramienta abierta para el análisis de reactores batch ideales desde una perspectiva de Ingeniería Química.

En conjunto, ReactSim Studio se plantea como una contribución académica y técnica que fortalece la relación entre teoría de reactores, modelado físico-

matemático y simulación computacional. Su carácter de código abierto permite revisar, modificar y ampliar el programa en futuros trabajos, favoreciendo la transparencia del cálculo y la evolución del simulador hacia modelos más complejos y nuevas estrategias numéricas.

II. Objetivos

2.1. Objetivo General

Desarrollar un software de código abierto para la simulación de reactores batch ideales homogéneos en fase líquida, bajo condiciones isotérmicas y no isotérmicas, de volumen constante y con una sola reacción química irreversible, basado en modelos de cinética química y balances de materia y energía.

2.2. Objetivos Específicos

- Formular modelos matemáticos basados en balances de materia y energía para describir el comportamiento dinámico de reactores batch ideales homogéneos en fase líquida, de volumen constante, operando bajo condiciones isotérmicas y no isotérmicas, con una sola reacción química irreversible.
- Implementar numéricamente los modelos formulados mediante métodos de integración de ecuaciones diferenciales ordinarias, empleando bibliotecas científicas del lenguaje de programación Python.
- Desarrollar una herramienta computacional con interfaz gráfica que permita al usuario ingresar parámetros cinéticos, térmicos y operativos, ejecutar simulaciones y visualizar resultados de manera clara y consistente con el modelo teórico formulado.
- Validar el software desarrollado mediante la simulación de problemas de referencia seleccionados del texto *Elementos de Ingeniería de las Reacciones Químicas* de H. Scott Fogler, complementando este proceso con pruebas de estabilidad, convergencia y consistencia numérica de los métodos implementados, y con la contrastación de resultados frente a casos equivalentes resueltos en Cantera como herramienta externa de simulación.

III. Marco Teórico

3.1. Ingeniería de las Reacciones Químicas

La Ingeniería de las Reacciones Químicas constituye una de las áreas fundamentales de la Ingeniería Química, ya que proporciona los principios teóricos y metodológicos necesarios para el análisis, modelado y diseño de sistemas en los que ocurren transformaciones químicas. Su propósito central es describir, predecir y evaluar el comportamiento de los sistemas reaccionantes bajo condiciones específicas de operación, de manera que puedan establecerse criterios técnicos para su diseño, control y optimización (Smith, 1981; Levenspiel, 1999).

De acuerdo con Smith (1981), el estudio de los reactores químicos debe fundamentarse en la formulación de modelos matemáticos capaces de representar de forma adecuada los fenómenos físicos y químicos que intervienen en el proceso reaccionante. Dichos modelos permiten cuantificar variables de interés ingenieril, tales como la concentración de las especies, la conversión, la velocidad de reacción y el comportamiento térmico del sistema. En este sentido, el reactor químico no se estudia únicamente como un equipo físico, sino como un sistema cuya respuesta puede describirse mediante relaciones entre balances de conservación, ecuaciones cinéticas y condiciones de operación.

Levenspiel (1999) señala que el análisis de reactores no debe reducirse únicamente a la resolución formal de ecuaciones, sino que debe orientarse también a la interpretación física de los resultados obtenidos. Esta perspectiva resulta especialmente importante en la formación y práctica de la Ingeniería Química, puesto que permite relacionar la respuesta matemática del modelo con el comportamiento real del reactor, identificar la influencia de los parámetros del sistema y comprender el efecto de las condiciones de operación sobre el desempeño del proceso.

En términos generales, la estructura de modelado empleada en la Ingeniería de las Reacciones Químicas se basa en la aplicación del principio de conservación a una especie química o a una propiedad extensiva del sistema. De manera compacta, este principio puede expresarse como:

$$Acumulación = Entrada - Salida + Generación\ (por\ reacción)$$

A partir de esta relación general, el reactor químico puede concebirse como un volumen de control cuya descripción matemática se obtiene al combinar los balances correspondientes con una ley cinética que represente la velocidad de reacción. Cuando el sistema opera bajo condiciones no isotérmicas, la velocidad de reacción depende no solo de la composición química, sino también de la temperatura, lo que introduce un acoplamiento entre los balances de materia y energía. Fogler (2016) destaca que este acoplamiento constituye uno de los aspectos más relevantes en el análisis dinámico de reactores, debido a la influencia simultánea de los fenómenos cinéticos y térmicos sobre la evolución del sistema.

Según Levenspiel (1999) y Fogler (2016), el uso de modelos ideales permite estudiar de forma aislada los efectos cinéticos y energéticos del proceso sin incorporar, en una primera aproximación, complejidades adicionales asociadas a gradientes espaciales, resistencias de transferencia de masa o comportamientos hidrodinámicos no ideales. Esta simplificación resulta particularmente útil cuando se desea establecer una base teórica clara para la formulación de modelos y su posterior implementación computacional. En el presente trabajo, dicho marco general se particulariza al estudio de reactores batch ideales homogéneos, tanto isotérmicos como no isotérmicos, como base para el desarrollo del modelo matemático y del simulador propuesto.

3.2. Clasificación de los Reactores Químicos

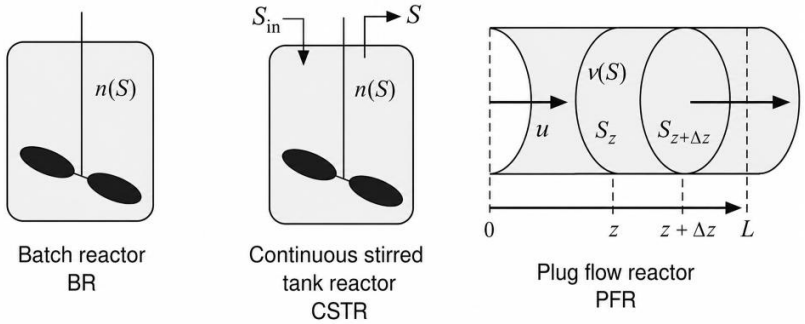
Los reactores químicos pueden clasificarse de acuerdo con su modo de operación y con sus características hidrodinámicas, criterios que determinan la forma en que se desarrollan las reacciones químicas y el enfoque de análisis requerido para su estudio. Esta clasificación constituye un elemento fundamental dentro de la Ingeniería de las Reacciones Químicas, ya que permite identificar las variables de diseño más relevantes y establecer el modelo matemático apropiado para describir el comportamiento del sistema reaccionante (Smith, 1981; Levenspiel, 1999).

De manera general, la literatura especializada reconoce tres tipos ideales de reactores químicos, ampliamente utilizados como referencia para el análisis teórico y el diseño:

- Reactores batch o por lotes.
- Reactores continuos de tanque agitado, conocidos como CSTR.
- Reactores tubulares o de flujo pistón, conocidos como PFR.

Esta clasificación puede representarse de manera esquemática en la **Figura 1**, donde se ilustran los distintos modos de operación y la forma en que ocurre el flujo de materia en cada tipo de reactor.

Figura 1.
Representación esquemática de la operación de los reactores Batch, CSTR y PFR



Nota. Adaptado de Simple. *Block diagrams of three basic reactor types: (a) Batch reactor, (b) Continuous stirred tank reactor (CSTR), and (c) Plug flow reactor*, en

Laboratory Demonstration of Abiotic Technologies for Removal of RDX from a Process Waste Stream, por D. Gent, J. L. Johnson, D. R. Felt y S. L. Larson, 2010.

En el reactor batch, el sistema opera de manera discontinua, sin flujo de entrada ni salida de materia durante el tiempo en que ocurre la reacción. Los reactivos se cargan inicialmente en el recipiente y el proceso evoluciona en función del tiempo, el cual constituye la variable de diseño principal. Debido a ello, este tipo de reactor se analiza bajo condiciones dinámicas, ya que tanto las concentraciones como la temperatura pueden variar a lo largo del tiempo (Fogler, 2016).

Por su parte, el reactor continuo de tanque agitado o CSTR opera con corrientes continuas de entrada y salida y, en condiciones normales de análisis, se estudia en estado estacionario. La agitación permite asumir una composición y una temperatura uniformes en todo el volumen del reactor, lo que simplifica el planteamiento de los balances de materia y energía al eliminar gradientes espaciales dentro del sistema (Smith, 1981).

En contraste, el reactor tubular o de flujo pistón se caracteriza por el desplazamiento continuo del fluido a lo largo de un conducto, de modo que las propiedades del sistema, como la concentración y la temperatura, varían con la posición axial. En este caso, el análisis se fundamenta en balances diferenciales espaciales que permiten describir la evolución del sistema a lo largo del reactor (Levenspiel, 1999).

La comparación entre estos tres modelos ideales resulta de gran importancia, ya que cada uno representa una forma distinta de interacción entre el tiempo, el espacio, la mezcla y la reacción química. En el caso particular del presente trabajo, el interés se centra en el reactor batch ideal homogéneo, debido a que su formulación permite estudiar con claridad el comportamiento dinámico de sistemas reaccionantes isotérmicos y no isotérmicos, así como la interacción entre los balances de materia y energía.

3.3. Reactor Batch Ideal Homogéneo

El reactor batch ideal homogéneo constituye uno de los modelos más importantes dentro del análisis de reactores químicos, tanto por su simplicidad conceptual como por su utilidad para estudiar la evolución temporal de sistemas reaccionantes. A diferencia de los reactores continuos, en el reactor batch la masa reaccionante se carga al inicio del proceso y permanece dentro del recipiente durante todo el tiempo de reacción, sin intercambio de materia con el entorno. Esta característica hace que su comportamiento se describa naturalmente como un problema dinámico dependiente del tiempo.

En el presente trabajo, el reactor batch se estudia bajo el enfoque de sistema ideal homogéneo, en fase líquida, con volumen constante y con una sola reacción química irreversible. Esta idealización permite formular modelos matemáticos claros y tratables, capaces de representar la evolución de las concentraciones y de la temperatura sin incorporar complejidades adicionales asociadas a gradientes espaciales, fenómenos multifásicos o comportamientos hidrodinámicos no ideales.

3.3.1. Definición y Características

Un reactor batch puede definirse como un sistema cerrado con respecto a la materia, en el cual los reactivos se introducen al inicio del proceso y la reacción química se deja evolucionar durante un intervalo de tiempo determinado, sin flujo de entrada ni salida mientras ocurre la transformación. Una vez finalizado el proceso, los productos son retirados del reactor.

Para fines de análisis teórico, el reactor batch ideal homogéneo se describe mediante un conjunto de supuestos simplificadores ampliamente aceptados en la literatura clásica (Smith, 1981; Levenspiel, 1999; Fogler, 2016):

- **Mezcla perfecta:** las concentraciones y la temperatura son uniformes en todo el volumen del reactor en cada instante de tiempo.
- **Ausencia de gradientes espaciales:** no existen variaciones de composición ni de temperatura con la posición dentro del sistema.
- **Sistema cerrado respecto a la materia:** no hay flujo de entrada ni salida durante el tiempo de reacción.
- **Sistema homogéneo:** el análisis se restringe a una sola fase, sin considerar interfaces o resistencias interfaciales.
- **Volumen constante:** el volumen del reactor permanece constante durante el proceso.
- **Operación isotérmica o no isotérmica:** el reactor puede analizarse tanto a temperatura constante como con temperatura variable en el tiempo.
- **Reacción única e irreversible:** el modelo considera una sola reacción química directa, descrita mediante una ley cinética apropiada.

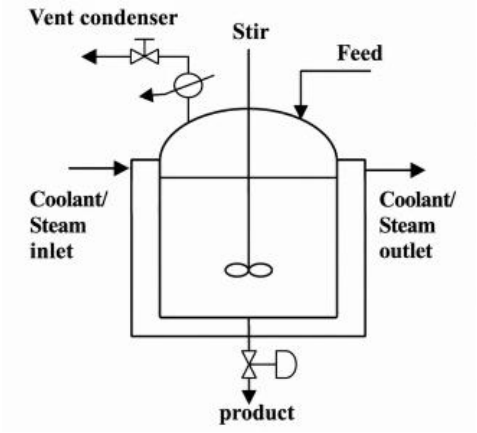
Bajo estos supuestos, el comportamiento del reactor puede expresarse mediante ecuaciones diferenciales ordinarias dependientes únicamente del tiempo. Esta simplificación reduce de forma considerable la complejidad del problema y permite centrar el análisis en la interacción entre cinética química, conversión y comportamiento térmico.

Desde el punto de vista del diseño, una característica distintiva del reactor batch es que el tiempo de operación se convierte en la variable principal de análisis. Mientras que en los reactores continuos el diseño suele relacionarse directamente con el volumen o con el tiempo de residencia, en el reactor batch el interés se orienta hacia la determinación del tiempo requerido para alcanzar una conversión, una concentración final o una condición térmica deseada.

3.3.2. Descripción Física y Operación del Reactor

Desde el punto de vista físico, el reactor batch ideal homogéneo puede representarse como un recipiente agitado, provisto de un sistema de mezclado que garantiza la uniformidad de la composición y, en muchos casos, de un dispositivo de intercambio de calor, como una chaqueta o un serpentín. Esta configuración se muestra de manera esquemática en la **Figura 2**.

Figura 2.
Esquema representativo de un reactor batch con chaqueta de transferencia de calor



Nota. Adaptado de Schematic. Diagram of a jacketed batch reactor, en *Optimal operation policies in batch reactors*.

En este tipo de reactor, los reactivos son cargados inicialmente al recipiente y, una vez iniciado el proceso, la agitación promueve una mezcla homogénea del contenido. Según Levenspiel (1999), la agitación constituye un elemento esencial dentro del modelo ideal, ya que respalda la hipótesis de mezcla perfecta y permite asumir que cualquier punto del reactor representa adecuadamente el estado global del sistema.

Adicionalmente, el reactor puede disponer de una chaqueta o serpentín de intercambio de calor, lo que permite controlar la temperatura durante la operación. Este aspecto resulta especialmente importante en sistemas no isotérmicos, donde la temperatura varía como consecuencia del calor liberado o absorbido por la reacción química y del intercambio térmico con el entorno. Tales variaciones influyen directamente sobre la velocidad de reacción y, por tanto, sobre la dinámica global del proceso (Fogler, 2016).

Desde el punto de vista operativo, el funcionamiento del reactor batch comprende tres etapas principales: carga de reactivos, reacción durante un intervalo de tiempo determinado y descarga final de productos. Durante la etapa de reacción no existe flujo de materia hacia el interior ni hacia el exterior del sistema, por lo que la evolución de las variables depende exclusivamente de las condiciones iniciales, de la cinética química y, cuando corresponde, de las interacciones energéticas del reactor con su entorno.

En consecuencia, el análisis del reactor batch se centra en la descripción de la evolución temporal de las variables de estado, particularmente la concentración de las especies y la temperatura del sistema. A diferencia de los reactores continuos, cuyo estudio se enfoca frecuentemente en el estado estacionario, en el reactor batch el tiempo constituye la variable independiente principal, y el problema de diseño consiste en determinar la trayectoria del sistema y el tiempo necesario para alcanzar una condición de operación deseada (Smith, 1981).

3.3.3. Aplicaciones del Reactor Batch en Procesos Químicos

El reactor batch posee una amplia aplicación en la industria química, especialmente en procesos donde se requiere flexibilidad operativa, producción por lotes y control preciso de las condiciones de reacción. Este tipo de reactor es común en la fabricación de productos farmacéuticos, resinas, polímeros especiales, colorantes, aditivos, productos de química fina y otras sustancias de alto valor agregado, en las cuales la variabilidad en la formulación o el tamaño de producción hace menos conveniente una operación continua (Smith, 1981; Fogler, 2016).

Una de las principales ventajas del reactor batch radica en su versatilidad. Al operar por lotes, permite modificar con relativa facilidad las condiciones de carga, la composición de la mezcla, el tiempo de reacción y la estrategia de control térmico entre una operación y otra. Esta característica resulta especialmente útil en procesos donde se requiere fabricar distintas formulaciones en un mismo equipo o donde el seguimiento de la calidad del producto exige un control riguroso de la evolución temporal del sistema.

Desde el punto de vista técnico, el reactor batch también es relevante porque permite estudiar de manera directa la relación entre tiempo de operación y transformación química. En este tipo de reactor, variables como la concentración de reactivos, la formación de productos, la conversión y la temperatura evolucionan continuamente a lo largo del proceso, lo que lo convierte en un sistema adecuado para el análisis dinámico de reacciones homogéneas. En consecuencia, constituye un modelo particularmente útil para examinar la influencia de la cinética, la estequiometría y los efectos térmicos sobre el comportamiento global del reactor.

Además de su importancia industrial, el reactor batch ocupa un lugar significativo dentro del análisis y simulación de procesos químicos, ya que ofrece una estructura matemática clara para el planteamiento de balances de materia y energía dependientes del tiempo. Esta característica facilita su implementación computacional y permite utilizarlo como base para el desarrollo de herramientas de simulación orientadas al estudio técnico del comportamiento de sistemas reaccionantes.

En el contexto del presente trabajo, la selección del reactor batch responde precisamente a esa doble relevancia: por una parte, su aplicabilidad en procesos químicos reales y, por otra, su idoneidad como sistema de referencia para integrar de manera explícita la formulación matemática, la resolución numérica y la interpretación técnica de resultados en un entorno de simulación computacional.

3.3.4. Supuestos del Reactor Batch

La formulación del reactor batch ideal homogéneo se apoya en un conjunto de supuestos que permiten simplificar la descripción del sistema real sin perder los fenómenos fundamentales que gobiernan su comportamiento. Estos supuestos son necesarios para obtener un modelo matemático tratable y coherente con el objetivo de analizar la evolución temporal de las variables principales del proceso, particularmente la concentración y la temperatura (Levenspiel, 1999; Fogler, 2016).

El primer supuesto es la mezcla perfecta, según el cual la composición y la temperatura son uniformes en todo el volumen del reactor en cada instante de tiempo. Esta idealización implica que no existen gradientes espaciales internos y que cualquier punto del sistema representa adecuadamente el estado global del contenido del reactor. Como resultado, las variables del modelo dependen únicamente del tiempo.

En segundo lugar, se considera que el sistema es homogéneo, es decir, que la reacción ocurre en una sola fase. Bajo esta condición no se contemplan fenómenos asociados a interfaces entre fases ni resistencias de transferencia interfacial, lo cual permite enfocar el análisis en la cinética química y en la evolución macroscópica del sistema reaccionante.

Asimismo, se adopta la hipótesis de sistema cerrado respecto a la materia durante la reacción. Esto significa que, una vez cargados los reactivos, no existe entrada ni salida de masa mientras ocurre la transformación química. En consecuencia, la variación en la cantidad de cada especie dentro del reactor se debe únicamente a su consumo o formación por reacción.

Otro supuesto importante es el de volumen constante, el cual simplifica la formulación de los balances y permite expresar el modelo en términos de concentraciones sin considerar cambios volumétricos durante el proceso. Esta hipótesis es coherente con el alcance del presente trabajo, orientado a sistemas líquidos en los cuales dicha aproximación resulta razonable.

De igual manera, se considera una sola reacción química irreversible, descrita mediante una ley cinética apropiada. Esta restricción permite relacionar de forma directa la variación de cada especie con una única velocidad de reacción y con sus respectivos coeficientes estequiométricos, evitando la complejidad adicional asociada a mecanismos múltiples o reacciones reversibles.

En cuanto al tratamiento energético, se asume que las propiedades físicas globales del sistema, como la densidad y la capacidad calorífica, pueden considerarse constantes dentro del intervalo de operación estudiado. Cuando existe intercambio de calor con el entorno, este se representa mediante una formulación global simplificada, generalmente expresada a través del producto entre el coeficiente global de transferencia de calor y el área de intercambio, junto con una temperatura externa prescrita. Esta aproximación resulta suficiente para describir el comportamiento térmico del reactor dentro del nivel de idealización adoptado.

En conjunto, estos supuestos delimitan el alcance del modelo teórico y establecen las condiciones de validez de los resultados que se obtienen mediante simulación. Lejos de restar valor al análisis, proporcionan una base ordenada y consistente para estudiar la interacción entre cinética y energía, formular el sistema de ecuaciones diferenciales ordinarias correspondiente e implementar una solución numérica acorde con los objetivos del presente trabajo.

3.4. Balance de Materia en Reactores Batch

El balance de materia constituye el fundamento para describir la evolución temporal de las especies químicas presentes en un reactor batch. Debido a que este tipo de reactor opera como un sistema cerrado respecto a la materia durante el tiempo de reacción, la formulación del balance se deriva directamente del principio de conservación, considerando únicamente los términos de acumulación y generación o consumo por reacción química (Smith, 1981; Levenspiel, 1999).

En un reactor batch ideal homogéneo, la ausencia de corrientes de entrada y salida simplifica notablemente el planteamiento del problema, ya que la variación en la cantidad de materia de cada especie depende exclusivamente del avance de la reacción. Esta característica convierte al balance de materia en la base del análisis dinámico del sistema y permite expresar el comportamiento del reactor mediante ecuaciones diferenciales ordinarias dependientes del tiempo.

3.4.1. Formulación del Balance de Materia

Para una especie química i en un reactor batch ideal, el balance general de materia puede expresarse, de forma simplificada, como:

$$Acumulación = Generación\ por\ Reacción$$

Dado que durante la operación no existe flujo de entrada ni salida de materia, el balance diferencial se escribe como:

$$\frac{dN_i}{dt} = V_{r_i}$$

Donde:

- N_i es el número de moles de la especie i.
- t es el tiempo.
- V es el volumen del reactor.
- r_i es la velocidad de formación o consumo de la especie i.

Si además se considera que el volumen del reactor permanece constante, la ecuación anterior puede expresarse en términos de concentración mediante la relación $C_i = \frac{N_i}{V}$, obteniéndose:

$$\frac{dC_i}{dt} = r_i \tag{3.1}$$

La Ecuación (3.1) indica que la variación temporal de la concentración de cada especie está gobernada directamente por la velocidad de reacción. En consecuencia, el balance de materia en un reactor batch ideal se formula como una ecuación diferencial ordinaria de primer orden, lo que pone de manifiesto el carácter dinámico del sistema y establece la base para su posterior análisis matemático y numérico.

3.4.2. Aplicación a una Reacción Química Irreversible

Considérese una reacción química irreversible del tipo:

$$aA \rightarrow bB$$

Para este caso, la velocidad de reacción puede expresarse de manera general como:

$$r = -\frac{1}{a}\frac{dC_A}{dt} = kC_A^n$$

Donde:

- k es la constante cinética.
- n el orden de la reacción.

Al sustituir esta expresión cinética en el balance de materia correspondiente a la especie A, se obtiene:

$$\frac{dC_A}{dt} = -akC_A^n$$

De manera análoga, para la especie producto B, el balance puede escribirse como:

$$\frac{dC_B}{dt} = bkC_A^n$$

Estas ecuaciones diferenciales ordinarias describen la evolución temporal de las concentraciones de reactivos y productos dentro del reactor. Su solución permite determinar cómo disminuye la concentración del reactivo a medida que avanza la reacción y cómo aumenta simultáneamente la concentración del producto, todo ello en función del tiempo de operación.

3.4.3. Relación entre Estequiometría y Velocidad de Reacción

La estequiometría de la reacción establece la proporción en que las especies son consumidas o formadas, mientras que la ley cinética determina la rapidez con la que ocurre dicha transformación. En el caso de una reacción única irreversible, ambos elementos quedan vinculados mediante una sola velocidad global de reacción.

Para la reacción general:

$$aA \rightarrow bB$$

La relación entre estequiometría y velocidad de reacción se expresa mediante la Ecuación (3.2).

$$-\frac{1}{a}\frac{dC_A}{dt} = \frac{1}{b}\frac{dC_B}{dt} = r \tag{3.2}$$

La Ecuación (3.2) muestra que el consumo del reactivo y la formación del producto no son fenómenos independientes, sino manifestaciones simultáneas de una misma transformación química. En consecuencia, la evolución temporal de cada especie

queda determinada por la misma velocidad de reacción, modulada por sus respectivos coeficientes estequiométricos.

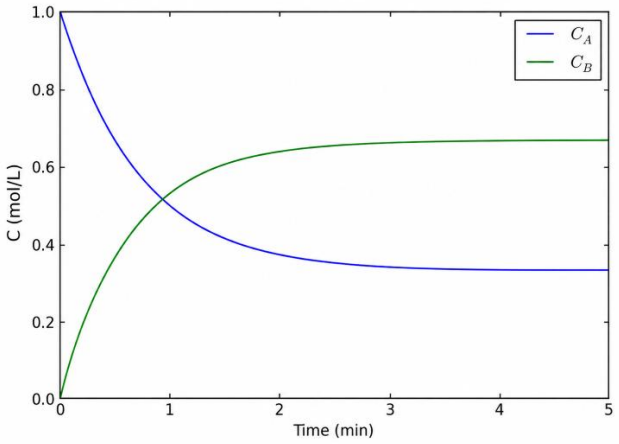
Desde el punto de vista del modelado, esta relación resulta fundamental, ya que permite construir el sistema completo de ecuaciones diferenciales a partir de una sola expresión cinética. De esta manera, la estequiometría actúa como el vínculo entre la formulación química de la reacción y la representación matemática del reactor.

3.4.4. Interpretación Física del Balance de Materia

Desde una perspectiva física, el balance de materia en un reactor batch expresa que todo cambio en la cantidad o concentración de una especie química se debe exclusivamente a su formación o consumo por reacción, dado que no existe intercambio de materia con el entorno durante la operación. Esta característica distingue al reactor batch de los reactores continuos y hace que su análisis se enfoque directamente en la evolución temporal del sistema.

La variación de la concentración ocurre únicamente en función del tiempo y no de la posición espacial, como consecuencia de la hipótesis de mezcla perfecta. Esto significa que, en cada instante, la composición del reactor puede representarse mediante un único valor de concentración para cada especie. Tal comportamiento se ilustra conceptualmente en la **Figura 3**, donde se representa la disminución temporal de la concentración de un reactivo en un reactor batch ideal.

Figura 3.
Representación conceptual de la evolución temporal de la concentración en un reactor batch ideal



Nota. Elaboración propia.

La figura se basa en el comportamiento ideal de un reactor batch, en el cual la concentración es uniforme en todo el volumen y su variación ocurre únicamente en función del tiempo.

Esta característica contrasta con la de los reactores continuos ideales. En un reactor tubular de flujo pistón, por ejemplo, la concentración varía principalmente con la

posición axial; mientras que, en un reactor de tanque agitado continuo, el análisis suele realizarse bajo condiciones de estado estacionario, donde el interés principal recae en el volumen del reactor y en el tiempo de residencia. En el reactor batch, en cambio, el tiempo de operación constituye la variable central para describir el avance de la reacción y el comportamiento global del sistema.

En este sentido, el balance de materia no solo proporciona una formulación matemática del fenómeno, sino también una interpretación física clara del proceso reaccionante. A través de él es posible comprender cómo evoluciona la composición del reactor, estimar la conversión alcanzada y establecer la base para la formulación de la ecuación de diseño y de las variables de desempeño que serán analizadas en las secciones posteriores.

3.5. Ecuación de Diseño del Reactor Batch Ideal

La ecuación de diseño del reactor batch establece la relación fundamental entre el tiempo de reacción y la conversión alcanzada para una reacción química dada. A diferencia de los reactores continuos, en los cuales el diseño suele expresarse en función del volumen del reactor o del tiempo de residencia, en el reactor batch la variable de diseño principal es el tiempo de operación, ya que el sistema evoluciona de manera transitoria desde unas condiciones iniciales definidas (Levenspiel, 1999; Fogler, 2016).

Esta ecuación constituye una herramienta central dentro del análisis de reactores batch, porque permite relacionar directamente el avance de la reacción con el tiempo requerido para alcanzar una condición de operación deseada. En consecuencia, representa un puente entre el balance de materia, la cinética química y las variables de desempeño del sistema.

3.5.1. Forma General de la Ecuación de Diseño

Para un reactor batch ideal, perfectamente mezclado y operando como un sistema cerrado, el balance de materia para un reactivo A puede expresarse como:

$$\frac{dN_A}{dt} = Vr_A$$

Donde N_A es el número de moles del reactivo A, V es el volumen del reactor y r_A es la velocidad de formación o consumo de dicha especie. Dado que se trata de un reactivo, r_A toma valores negativos cuando la especie se consume.

La conversión del reactivo A se define mediante la Ecuación (3.3):

$$X_A = \frac{N_{A0} - N_A}{N_{A0}} \tag{3.3}$$

Entonces puede escribirse:

$$N_A = N_{A0}(1 - X_A)$$

Al derivar esta expresión respecto al tiempo, se obtiene:

$$\frac{dN_A}{dt} = -N_{A0}\frac{dX_A}{dt}$$

Sustituyendo en el balance de materia:

$$N_{A0}\frac{dX_A}{dt} = Vr_A$$

Considerando que, para volumen constante, $N_{A0} = C_{A0}V$, resulta:

$$\frac{dX_A}{dt} = -\frac{r_A}{C_{A0}}$$

Integrando entre el estado inicial y una conversión X_A, se obtiene la ecuación de diseño del reactor batch ideal, presentada en la Ecuación (3.4).

$$t = C_{A0}\int_0^{X_A}\frac{dX_A}{-r_A} \tag{3.4}$$

Donde:

- t es el tiempo de reacción.
- X_A es la conversión de reactivo A.
- C_{A_0} es la concentración inicial del reactivo.
- r_A es la velocidad de reacción, evaluada en función de la conversión.

Esta expresión muestra que el tiempo necesario para alcanzar una determinada conversión depende directamente de la forma funcional de la velocidad de reacción.

3.5.2. Relación entre la Ecuación de Diseño y la Conversión

La conversión es una variable de gran importancia en Ingeniería de las Reacciones Químicas, ya que cuantifica la fracción del reactivo que ha sido transformada durante el proceso. En el reactor batch, la ecuación de diseño permite vincular esta variable de desempeño con el tiempo de operación, haciendo posible estimar cuánto debe durar la reacción para alcanzar un nivel deseado de transformación.

Para una reacción química irreversible del tipo:

$$aA \rightarrow bB$$

La ley cinética de potencia utilizada para representar el consumo del reactivo se expresa mediante la Ecuación (3.5).

$$-r_A = kC_A^n \tag{3.5}$$

Puede utilizarse la relación entre concentración y conversión, válida para un reactor batch de volumen constante:

$$C_A = C_{A0}(1 - X_A)$$

Al sustituir la ley cinética de la Ecuación (3.5) y la relación concentración-conversión en la Ecuación (3.4), se obtiene la Ecuación (3.6).

$$t = \int_0^{X_A} \frac{dX_A}{kC_{A0}^{n-1}(1 - X_A)^n} \tag{3.6}$$

La Ecuación (3.6) pone de manifiesto que la relación entre tiempo y conversión depende tanto del orden de reacción como de la concentración inicial del reactivo. Para algunos casos particulares pueden obtenerse soluciones analíticas cerradas. Por ejemplo:

- Para una reacción de **orden cero**:

$$t = \frac{C_{A0}X_A}{k}$$

- Para una reacción de **primer orden**:

$$t = -\frac{1}{k}\ln(1 - X_A)$$

- Para una reacción de **segundo orden**:

$$t = \frac{X_A}{kC_{A0}(1 - X_A)}$$

Estas expresiones son de gran utilidad porque permiten validar el comportamiento del modelo en casos simples y sirven como referencia para contrastar soluciones numéricas. Sin embargo, cuando se trabaja con órdenes fraccionarios, leyes cinéticas más complejas o condiciones no isotérmicas, la integración analítica deja de ser viable en muchos casos, por lo que se requiere el uso de métodos numéricos para resolver el problema (Smith, 1981; Levenspiel, 1999).

3.5.3. Importancia de la Ecuación de Diseño en la Simulación

La ecuación de diseño del reactor batch no solo tiene relevancia desde el punto de vista teórico, sino también desde la perspectiva computacional. En simulación, esta ecuación constituye una referencia fundamental para interpretar la relación entre el tiempo de operación y el grado de transformación alcanzado por el sistema. En particular, permite identificar la conversión como una de las variables de desempeño más importantes del reactor y justificar su uso como criterio para analizar y comparar distintos escenarios de operación.

Además, en casos isotérmicos sencillos, las expresiones analíticas derivadas de la Ecuación (3.4), y en particular de su forma sustituida con ley de potencia dada por la Ecuación (3.6), proporcionan una base de validación para el simulador, ya que permiten contrastar los resultados obtenidos mediante integración numérica con soluciones exactas conocidas. Esta comparación resulta especialmente útil en el desarrollo y verificación del software, dado que permite evaluar la consistencia del modelo implementado y la precisión de los métodos numéricos empleados.

En problemas más complejos, particularmente en reactores no isotérmicos, la ecuación de diseño sigue conservando valor conceptual, aunque su resolución directa en términos de conversión suele quedar incorporada dentro de un sistema acoplado de ecuaciones diferenciales ordinarias. En tales condiciones, la conversión deja de depender únicamente del tiempo y pasa a estar influida también por la evolución térmica del reactor, la cual modifica la velocidad de reacción a través de la dependencia cinética con la temperatura. Por ello, la ecuación de diseño constituye un punto de enlace entre el análisis clásico del reactor batch y la necesidad de formular modelos numéricos más completos para su simulación.

3.6. Balance de Energía en Reactores Batch

Cuando el reactor batch opera bajo condiciones no isotérmicas, es necesario complementar el balance de materia con un balance de energía que permita describir la evolución temporal de la temperatura del sistema. Este balance resulta esencial para analizar el efecto térmico de la reacción y su influencia directa sobre la velocidad de reacción, especialmente en aquellos casos donde el calor liberado o absorbido modifica de manera significativa la dinámica del proceso (Fogler, 2016).

En un reactor batch ideal homogéneo, el balance de energía se formula a partir de la primera ley de la termodinámica aplicada al sistema reaccionante, considerando los efectos de generación o absorción de calor por reacción y, cuando corresponda, la transferencia de calor con el entorno. De esta forma, la temperatura deja de ser una condición impuesta y pasa a constituir una variable dinámica del modelo.

3.6.1. Formulación del Balance de Energía

Para un reactor batch ideal, de volumen constante, perfectamente mezclado y operando como sistema cerrado respecto a la materia, el balance general de energía puede derivarse a partir de la primera ley de la termodinámica aplicada a un volumen de control sin flujo de masa:

$$\frac{dE}{dt} = \dot{Q} + \dot{W}$$

Bajo los supuestos de operación adoptados en este trabajo se consideran despreciables:

- Los cambios de energía cinética y potencial.
- El trabajo de frontera, dado que el volumen del reactor permanece constante.
- Otros aportes mecánicos distintos del intercambio térmico.

En estas condiciones, el balance se reduce a una relación entre la variación de energía interna del sistema y el calor intercambiado con el entorno:

$$\frac{dU}{dt} = \dot{Q}$$

Si además se asume que el fluido reaccionante posee propiedades térmicas promedio constantes, la variación de energía interna puede expresarse como:

$$\frac{dU}{dt} = \rho C_p V \frac{dT}{dt}$$

Donde:

- ρ es la densidad del sistema.
- C_p el calor específico a presión constante.
- V el volumen del reactor.
- T es la temperatura del reactor.

Sustituyendo esta relación en el balance, se obtiene la forma general:

$$\rho C_p V \frac{dT}{dt} = \dot{Q}_{reacción} + \dot{Q}_{transferencia}$$

Esta expresión indica que la variación temporal de la temperatura del reactor depende del efecto combinado entre el calor generado o absorbido por la reacción química y el calor intercambiado con el entorno.

3.6.2. Calor Absorbido o Generado por la Reacción

El término asociado al efecto térmico de la reacción representa la energía liberada o absorbida como consecuencia de la transformación química. Para una reacción única, este término puede expresarse como:

$$\dot{Q}_{reacción} = -\Delta H_r V r$$

Donde:

- ΔH_r es el calor de reacción.
- V es el volumen del reactor.
- r es la velocidad de reacción.

La convención de signos adoptada en este trabajo es la siguiente:

- $\Delta H_r < 0$: reacción exotérmica (libera calor al sistema).
- $\Delta H_r > 0$: reacción endotérmica (absorbe calor del sistema).

El signo negativo en la expresión anterior garantiza coherencia física. En una reacción exotérmica, donde $\Delta H_r < 0$, el término $-\Delta H_r V r$ resulta positivo, lo que representa un aporte de energía térmica al reactor. En cambio, en una reacción endotérmica, dicho término resulta negativo, indicando que la reacción consume energía del sistema y tiende a disminuir su temperatura si no existe compensación térmica externa.

Este término establece el vínculo directo entre el avance de la reacción y el comportamiento térmico del reactor, haciendo evidente que la evolución de la temperatura depende del ritmo con que ocurre la transformación química.

3.6.3. Transferencia de Calor con el Entorno

Cuando el reactor batch dispone de una chaqueta o de un sistema externo de calentamiento o enfriamiento, el intercambio térmico con el entorno puede modelarse mediante la ley de enfriamiento de Newton:

$$\dot{Q}_{transferencia} = -UA(T - T_c)$$

Donde:

- U es el coeficiente global de transferencia de calor.
- A es el área de transferencia.
- T_c es la temperatura del fluido de la chaqueta o del medio externo.

Este término toma signo negativo cuando $T > T_c$, indicando que el reactor pierde calor hacia el entorno, y signo positivo cuando $T < T_c$, indicando que el sistema recibe calor desde el exterior. De esta manera, el término de transferencia actúa como un mecanismo de compensación térmica que puede atenuar o intensificar la evolución de la temperatura, según la diferencia entre la temperatura del reactor y la temperatura externa.

En el caso de un reactor con intercambio térmico, la evolución de la temperatura no depende únicamente del efecto de la reacción, sino también de la capacidad del sistema de remover o suministrar calor. Por ello, este término resulta fundamental para el análisis de reactores no isotérmicos controlados térmicamente.

3.6.4. Ecuación Final del Balance de Energía

Al incorporar el calor asociado a la reacción y el intercambio térmico con el entorno, se obtiene el balance de energía no isotérmico con chaqueta, presentado en la Ecuación (3.7).

$$\rho C_p V \frac{dT}{dt} = -\Delta H_r V_r - UA(T - T_c) \tag{3.7}$$

Esta ecuación expresa que la variación de temperatura en el reactor es el resultado del balance entre dos contribuciones principales: el calor generado o absorbido por la reacción química y el calor intercambiado con el entorno.

La importancia de esta ecuación radica en que introduce de forma explícita el acoplamiento entre comportamiento térmico y cinética química. En efecto, la velocidad de reacción r puede depender de la temperatura mediante la ecuación de Arrhenius, por lo que la temperatura influye sobre la reacción y, simultáneamente, la reacción influye sobre la temperatura. Esta interacción constituye uno de los elementos más importantes en el análisis de reactores batch no isotérmicos.

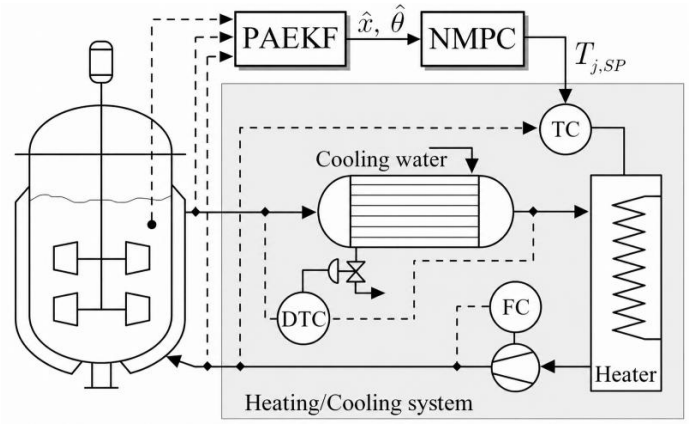
3.6.5. Interpretación Física del Balance de Energía

Desde el punto de vista físico, el balance de energía describe cómo evoluciona térmicamente el reactor a medida que avanza la reacción. Si la reacción es exotérmica, el sistema tiende a elevar su temperatura, mientras que, si es endotérmica, tiende a disminuirla en ausencia de aporte externo de calor. El

intercambio térmico con el entorno puede modificar esta tendencia natural, ya sea removiendo energía del reactor o suministrándola según las condiciones de operación.

Esta interacción entre reacción y transferencia de calor puede representarse esquemáticamente mediante la **Figura 4**, en la cual se ilustran los flujos térmicos principales involucrados en un reactor batch no isotérmico con chaqueta de intercambio de calor.

Figura 4.
Esquema del balance de energía en un reactor batch con chaqueta de transferencia de calor



Nota. Adaptado de Schematic. *Representation of the batch reactor with the heating/cooling system*, en *Nonlinear Model Predictive Control of Batch Processes: An Industrial Case Study*, por Z. K. Nagy, R. Franke y F. Allgöwer, 2005.

El esquema permite visualizar cómo el calor liberado o absorbido por la reacción puede ser compensado parcial o totalmente mediante transferencia de calor con el entorno, lo que determina la trayectoria temporal de la temperatura del reactor. Dado que la temperatura influye directamente sobre la velocidad de reacción, la evolución térmica del sistema repercute de manera inmediata en el comportamiento cinético global del proceso (Fogler, 2016).

En consecuencia, el balance de energía no debe entenderse únicamente como una ecuación auxiliar, sino como una parte esencial del modelo del reactor no isotérmico. Su incorporación permite describir fenómenos que no pueden captarse con un modelo isotérmico, tales como incrementos térmicos pronunciados, disminuciones de velocidad por enfriamiento o efectos de control térmico sobre el tiempo de reacción y la conversión alcanzada.

3.6.6. Modos de Operación Térmica de un Reactor

Desde la perspectiva del análisis térmico, el reactor batch puede estudiarse bajo distintos modos de operación, dependiendo de cómo se trate la temperatura del sistema y su interacción con el entorno.

En el modo isotérmico, se asume que la temperatura del reactor permanece constante durante todo el proceso. Esta condición puede interpretarse como una idealización o como el resultado de un control térmico suficientemente efectivo. Bajo este supuesto, no es necesario resolver una ecuación diferencial para la temperatura, y la cinética se evalúa a una temperatura fija.

En el modo no isotérmico adiabático, el reactor no intercambia calor con el entorno, por lo que:

$$UA(T - T_c) = 0$$

Para el caso adiabático, al no existir intercambio de calor con el entorno, el balance de energía se reduce a la Ecuación (3.8).

$$\rho C_p V \frac{dT}{dt} = -\Delta H_r V r \tag{3.8}$$

En este caso, toda la variación térmica del sistema se debe exclusivamente al efecto de la reacción química.

Por otro lado, en el modo no isotérmico con transferencia de calor, el reactor intercambia energía con el entorno a través de una chaqueta o de un sistema equivalente, de modo que la evolución de la temperatura queda gobernada por la ecuación completa del balance de energía. Este es el caso más general y permite estudiar escenarios de calentamiento, enfriamiento o control térmico del reactor.

La distinción entre estos modos de operación térmica es primordial tanto desde el punto de vista teórico como desde la simulación, dado que cada uno conduce a un comportamiento dinámico diferente y condiciona la forma en que se acoplan los balances de materia y energía dentro del modelo del reactor.

3.7. Acoplamiento entre Balances de Materia y Energía

En un reactor batch ideal no isotérmico, los balances de materia y energía no pueden tratarse como ecuaciones independientes, dado que ambos se encuentran intrínsecamente vinculados a través de la cinética química y de los efectos térmicos de la reacción. Este acoplamiento constituye uno de los rasgos más importantes del comportamiento dinámico del sistema y justifica la necesidad de un tratamiento matemático simultáneo mediante un sistema de ecuaciones diferenciales ordinarias acopladas.

En términos generales, la evolución de las concentraciones depende de la velocidad de reacción, mientras que la evolución de la temperatura depende del calor generado o absorbido por dicha reacción y de la transferencia de calor con el entorno. Cuando la velocidad de reacción depende a su vez de la temperatura, ambas descripciones quedan ligadas dentro de un mismo problema dinámico, cuya solución requiere considerar de forma conjunta los fenómenos químicos y térmicos que ocurren en el reactor.

3.7.1. Naturaleza del Acoplamiento Materia-Energía

El acoplamiento entre los balances de materia y energía surge principalmente por dos mecanismos físicos.

El primero corresponde a la dependencia de la velocidad de reacción con la temperatura. En los sistemas no isotérmicos, la velocidad de reacción r depende explícitamente de la temperatura del reactor a través de la ecuación de Arrhenius. En consecuencia, cualquier variación térmica modifica directamente la constante cinética y, por tanto, altera la rapidez con la que evolucionan las concentraciones de las especies.

El segundo mecanismo corresponde a la dependencia de la temperatura con la velocidad de reacción. Como se mostró en el balance de energía, el cambio térmico del sistema está gobernado por la magnitud de la velocidad de reacción, puesto que el calor liberado o absorbido por la reacción depende directamente de ella.

Como se estableció en la Ecuación (3.7), la evolución de la temperatura no solo depende de la transferencia de calor con el entorno, sino también del ritmo con que ocurre la reacción química. En consecuencia, un cambio de temperatura modifica la velocidad de reacción y, a su vez, un cambio en la velocidad de reacción modifica la temperatura del sistema. Este mecanismo de retroalimentación térmica es característico de los reactores no isotérmicos y puede conducir a comportamientos no lineales de gran importancia práctica, tales como aceleraciones térmicas, atenuaciones por enfriamiento o cambios significativos en el tiempo necesario para alcanzar una conversión dada.

3.7.2. Sistema de Ecuaciones Diferenciales Acopladas

Para una reacción única irreversible, el acoplamiento entre el balance de materia y el balance de energía puede expresarse de forma general mediante la Ecuación (3.9), donde la evolución de cada especie queda vinculada con la velocidad global de reacción a través de su coeficiente estequiométrico.

$$\begin{cases} \dfrac{dC_i}{dt} = \nu_i r(C,T) \\ \dfrac{dT}{dt} = \dfrac{-\Delta H_r V r(C,T) - UA(T - T_c)}{\rho C_p V} \end{cases} \tag{3.9}$$

Donde $C_i(t)$ es la concentración de la especie i, $T(t)$ es la temperatura del reactor, ν_i es el coeficiente estequiométrico de la especie i, $r(C,T)$es la velocidad global de reacción y $UA(T - T_c)$ representa el intercambio térmico con la chaqueta o con el entorno.

Este sistema representa el núcleo matemático del modelo del reactor batch no isotérmico desarrollado en el presente trabajo, ya que integra en una sola estructura la cinética química, el efecto térmico de la reacción y la transferencia de calor con el entorno.

3.7.3. Implicaciones Matemáticas del Acoplamiento

Desde el punto de vista matemático, el sistema de ecuaciones acopladas del reactor batch no isotérmico presenta una naturaleza no lineal. Esta no linealidad surge, principalmente, por dos razones.

En primer lugar, la velocidad de reacción depende de la temperatura mediante una relación exponencial, tal como ocurre en la ecuación de Arrhenius. En segundo lugar, la temperatura depende de la propia velocidad de reacción a través del balance de energía, lo que introduce un mecanismo de retroalimentación entre ambas variables.

Como consecuencia, el sistema resultante no suele admitir soluciones analíticas cerradas, salvo en casos altamente simplificados, como ciertos problemas isotérmicos o situaciones con cinéticas particularmente sencillas. En condiciones generales de operación no isotérmica, la resolución del problema requiere el uso de métodos numéricos de integración temporal, capaces de aproximar simultáneamente la evolución de la concentración y la temperatura en el tiempo.

Además, dependiendo de los parámetros del sistema, como la energía de activación, el calor de reacción o la intensidad del intercambio térmico, el problema puede presentar distintos niveles de sensibilidad y, en algunos casos, comportamiento rígido desde el punto de vista numérico. Esta característica justifica la selección cuidadosa de métodos de integración apropiados y el establecimiento de criterios de control de error para asegurar soluciones estables y físicamente consistentes.

3.7.4. Interpretación Física del Acoplamiento

Desde el punto de vista físico, el acoplamiento entre los balances de materia y energía implica que la evolución química y la evolución térmica del reactor no pueden interpretarse por separado. Ambas forman parte de un mismo fenómeno dinámico, en el cual la reacción modifica la temperatura y la temperatura modifica, a su vez, la rapidez de la reacción.

En una reacción exotérmica, un incremento de temperatura puede acelerar la velocidad de reacción, lo que genera a su vez una mayor liberación de calor. Si la capacidad de remoción térmica del sistema es insuficiente, esta retroalimentación puede dar lugar a aumentos térmicos importantes y a una respuesta más rápida del sistema. En una reacción endotérmica ocurre el efecto opuesto: la disminución de temperatura tiende a reducir la velocidad de reacción, pudiendo requerirse un aporte externo de calor para mantener el proceso dentro de condiciones operativas adecuadas.

Por ello, el estudio simultáneo de los balances de materia y energía resulta esencial en el análisis de reactores batch no isotérmicos. Considerar únicamente la cinética sin la evolución térmica, o viceversa, conduciría a una descripción incompleta del sistema. En consecuencia, el acoplamiento materia-energía no debe verse solo como una complejidad matemática adicional, sino como una representación más fiel del comportamiento físico real del reactor.

3.8. Cinética Química y Parámetros Cinéticos

La cinética química describe la rapidez con que ocurre una reacción y establece la relación entre la velocidad de transformación de las especies y las variables que afectan dicho proceso, particularmente la concentración y la temperatura. En el análisis de reactores batch ideales homogéneos, la formulación cinética constituye un elemento central, ya que permite cerrar matemáticamente el sistema de balances y determinar la evolución temporal de las especies químicas dentro del reactor (Fogler, 2016; Levenspiel, 1999).

En los sistemas isotérmicos, la velocidad de reacción depende principalmente de la composición del sistema. Sin embargo, en reactores no isotérmicos, la temperatura adquiere un papel adicional de gran importancia, ya que modifica de forma directa la constante cinética y, por tanto, la rapidez con la que avanza la reacción. Esta dependencia térmica es la que introduce el vínculo entre cinética y balance de energía, haciendo de la ecuación de Arrhenius una herramienta fundamental para la formulación del modelo del reactor batch no isotérmico.

3.8.1. Velocidad de Reacción y Ley Cinética

La velocidad de reacción expresa la rapidez con que una especie química es consumida o formada dentro del sistema reaccionante. Para una reacción irreversible, esta velocidad se describe mediante una ley cinética que relaciona la rapidez de transformación con la concentración de las especies participantes y, cuando corresponde, con la temperatura del sistema.

En el presente trabajo, el consumo del reactivo se representa mediante una ley cinética de potencia, definida previamente en la Ecuación (3.5). Esta formulación permite vincular directamente el balance de materia con la cinética de la reacción. En sistemas isotérmicos, la constante cinética se considera evaluada a una temperatura fija; mientras que, en sistemas no isotérmicos, dicha constante puede variar con la temperatura de acuerdo con la dependencia establecida posteriormente mediante la Ecuación (3.10).

De esta manera, la ley cinética constituye un componente esencial del modelo matemático, ya que permite calcular la velocidad de consumo o formación de especies en cada instante de la simulación y establecer la relación entre composición, temperatura y evolución temporal del reactor.

3.8.2. Orden de Reacción

El orden de reacción describe la forma en que la velocidad de reacción responde ante cambios en la concentración de los reactivos. En una ley cinética de potencia, el orden se identifica con el exponente que acompaña a la concentración de la especie reaccionante, como se observa en la Ecuación (3.5). En este caso, el parámetro n representa el orden de reacción con respecto al reactivo A.

Desde el punto de vista matemático, el orden de reacción controla la sensibilidad de la velocidad frente a la concentración y condiciona la forma de la ecuación diferencial que describe la evolución temporal del sistema. Por ejemplo, una

reacción de orden cero presenta una velocidad independiente de la concentración del reactivo; en cambio, en reacciones de primer o segundo orden, la velocidad disminuye progresivamente conforme el reactivo se consume.

Desde la perspectiva del diseño y la simulación, el orden de reacción influye directamente en la forma de la ecuación de diseño del reactor batch y en la posibilidad de obtener soluciones analíticas cerradas. En casos simples, como reacciones de orden cero, primer orden o segundo orden, pueden derivarse expresiones analíticas para la relación entre tiempo y conversión. Sin embargo, cuando se consideran órdenes no enteros, dependencia térmica de la constante cinética o modelos acoplados de materia y energía, la resolución analítica suele dejar de ser práctica, por lo que se recurre a métodos numéricos.

3.8.3. Ecuación de Arrhenius

La influencia de la temperatura sobre la velocidad de reacción constituye un aspecto fundamental en el análisis de reactores batch no isotérmicos. En el presente trabajo, dicha dependencia se representa mediante la Ecuación (3.10), correspondiente a la ley de Arrhenius, la cual ha sido ampliamente validada en la literatura de cinética química y constituye una base teórica esencial en la Ingeniería de las Reacciones Químicas (Fogler, 2016; Levenspiel, 1999).

La constante cinética se expresa como:

$$k(T) = k_0 \exp\left(-\frac{E_a}{RT}\right) \tag{3.10}$$

Donde:

- k_0 es el factor preexponencial (o factor de frecuencia).
- E_a es la energía de activación de la reacción $\left[\frac{J}{mol}\right]$
- R es la constante universal de los gases $\left[8.314 \frac{J}{molK}\right]$
- T es la temperatura absoluta del reactor $[K]$

Esta expresión refleja que la velocidad de reacción aumenta de manera exponencial con la temperatura, debido a que, al incrementarse T, aumenta la fracción de moléculas con energía suficiente para superar la barrera energética necesaria para reaccionar.

3.8.4. Significado Físico de los Parámetros Cinéticos

El factor preexponencial k_0 representa, desde una perspectiva molecular, la frecuencia efectiva de colisiones con orientación favorable para que ocurra la reacción. Aunque su interpretación no es tan directa como la de la energía de activación, constituye un parámetro esencial de la ley cinética y depende de la naturaleza de los reactivos, del medio de reacción y del mecanismo involucrado.

Por su parte, la energía de activación E_a cuantifica la barrera energética mínima que debe superarse para que la reacción tenga lugar. Valores elevados de E_a indican que la velocidad de reacción es más sensible a cambios de temperatura, mientras que valores menores implican una respuesta térmica más moderada.

Desde el punto de vista del modelado de reactores batch no isotérmicos, la energía de activación tiene una importancia especial, ya que controla la intensidad con la que la cinética responde a la evolución térmica del sistema. Por ello, este parámetro incide directamente en el grado de acoplamiento entre los balances de materia y energía y, en consecuencia, en la dinámica global del reactor.

3.8.5. Unidades de la Constante Cinética Según el Orden de Reacción

Dado que la velocidad de reacción se expresa de manera general como:

$$-r_A = k(T)C_A^n$$

Las unidades de la constante cinética dependen del orden global de la reacción. En el presente trabajo, estas unidades se expresan en el Sistema Internacional, de forma coherente con la formulación del modelo matemático y con la implementación numérica del simulador.

Tabla 1.
Unidades de $k(T)$ y k_0 según el orden de reacción

Orden (n)	*Unidades de* $k(T)$	*Unidades de* k_0
0	$mol\, m^{-3} s^{-1}$	$mol\, m^{-3} s^{-1}$
1	s^{-1}	s^{-1}
2	$m^3\, mol^{-1} s^{-1}$	$m^3\, mol^{-1} s^{-1}$

Nota. Elaboración propia con base en Fogler.

Esta consideración dimensional es de especial importancia en simulación, ya que evita inconsistencias al introducir parámetros cinéticos y garantiza coherencia entre las ecuaciones del modelo y los resultados calculados.

3.8.6. Relación entre Cinética y Comportamiento Térmico del Reactor

Al combinar la ley cinética de potencia de la Ecuación (3.5) con la dependencia térmica de Arrhenius definida en la Ecuación (3.10), se obtiene la Ecuación (3.11).

$$-r_A = k_0 \exp\left(-\frac{E_a}{RT}\right) C_A^n \tag{3.11}$$

Esta formulación muestra de manera explícita que la velocidad de reacción depende simultáneamente de la concentración del reactivo y de la temperatura del reactor. Cuando esta relación se incorpora al balance de energía no isotérmico de la Ecuación (3.7), se obtiene un sistema fuertemente acoplado, dado que:

- $T(t)$ influye sobre $r_A(t)$ a través de la constante cinética.
- $r_A(t)$ influye sobre $T(t)$ mediante el calor generado o absorbido por la reacción.

Este doble acoplamiento es el origen de la no linealidad característica de los reactores batch no isotérmicos y explica por qué la evolución de la temperatura puede modificar significativamente el tiempo de reacción, la conversión alcanzada y el comportamiento general del sistema.

3.8.7. Consideraciones Prácticas para la Simulación

En la simulación computacional de reactores batch, la formulación cinética debe implementarse de forma dimensionalmente consistente y físicamente coherente con el resto del modelo. Esto implica trabajar con unidades compatibles, emplear temperatura absoluta en la ecuación de Arrhenius y asegurar que la velocidad de reacción se evalúe a partir de concentraciones y parámetros definidos bajo una misma base de unidades.

Asimismo, la sensibilidad de la velocidad de reacción frente a la temperatura puede hacer que pequeñas variaciones térmicas generen cambios significativos en la dinámica del sistema, especialmente en reacciones exotérmicas con alta energía de activación. Por esta razón, la simulación de reactores no isotérmicos exige especial cuidado en la elección del método numérico, el control del error y la interpretación física de los resultados.

En este trabajo, la formulación cinética adoptada constituye un componente esencial del modelo del reactor batch, ya que conecta de manera directa la descripción química del proceso con el sistema de ecuaciones diferenciales ordinarias que será resuelto numéricamente. Por ello, su correcta definición no solo tiene importancia teórica, sino también metodológica y computacional.

3.9. Métodos Numéricos para la Solución del Modelo

La formulación matemática del reactor batch ideal homogéneo conduce, en general, a un sistema de ecuaciones diferenciales ordinarias cuya solución describe la evolución temporal de las concentraciones y, en condiciones no isotérmicas, también de la temperatura. Aunque en algunos casos particulares es posible obtener soluciones analíticas cerradas, la mayoría de los problemas de interés técnico requieren el uso de métodos numéricos para aproximar la respuesta del sistema con un nivel adecuado de precisión y estabilidad.

En el caso de reactores batch isotérmicos sencillos, ciertas formas de la ecuación de diseño permiten derivar expresiones exactas para la relación entre tiempo y conversión. Sin embargo, cuando la velocidad de reacción depende de la temperatura, cuando se trabaja con órdenes no enteros, o cuando el sistema incluye intercambio térmico con el entorno, el problema adquiere una naturaleza no lineal que dificulta o imposibilita su resolución analítica directa. En estos casos, los métodos numéricos constituyen la vía principal para resolver el modelo y predecir el comportamiento dinámico del reactor.

3.9.1. Naturaleza del Sistema de Ecuaciones Diferenciales Ordinarias

El modelo del reactor batch ideal homogéneo se expresa como un problema dinámico en el que las variables de estado dependen únicamente del tiempo. En el

caso isotérmico, el sistema puede representarse mediante ecuaciones diferenciales ordinarias para las concentraciones de las especies químicas. En el caso no isotérmico, se añade además una ecuación diferencial para la temperatura del reactor, obteniéndose un sistema acoplado.

Desde el punto de vista numérico, el modelo del reactor puede expresarse en la forma general de la Ecuación (3.12).

$$\frac{dy}{dt} = f(t, y) \tag{3.12}$$

Donde y representa el vector de variables de estado del sistema. Para un reactor batch isotérmico, este vector puede tomar la forma:

$$\mathrm{y}(t) = [C_1(t), C_2(t), \dots, C_m(t)]$$

Mientras que, para un reactor no isotérmico, se extiende a:

$$y(t) = [C_1(t), C_2(t), \dots, C_m(t), T(t)]$$

Esta formulación permite tratar el comportamiento del reactor como un problema de valor inicial, en el que la evolución futura del sistema queda determinada por las condiciones iniciales y por la forma funcional de los balances de materia y energía.

3.9.2. Necesidad de Métodos Numéricos en Reactores Batch Ideales Isotérmicos y no Isotérmicos

La necesidad de métodos numéricos surge cuando la solución analítica del sistema no puede obtenerse de forma cerrada o cuando su formulación se vuelve excesivamente compleja para fines prácticos. Esto ocurre con frecuencia en reactores batch no isotérmicos, donde la temperatura modifica la velocidad de reacción mediante la ecuación de Arrhenius y, simultáneamente, la reacción modifica la temperatura mediante el balance de energía.

Incluso en sistemas isotérmicos, la presencia de leyes cinéticas no lineales o expresiones de velocidad definidas de forma general puede hacer más conveniente la resolución numérica. En consecuencia, los métodos numéricos no solo constituyen una alternativa a la solución analítica, sino una herramienta esencial para estudiar escenarios de operación más realistas, analizar sensibilidad a parámetros y explorar el comportamiento del reactor bajo distintas condiciones de diseño.

Desde esta perspectiva, el problema del reactor batch se aborda como un problema de integración temporal, en el cual se busca aproximar el valor de las variables de estado en un intervalo de tiempo determinado, a partir de las ecuaciones diferenciales que gobiernan el sistema.

3.9.3. Métodos Explícitos e Implícitos

Los métodos numéricos para resolver ecuaciones diferenciales ordinarias pueden clasificarse, de forma general, en métodos explícitos e implícitos. En los métodos explícitos, el valor de la solución en el siguiente paso de tiempo se calcula

directamente a partir de información conocida del paso actual. Estos métodos suelen ser conceptualmente más simples y eficientes en problemas no rígidos, aunque pueden perder estabilidad cuando el sistema presenta cambios muy bruscos o alta sensibilidad numérica.

Por su parte, los métodos implícitos requieren resolver, en cada paso, una relación algebraica en la que la solución futura aparece de forma implícita. Aunque su implementación y costo computacional suelen ser mayores, ofrecen mejores propiedades de estabilidad en problemas rígidos o fuertemente acoplados, por lo que resultan especialmente útiles en sistemas no isotérmicos con cinética altamente sensible a la temperatura.

La elección entre métodos explícitos e implícitos depende, por tanto, de la naturaleza del problema, del nivel de no linealidad, de la sensibilidad del sistema y del comportamiento numérico observado durante la integración.

3.9.4. Método de Runge-Kutta de Orden Adaptable (RK45)

Uno de los métodos explícitos más utilizados para la solución de problemas de valor inicial es el método Runge-Kutta de orden adaptable, comúnmente identificado como RK45. Este método calcula la solución mediante una combinación de evaluaciones sucesivas de la función diferencial y ajusta automáticamente el tamaño de paso para mantener el error local dentro de una tolerancia establecida.

La principal ventaja de RK45 radica en que proporciona un equilibrio adecuado entre precisión y eficiencia para problemas no rígidos. Su capacidad de adaptar el paso temporal permite refinar la integración en regiones donde la solución cambia rápidamente y ampliarlo cuando la evolución es más suave, optimizando así el costo computacional.

En el contexto del reactor batch isotérmico, RK45 resulta especialmente útil para resolver problemas con cinéticas de ley de potencia y perfiles temporales regulares. También puede emplearse en ciertos casos no isotérmicos moderados; sin embargo, cuando el acoplamiento térmico se intensifica o la sensibilidad con respecto a la temperatura es elevada, pueden presentarse dificultades de estabilidad que hacen preferible el uso de métodos implícitos.

3.9.5. Métodos Implícitos Radau y BDF

Para sistemas con comportamiento más sensible o potencialmente rígido, los métodos implícitos constituyen una alternativa más robusta. Entre ellos destacan los métodos Radau y BDF, ambos ampliamente utilizados en la resolución de ecuaciones diferenciales ordinarias en problemas de ingeniería.

El método Radau pertenece a la familia de métodos implícitos de Runge-Kutta y se caracteriza por ofrecer buena estabilidad en problemas no lineales y rígidos. Por su parte, el método BDF o *Backward Differentiation Formula* se basa en fórmulas de diferenciación hacia atrás y es particularmente adecuado para sistemas donde la rigidez numérica obliga a utilizar esquemas estables con pasos relativamente grandes.

En el análisis de reactores batch no isotérmicos, estos métodos resultan especialmente valiosos porque permiten integrar sistemas donde pequeñas variaciones térmicas generan cambios significativos en la velocidad de reacción. Su uso reduce el riesgo de inestabilidades numéricas y mejora la confiabilidad de la solución cuando el sistema presenta acoplamiento fuerte entre balances de materia y energía.

3.9.6. Rigidez del Sistema y Criterio de Selección del Método Numérico

La rigidez es una propiedad numérica que aparece cuando el sistema contiene escalas de variación muy diferentes entre sí, de modo que algunos componentes cambian lentamente mientras otros lo hacen con rapidez. En tales condiciones, un método explícito puede requerir pasos extremadamente pequeños para conservar la estabilidad, incluso si la solución física evoluciona de forma relativamente suave.

En reactores batch no isotérmicos, la rigidez puede presentarse cuando existe una fuerte dependencia de la velocidad de reacción con la temperatura, especialmente en reacciones exotérmicas con energía de activación elevada o en sistemas con intercambio térmico significativo. En estos casos, el acoplamiento entre cinética y energía puede producir respuestas rápidas en ciertos intervalos de tiempo, dificultando la integración con métodos explícitos.

Por esta razón, el criterio de selección del método numérico debe considerar la naturaleza del problema. Para sistemas isotérmicos o no rígidos, un método explícito como RK45 puede ser suficiente. En cambio, para sistemas no isotérmicos con mayor complejidad numérica, resulta más conveniente emplear métodos implícitos como Radau o BDF. Esta distinción no solo mejora la estabilidad del cálculo, sino que también contribuye a la obtención de resultados físicamente coherentes.

3.9.7. Tolerancias, Estabilidad y Convergencia Numérica

La calidad de una solución numérica no depende únicamente del método empleado, sino también de los criterios de control de error utilizados durante la integración. En la práctica, la solución aproximada se regula mediante tolerancias numéricas que limitan el error permitido en cada paso de cálculo. Entre las más comunes se encuentran la tolerancia relativa y la tolerancia absoluta, las cuales determinan la precisión con que se sigue la evolución de las variables del sistema.

La estabilidad numérica se refiere a la capacidad del método para producir soluciones acotadas y físicamente razonables a lo largo del intervalo de integración. La convergencia, por su parte, implica que la solución aproximada tienda a la solución exacta a medida que se refina el proceso de integración. En el análisis de reactores batch, ambos aspectos son esenciales, ya que una solución numéricamente inestable o no convergente puede conducir a interpretaciones incorrectas del comportamiento del sistema.

En este contexto, el control adecuado de tolerancias y la elección del método de integración constituyen elementos decisivos para asegurar la confiabilidad del modelo. En aplicaciones de simulación, estos criterios permiten obtener perfiles

temporales consistentes de concentración y temperatura, comparar soluciones entre distintos métodos y validar el comportamiento del simulador frente a casos de referencia.

3.10. Variables de Desempeño y Criterios de Evaluación del Reactor

El análisis de un reactor batch ideal no se limita a la formulación de balances de materia y energía, sino que requiere también la definición de variables de desempeño que permitan interpretar el comportamiento del sistema desde una perspectiva técnica. Estas variables facilitan la evaluación del grado de transformación química, la evolución térmica del reactor y la conveniencia operativa de mantener la reacción durante un determinado intervalo de tiempo.

En el contexto del presente trabajo, las principales variables de desempeño consideradas son la concentración de las especies, la conversión, la especie de referencia, el reactivo limitante, la temperatura del reactor, la temperatura máxima permisible y el tiempo recomendado de operación o parada. En conjunto, estas magnitudes permiten caracterizar la respuesta dinámica del sistema y constituyen la base para la interpretación de los resultados obtenidos mediante simulación.

3.10.1. Concentración y Perfiles Temporales de Especie

La concentración de cada especie química representa una de las variables importantes en el análisis de reactores batch, dado que describe la cantidad de materia presente por unidad de volumen en cada instante de tiempo. En un reactor batch ideal homogéneo, estas concentraciones evolucionan únicamente con el tiempo, como consecuencia del avance de la reacción y de la estequiometría del sistema.

Matemáticamente, el comportamiento temporal de cada especie puede representarse mediante funciones del tipo:

$$C_i = C_i(t)$$

Donde C_i es la concentración de la especie i y t es el tiempo. Los perfiles temporales de concentración permiten visualizar cómo disminuyen los reactivos y cómo aumentan los productos a medida que avanza la reacción. Desde el punto de vista del diseño y la simulación, estos perfiles constituyen una herramienta de gran utilidad para analizar la velocidad de transformación, comparar escenarios de operación y detectar cambios significativos en la dinámica del sistema.

3.10.2. Conversión y Especie de Referencia

La conversión es una variable de desempeño ampliamente utilizada en Ingeniería de las Reacciones Químicas, ya que mide la fracción de un reactivo que ha sido transformada en el transcurso de la reacción. Para una especie de referencia A, la conversión puede definirse como:

$$X_A = \frac{N_{A0} - N_A}{N_{A0}}$$

O, para volumen constante, de manera equivalente:

$$X_A = \frac{C_{A0} - C_A}{C_{A0}}$$

Donde N_{A0} y C_{A0} representan la cantidad y la concentración iniciales de la especie de referencia, respectivamente.

La elección de una especie de referencia resulta necesaria para cuantificar el avance global del proceso. En la mayoría de los casos, esta especie corresponde a un reactivo, ya que la conversión se interpreta de manera más natural como la fracción consumida de dicho componente. En términos de diseño, la conversión permite relacionar el comportamiento del reactor con el tiempo de operación y constituye una de las variables más utilizadas para establecer criterios de evaluación del sistema.

3.10.3. Reactivo Limitante

El reactivo limitante es aquel que, de acuerdo con la relación estequiométrica de la reacción, se consume primero y determina el máximo avance teórico posible del proceso. Su identificación es importante porque permite seleccionar una base coherente para el análisis de la conversión y establecer una referencia adecuada para la interpretación de los resultados.

Para una reacción general, el reactivo limitante puede identificarse comparando la relación entre la cantidad inicial de cada reactivo y el valor absoluto de su coeficiente estequiométrico. De forma general, para cada reactivo i se analiza la magnitud:

$$\frac{N_{i0}}{|\nu_i|}$$

Y el menor de estos valores corresponde al reactivo limitante. Desde el punto de vista práctico, esta especie suele emplearse como referencia por defecto en el análisis del reactor, ya que su comportamiento refleja de manera directa el grado de aprovechamiento estequiométrico del sistema.

3.10.4. Evolución de la Temperatura del Reactor

En los reactores batch no isotérmicos, la temperatura constituye una variable de estado esencial, debido a que influye directamente sobre la velocidad de reacción y, al mismo tiempo, es afectada por el calor generado o absorbido por la reacción y por el intercambio térmico con el entorno. Su evolución temporal puede representarse mediante:

$$T = T(t)$$

El análisis del perfil térmico del reactor permite identificar si el sistema experimenta incrementos o disminuciones de temperatura durante la operación, así como evaluar la intensidad del acoplamiento entre el balance de energía y la cinética química. Desde una perspectiva técnica, el seguimiento de $T(t)$ resulta indispensable para interpretar el comportamiento global del reactor, especialmente

en reacciones exotérmicas o endotérmicas donde la respuesta térmica condiciona el tiempo de reacción y la conversión alcanzada.

3.10.5. Temperatura Máxima Permisible

La temperatura máxima permisible constituye un criterio de evaluación asociado a las restricciones operativas del reactor. Esta magnitud representa un valor límite de temperatura que no debe superarse durante la operación, ya sea por razones de seguridad, estabilidad del producto, integridad del equipo o coherencia con las condiciones de diseño.

Si se designa este límite como T_{MP}, el criterio de aceptación térmica puede expresarse como:

$$T(t) < T_{MP}\ para\ todo\ t$$

Cuando la temperatura calculada del reactor supera este umbral en algún instante del proceso, se interpreta que el sistema opera fuera de la condición térmica permisible establecida. En consecuencia, la temperatura máxima permisible no solo actúa como una referencia de control, sino también como un criterio para evaluar si un escenario de operación resulta adecuado o no desde el punto de vista técnico.

3.10.6. Tiempo Recomendado de Operación o Parada

En un reactor batch, el tiempo de operación representa una variable de diseño central. Sin embargo, desde una perspectiva práctica, no siempre resulta conveniente prolongar la reacción hasta el tiempo máximo fijado de manera arbitraria, especialmente si las variaciones posteriores de concentración, conversión o temperatura se vuelven muy pequeñas y el sistema se aproxima a un estado de cambio reducido.

Bajo esta idea, puede definirse un tiempo recomendado de operación o parada como el instante a partir del cual el sistema presenta cambios suficientemente pequeños en sus variables de interés, de manera que continuar la reacción produce beneficios marginales desde el punto de vista del desempeño global. Este criterio puede establecerse con base en la disminución progresiva de la velocidad de cambio de la conversión, de las concentraciones o de la temperatura.

Desde el punto de vista de simulación, esta variable no reemplaza a la solución del modelo, sino que actúa como un indicador adicional para interpretar los resultados y sugerir un intervalo operativo razonable. Su utilidad radica en que permite relacionar la evolución matemática del sistema con una decisión técnica sobre la conveniencia de mantener o detener el proceso.

3.10.7. Interpretación de los Resultados de Simulación

La interpretación de los resultados de simulación debe realizarse de forma integrada, considerando simultáneamente la evolución de las concentraciones, la conversión, la temperatura y los criterios de aceptación definidos para el sistema. Una lectura aislada de una sola variable puede conducir a conclusiones

incompletas; por ello, el análisis del reactor batch requiere examinar la relación entre el avance químico, la respuesta térmica y las restricciones operativas del proceso.

En este sentido, los perfiles temporales de especie permiten identificar el ritmo de consumo y formación de componentes, mientras que la conversión cuantifica el grado global de transformación alcanzado. Por su parte, la evolución de la temperatura aporta información sobre la influencia del efecto térmico de la reacción y sobre la estabilidad de la operación. Finalmente, criterios como la temperatura máxima permisible y el tiempo recomendado de operación permiten trasladar la información del modelo a una evaluación más cercana a la toma de decisiones técnicas.

Por tanto, las variables de desempeño no deben entenderse únicamente como salidas numéricas del simulador, sino como herramientas para valorar la factibilidad, conveniencia y coherencia física de un caso de operación determinado. Su análisis conjunto constituye un paso esencial para vincular el modelo matemático con la interpretación técnica del comportamiento del reactor.

3.11. Simulación Computacional de Reactores Batch

La simulación computacional de reactores batch constituye una herramienta de gran importancia en el análisis técnico de sistemas reaccionantes, ya que permite representar numéricamente la evolución temporal de las variables de interés a partir de modelos matemáticos fundamentados en balances de materia, balances de energía y expresiones cinéticas. En el caso de reactores isotérmicos y no isotérmicos, esta aproximación resulta especialmente útil cuando el comportamiento del sistema no puede describirse de manera suficiente mediante soluciones analíticas cerradas o cuando se desea explorar distintos escenarios de operación de forma sistemática.

El desarrollo de herramientas de simulación para reactores químicos responde a la necesidad de integrar, en un mismo entorno, la formulación teórica del problema, la estrategia de solución numérica y la interpretación de los resultados. En este sentido, la simulación no debe entenderse únicamente como un medio para obtener respuestas numéricas, sino como un recurso que permite visualizar el comportamiento dinámico del reactor, evaluar la influencia de parámetros cinéticos y térmicos, y facilitar la comprensión del fenómeno reaccionante desde una perspectiva técnica.

3.11.1. Papel de la Simulación en el Análisis Técnico de Reactores

En Ingeniería de las Reacciones Químicas, la simulación computacional cumple un papel relevante porque permite estudiar el comportamiento del reactor bajo condiciones de operación específicas sin recurrir exclusivamente a ensayos experimentales o a desarrollos analíticos altamente restrictivos. A través de la simulación es posible estimar perfiles temporales de concentración, conversión y temperatura, analizar la sensibilidad del sistema frente a parámetros del modelo y examinar la respuesta dinámica del reactor ante diferentes condiciones iniciales y térmicas.

Esta capacidad resulta especialmente valiosa en reactores batch, donde el tiempo constituye la variable principal de análisis y donde el comportamiento del sistema puede cambiar significativamente a lo largo de la operación. En sistemas no isotérmicos, además, la simulación permite estudiar el efecto del acoplamiento entre cinética y energía, identificar comportamientos térmicos relevantes y evaluar si el reactor opera dentro de condiciones aceptables desde el punto de vista técnico.

Por ello, la simulación constituye un medio adecuado para traducir el modelo matemático en una representación dinámica del proceso, facilitando tanto el análisis del desempeño del reactor como la interpretación de los resultados obtenidos.

3.11.2. Relación entre el Modelo Matemático, Método Numérico y Software

La simulación de un reactor batch se apoya en tres niveles conceptuales estrechamente relacionados: el modelo matemático, el método numérico y el software que integra ambos componentes dentro de un entorno operativo.

- **Primer nivel:** corresponde al modelo matemático, formado por los balances de materia y energía, la estequiometría de la reacción y la ley cinética adoptada. Este nivel define la estructura físico-matemática del problema y establece qué variables deben resolverse y bajo qué supuestos opera el sistema.
- **Segundo nivel:** corresponde al método numérico, que proporciona la estrategia necesaria para aproximar la solución del sistema de ecuaciones diferenciales ordinarias. Dado que en muchos casos el problema no admite solución analítica, la elección del integrador, de las tolerancias y de la estrategia de resolución constituye un elemento decisivo para obtener resultados estables y físicamente coherentes.
- **Tercer nivel:** corresponde al software, entendido como la implementación computacional que permite organizar los datos de entrada, ejecutar la simulación, controlar la resolución numérica y presentar los resultados de manera interpretable. En este nivel, la calidad de la herramienta no depende solo de que resuelva correctamente el sistema, sino también de que conserve trazabilidad entre las ecuaciones del modelo, el procedimiento numérico aplicado y las magnitudes que se muestran al usuario.

En consecuencia, la simulación computacional de reactores batch puede entenderse como la integración operativa de una base teórica, una estrategia numérica y una estructura de software, cuyo propósito común es representar de forma consistente el comportamiento del reactor.

3.11.3. Ventajas de las Herramientas Abiertas y Trazables en Ingeniería Química

En el ámbito actual de la simulación de procesos químicos existen plataformas comerciales de amplio alcance, capaces de representar procesos complejos en contextos industriales. No obstante, muchas de estas herramientas operan bajo esquemas altamente automatizados y modelos cerrados, lo que puede dificultar la

comprensión explícita de la relación entre las ecuaciones del problema, el método de solución empleado y los resultados obtenidos.

Frente a ello, las herramientas abiertas y trazables ofrecen una ventaja significativa, particularmente cuando se requiere claridad en la formulación del modelo y transparencia en el proceso de cálculo. Una herramienta abierta permite examinar cómo se construye el problema, qué supuestos se adoptan, cómo se implementan los balances y qué método numérico se emplea para resolver el sistema. Esta trazabilidad favorece una interpretación técnica más rigurosa y facilita la identificación de errores, limitaciones o posibilidades de ampliación del modelo.

Además, el carácter abierto de una herramienta de simulación permite su mejora progresiva, la incorporación de nuevos modelos y la adaptación a distintos casos de estudio. En este sentido, el uso de software abierto no solo responde a un criterio de accesibilidad, sino también a una lógica metodológica orientada a la transparencia, la reproducibilidad y la evolución técnica del sistema desarrollado.

3.11.4. Cantera como Herramienta Abierta de Simulación Química

Cantera es una suite de herramientas de código abierto orientada al tratamiento computacional de problemas que involucran cinética química, termodinámica y procesos de transporte. Su propósito principal es facilitar el uso de modelos termoquímicos y cinéticos en aplicaciones científicas y de ingeniería, permitiendo el cálculo de propiedades, velocidades de reacción, composición de especies y evolución de sistemas reaccionantes. De acuerdo con Cantera Developers (s. f.), la herramienta está diseñada para problemas relacionados con cinética química, termodinámica y transporte, y puede utilizarse desde distintos lenguajes de programación, entre ellos Python, MATLAB, C/C++ y Fortran.

Desde el punto de vista computacional, Cantera utiliza una estructura orientada a objetos que permite representar fases, especies químicas, mecanismos de reacción, propiedades termodinámicas y modelos de transporte. Esta organización facilita la construcción de sistemas reaccionantes a partir de archivos de entrada que contienen información sobre especies, reacciones, parámetros cinéticos y modelos termodinámicos. En consecuencia, Cantera se emplea como una biblioteca científica capaz de integrarse en programas externos o scripts de simulación, especialmente cuando se requiere resolver problemas con cinética química detallada.

En el análisis de reactores, Cantera proporciona herramientas para construir y resolver redes de reactores. Su documentación incluye ejemplos relacionados con reactores de tanque agitado, simulaciones adiabáticas, reactores con transferencia de calor, reactores separados por paredes móviles, reactores de flujo pistón modelados mediante cadenas de reactores y otros sistemas reaccionantes dependientes del tiempo. Estos ejemplos muestran la utilidad de Cantera como entorno de simulación para estudiar la evolución temporal de temperatura, presión, composición y otras variables asociadas a sistemas químicos reaccionantes.

La disponibilidad de Cantera desde Python resulta especialmente relevante en el contexto de la simulación computacional, ya que permite combinar sus capacidades

termoquímicas y cinéticas con bibliotecas científicas del ecosistema Python. Esta característica facilita la elaboración de scripts reproducibles, el análisis de resultados, la generación de gráficas y la integración con herramientas numéricas adicionales. Por ello, Cantera constituye un referente importante dentro de las herramientas abiertas empleadas para el modelado y la simulación de sistemas reaccionantes en Ingeniería Química.

3.12. Síntesis del Modelo Teórico y Transición a la Metodología

El marco teórico desarrollado en este capítulo ha permitido establecer los fundamentos conceptuales y matemáticos necesarios para el análisis de reactores batch ideales homogéneos, tanto isotérmicos como no isotérmicos. En primer lugar, se abordó la Ingeniería de las Reacciones Químicas como disciplina base para la formulación de modelos de reactores, así como la clasificación general de los reactores químicos y la ubicación del reactor batch dentro de dichos sistemas ideales.

Posteriormente, se describieron las características físicas y operativas del reactor batch ideal homogéneo, junto con los supuestos que delimitan el alcance del modelo adoptado en este trabajo. A partir de ello, se desarrollaron los balances de materia y energía, la ecuación de diseño del reactor batch y el acoplamiento entre variables químicas y térmicas, mostrando que el comportamiento dinámico del sistema puede representarse mediante un conjunto de ecuaciones diferenciales ordinarias dependientes del tiempo.

Asimismo, se abordó la formulación cinética de la reacción mediante leyes de velocidad y la ecuación de Arrhenius, estableciendo la relación entre concentración, temperatura y velocidad de reacción. Finalmente, se analizaron los métodos numéricos requeridos para resolver el modelo, así como las variables de desempeño y los criterios de evaluación que permiten interpretar técnicamente los resultados de simulación.

En conjunto, estos elementos conforman la base teórica del simulador desarrollado en el presente trabajo. La transición hacia la metodología se sustenta precisamente en la necesidad de traducir este modelo teórico a una estructura computacional organizada, capaz de recibir datos de entrada, resolver numéricamente el sistema planteado y presentar resultados consistentes con la formulación físico-matemática del reactor. Por ello, el capítulo siguiente se orienta a describir el enfoque metodológico, la estrategia de modelado, la resolución numérica y la implementación del software desarrollado para la simulación de reactores batch ideales homogéneos.

IV. Metodología

4.1. Enfoque Metodológico de la Investigación

La presente investigación se desarrolló bajo un enfoque de investigación aplicada con orientación al desarrollo tecnológico, centrado en el diseño e implementación de una herramienta computacional para la simulación de reactores batch ideales homogéneos, tanto en régimen isotérmico como no isotérmico. El trabajo se fundamentó en la formulación, resolución e integración de modelos matemáticos clásicos de la Ingeniería de las Reacciones Químicas, particularmente balances de materia, balances de energía y leyes cinéticas empleadas en el análisis de reactores batch, dentro de un entorno de software capaz de representar de manera clara y reproducible el comportamiento dinámico del sistema (Smith, 1981; Levenspiel, 1999; Fogler, 2016).

Desde el punto de vista metodológico, el estudio tuvo una naturaleza cuantitativa, determinística y no experimental. Fue cuantitativo porque se apoyó en relaciones matemáticas que describen de manera explícita la evolución temporal de las variables del sistema; determinístico porque el comportamiento del reactor quedó completamente definido por las ecuaciones del modelo, los parámetros cinéticos y térmicos, y las condiciones iniciales de operación; y no experimental porque no contempló la realización de ensayos físicos de laboratorio ni la obtención de datos experimentales propios. En su lugar, el desarrollo se sustentó en modelos teóricos ampliamente establecidos en la literatura especializada, tales como balances de materia, balances de energía y ecuaciones cinéticas clásicas (Smith, 1981; Levenspiel, 1999; Fogler, 2016).

Bajo este enfoque, el problema de estudio se abordó como un proceso de traducción de fundamentos físico-matemáticos a una estructura computacional funcional, siguiendo el principio de correspondencia entre modelo de reactor, ecuaciones gobernantes, método numérico y resultados simulados, tal como se plantea en el análisis computacional de reactores químicos y en la resolución numérica de ecuaciones diferenciales ordinarias (Fogler, 2016; Virtanen et al., 2020).

Cabe destacar que la validación del modelo matemático y del software desarrollado no se realizó mediante contrastación experimental directa, sino a través de la comparación con soluciones teóricas, casos de referencia y problemas representativos reportados en la literatura clásica de la Ingeniería de las Reacciones Químicas. Este proceso se complementó con criterios de coherencia física, estabilidad y convergencia numérica, con el propósito de verificar que la herramienta produce resultados consistentes con el marco teórico adoptado.

De manera general, el proceso metodológico seguido en este trabajo comprendió las siguientes etapas:

1. **Formulación del modelo teórico**, a partir de los principios de conservación de la materia y la energía aplicados a un reactor batch ideal homogéneo, bajo un planteamiento general válido para regímenes isotérmicos y no isotérmicos.

2. **Definición del esquema computacional del problema**, mediante la identificación de variables de estado, parámetros cinéticos y térmicos, condiciones iniciales, restricciones del modelo y sistema de unidades.
3. **Resolución numérica del sistema**, empleando métodos de integración apropiados para ecuaciones diferenciales ordinarias, con atención a la estabilidad, el control del error y la naturaleza potencialmente no lineal o rígida del problema.
4. **Desarrollo e integración del software**, incorporando el modelo matemático, el motor numérico, la validación de entradas, la gestión de casos y la interfaz gráfica del simulador.
5. **Validación del simulador**, mediante la reproducción de problemas de referencia, la comparación con resultados teóricos reportados en la literatura y la verificación de consistencia numérica y físico-matemática.

En términos operativos, la metodología se organizó de acuerdo con los objetivos específicos del trabajo. Primero, se formularon los modelos matemáticos del reactor batch ideal homogéneo a partir de los balances de materia y energía, definiendo las variables de estado, parámetros cinéticos, parámetros térmicos y condiciones iniciales del sistema. Luego, dichos modelos se transformaron en una estructura computacional mediante la construcción de una función dinámica capaz de calcular las derivadas de concentración y temperatura en cada instante de integración. Posteriormente, esta función fue resuelta numéricamente con integradores de ecuaciones diferenciales ordinarias disponibles en Python, y sus resultados fueron conectados con una interfaz gráfica que permite ingresar datos, ejecutar simulaciones, visualizar perfiles y comparar resultados con casos de referencia.

4.2. Modelado Matemático del Reactor Batch Ideal (Modelo Isotérmico y no Isotérmico)

El modelado matemático desarrollado en este trabajo tuvo como finalidad representar de forma coherente los mecanismos que gobiernan la evolución temporal de las concentraciones y de la temperatura en un reactor batch ideal homogéneo. No se buscó reproducir con exactitud todos los fenómenos presentes en un reactor real, sino construir una formulación clara, consistente y computacionalmente tratable, que permitiera analizar el comportamiento dinámico del sistema bajo condiciones isotérmicas y no isotérmicas.

La formulación implementada en el simulador se basa en balances de materia y energía, en una ley cinética de tipo potencia y, cuando corresponde, en la dependencia térmica de la constante cinética mediante la ecuación de Arrhenius, fundamentos ampliamente utilizados en el modelado de reactores batch ideales isotérmicos y no isotérmicos (Smith, 1981; Levenspiel, 1999; Fogler, 2016).

4.2.1. Definición del Sistema y Supuestos del Modelo

El sistema bajo estudio correspondió a un reactor batch ideal homogéneo, en el cual tiene lugar una sola reacción química irreversible en fase líquida, dentro de un volumen cerrado y perfectamente mezclado. Esta representación corresponde al modelo clásico de reactor batch ideal, en el que no existen corrientes de entrada ni

salida durante la reacción y las variables del sistema dependen únicamente del tiempo (Levenspiel, 1999; Fogler, 2016).

- El reactor opera como un sistema cerrado respecto a la materia, sin flujo de entrada ni salida durante el tiempo de reacción.
- El sistema presenta mezcla perfecta, por lo que las concentraciones y la temperatura son uniformes en todo el volumen del reactor en cada instante.
- La reacción química considerada es única e irreversible.
- El sistema es homogéneo y de fase líquida.
- El volumen del reactor permanece constante durante todo el proceso.
- En la formulación no isotérmica, la temperatura puede variar con el tiempo.
- Las propiedades térmicas globales, como la densidad y la capacidad calorífica, se consideran constantes dentro del intervalo de operación analizado.
- Cuando existe intercambio térmico con el entorno, este se representa mediante una temperatura de chaqueta constante y un término global de transferencia de calor.

Estos supuestos permitieron simplificar el análisis matemático sin perder la capacidad de describir los fenómenos principales que gobiernan la dinámica del reactor. La adopción de mezcla perfecta, volumen constante, sistema homogéneo y reacción única irreversible es coherente con las formulaciones clásicas empleadas para el análisis de reactores batch ideales en la literatura de Ingeniería de las Reacciones Químicas (Smith, 1981; Levenspiel, 1999; Fogler, 2016).

4.2.2. Variables del Modelo

El modelo matemático se expresa en función de un conjunto de variables dependientes y de una variable independiente. La variable independiente es el tiempo, mientras que las variables dependientes corresponden a las concentraciones de las especies químicas presentes y, en la formulación no isotérmica, a la temperatura del reactor.

Tabla 2.
Variables del modelo matemático del reactor batch

Variable	***Descripción***	***Unidades***
t	Tiempo	s
$C_i(t)$	Concentración de la especie i	$\frac{mol}{m^3}$
$T(t)$	Temperatura del reactor	K

Nota. Elaboración propia.

En el caso isotérmico, el estado del sistema se representa mediante el vector de concentraciones de las especies químicas. En cambio, para el caso no isotérmico, el estado del sistema incorpora adicionalmente la temperatura del reactor. Estas

variables describen el estado dinámico del sistema y constituyen las incógnitas principales que se determinan mediante la integración numérica del modelo.

En todas las ecuaciones del simulador, la concentración se expresa internamente en mol/m^3 y el volumen en m^3. Cuando el usuario introduce datos en unidades diferentes, como mol/L, el sistema realiza una conversión previa al Sistema Internacional antes de resolver el problema, siguiendo el criterio de coherencia dimensional recomendado para el uso técnico del SI en cálculos científicos e ingenieriles (National Institute of Standards and Technology, 2008).

4.2.3. Balance de Materia para el Reactor Batch

Para una especie química i presente en el reactor batch, el balance de materia se fundamenta en la Ecuación (3.1), derivada del principio de conservación de la materia aplicado a un sistema cerrado y perfectamente mezclado, tal como se plantea en la formulación clásica de reactores batch ideales (Smith, 1981; Levenspiel, 1999; Fogler, 2016).

$$r_i = \nu_i r$$

Por tanto, el balance de materia implementado para cada especie puede escribirse como:

$$\frac{dC_i}{dt} = \nu_i r$$

Esta formulación se integra posteriormente dentro del sistema acoplado general definido en la Ecuación (3.9), constituyendo una de las bases del modelo dinámico implementado en el simulador. La relación entre velocidad global de reacción, coeficientes estequiométricos y velocidad de formación o consumo de cada especie corresponde al tratamiento convencional de la estequiometría en modelos cinéticos de reactores químicos (Smith, 1981; Fogler, 2016).

4.2.4. Dependencia de la Cinética con la Temperatura

En la formulación no isotérmica, la constante cinética se modela mediante la ecuación de Arrhenius, definida previamente en la Ecuación (3.10), la cual constituye una relación fundamental para representar la dependencia de la velocidad de reacción con la temperatura en sistemas reaccionantes (Smith, 1981; Fogler, 2016).

Esta relación introduce de manera explícita la influencia de la temperatura sobre la velocidad de reacción. Como consecuencia, la cinética deja de depender únicamente de la concentración y pasa a estar controlada también por la evolución térmica del sistema. Este hecho es el origen del acoplamiento entre el balance de materia y el balance de energía y contribuye al carácter no lineal del sistema de ecuaciones diferenciales ordinarias (Fogler, 2016).

4.2.5. Balance de Energía del Reactor Batch (Formulación no Isotérmica)

El balance de energía implementado corresponde a la formulación no isotérmica desarrollada en la Ecuación (3.7), obtenida a partir de la primera ley de la

termodinámica aplicada a un reactor batch ideal con propiedades térmicas globales constantes y posible intercambio de calor con el entorno (Bird et al., 2007; Incropera et al., 2007; Fogler, 2016). Esta ecuación describe la evolución temporal de la temperatura del reactor como resultado del efecto térmico de la reacción y del posible intercambio de calor con el entorno. Junto con los balances de materia, completa el sistema dinámico acoplado definido en la Ecuación (3.9).

Para el caso isotérmico, el modelo se obtiene imponiendo $dT/dt = 0$, por lo que no se resuelve una ecuación diferencial adicional para $T(t)$. Esta reducción es consistente con el tratamiento clásico de reactores batch isotérmicos, en los cuales la temperatura se considera constante y la dinámica del sistema se describe únicamente mediante los balances de materia y la ley cinética correspondiente (Levenspiel, 1999; Fogler, 2016).

4.2.6. Condiciones Iniciales

La solución del sistema de ecuaciones diferenciales ordinarias requiere la especificación de condiciones iniciales, las cuales definen el estado del sistema en el instante inicial $t = t_0$. Estas condiciones se incorporan formalmente en el problema de valor inicial definido en la Ecuación (4.1), mediante el vector $y(t_0) = y_0$.

En dicho vector se incluyen las concentraciones iniciales C_{i0}y, para los casos no isotérmicos, la temperatura inicial T_0. Estas condiciones constituyen una parte esencial de los datos de entrada del simulador, ya que determinan el punto de partida de la integración numérica y condicionan toda la trayectoria temporal del sistema.

Una vez definidas las variables, parámetros, ecuaciones gobernantes y condiciones iniciales, el modelo matemático fue trasladado a código mediante una representación estructurada del caso de simulación. Las concentraciones y, cuando corresponde, la temperatura, se organizaron en un vector de estado; los parámetros cinéticos, térmicos y operativos se almacenaron como datos de entrada del caso; y las ecuaciones diferenciales se implementaron como una función evaluable por el integrador numérico. De esta manera, la formulación teórica del reactor se convirtió en una rutina computacional capaz de calcular la evolución temporal del sistema.

4.3. Estrategia de Resolución Numérica

La formulación del reactor batch ideal homogéneo, en sus versiones isotérmica y no isotérmica, conduce a un sistema de ecuaciones diferenciales ordinarias acopladas y, en general, no lineales. Esta naturaleza surge del acoplamiento entre los balances de materia y energía, así como de la dependencia de la velocidad de reacción con la temperatura mediante la ecuación de Arrhenius. Debido a esta no linealidad, la obtención de soluciones analíticas se restringe a casos altamente simplificados, lo que hace necesario el empleo de métodos numéricos de integración temporal para determinar la evolución de las concentraciones y de la temperatura del sistema (Smith, 1981; Levenspiel, 1999; Fogler, 2016; Hairer et al., 1993; Ascher y Petzold, 1998).

En este contexto, la estrategia de resolución numérica adoptada en el presente trabajo tuvo como propósito garantizar resultados estables, precisos y reproducibles, adecuados para el análisis técnico del reactor y para la verificación de problemas representativos del diseño de reactores químicos. Para ello, el problema se formuló como un problema de valor inicial, se seleccionaron métodos de integración acordes con la posible rigidez del sistema y se establecieron criterios de control del error, discretización temporal y validación interna.

4.3.1. Planteamiento del Problema Numérico (IVP)

La simulación del reactor batch se formula como un problema de valor inicial (Initial Value Problem, IVP), dado que el estado del sistema queda completamente determinado a partir de un conjunto de condiciones iniciales especificadas en un tiempo inicial conocido. Este tipo de planteamiento es el adecuado para sistemas dinámicos gobernados por ecuaciones diferenciales ordinarias, en los cuales la solución se obtiene a partir de un estado inicial y de una función que describe la evolución temporal del sistema (Hairer et al., 1993; Ascher y Petzold, 1998).

De forma general, el problema numérico se expresa mediante la Ecuación (4.1):

$$\frac{dy}{dt} = f(t, y, \theta), \quad y(t_0) = y_0 \tag{4.1}$$

Donde y representa el vector de estado y θ el conjunto de parámetros del modelo, incluyendo parámetros cinéticos, térmicos y geométricos del reactor. Esta notación corresponde a la forma general de un sistema de ecuaciones diferenciales ordinarias de primer orden formulado como problema de valor inicial (Hairer et al., 1993; Ascher y Petzold, 1998).

En el caso isotérmico, el vector de estado contiene únicamente las concentraciones de las especies químicas, como se muestra en la Ecuación (4.2):

$$y(t) = [C_1(t), C_2(t), \dots, C_m(t)]^T \tag{4.2}$$

Para el caso no isotérmico, el vector de estado incorpora además la temperatura del reactor, de acuerdo con la Ecuación (4.3):

$$y(t) = [C_1(t), C_2(t), \dots, C_m(t), T(t)]^T \tag{4.3}$$

En la implementación computacional, la función f evalúa simultáneamente:

1. Las velocidades de reacción y los términos estequiométricos asociados a los balances de materia.
2. El balance de energía del sistema cuando el modo seleccionado es no isotérmico.

De este modo, la estructura numérica del problema conserva correspondencia directa con la formulación físico-matemática desarrollada en el marco teórico.

4.3.2. Selección del Integrador y Criterio de Rigidez

En problemas de cinética química es frecuente la presencia de rigidez numérica (*stiffness*), especialmente cuando coexisten procesos con escalas de tiempo significativamente diferentes, como reacciones rápidas acopladas a variaciones térmicas más lentas, o cuando la velocidad de reacción presenta alta sensibilidad respecto a la temperatura. Este fenómeno ha sido ampliamente documentado en la literatura de métodos numéricos aplicados a sistemas dinámicos en ingeniería (Hairer, Nørsett y Wanner, 1993; Ascher y Petzold, 1998).

Por esta razón, la estrategia de resolución adoptada contempló métodos adecuados tanto para problemas no rígidos como para problemas rígidos. En la implementación computacional, el simulador emplea la función solve_ivp de la biblioteca científica SciPy, debido a que proporciona integradores explícitos e implícitos con control adaptativo del tamaño de paso y del error local, en concordancia con prácticas estándar para la resolución de ecuaciones diferenciales ordinarias (SciPy Developers, 2023; Virtanen et al., 2020).

En la implementación computacional, el simulador emplea la función solve_ivp de la biblioteca científica SciPy, debido a que proporciona integradores explícitos e implícitos con control adaptativo del tamaño de paso y del error local, en concordancia con prácticas estándar para la resolución de ecuaciones diferenciales ordinarias (SciPy Developers, 2023; Virtanen et al., 2020).

- **RK45**, como opción principal para problemas no rígidos o moderadamente acoplados.
- **Radau**, como alternativa implícita para sistemas con mayor sensibilidad numérica o posible rigidez.
- **BDF**, como método implícito orientado a problemas rígidos y a escenarios con fuerte acoplamiento entre cinética y energía.

En términos metodológicos, el criterio de selección se basó en la naturaleza del problema. Para casos isotérmicos o escenarios con comportamiento numéricamente regular, el método RK45 ofrece un equilibrio adecuado entre precisión y costo computacional. En cambio, cuando el sistema presenta fuerte dependencia térmica, cinética tipo Arrhenius con alta sensibilidad o dificultades de integración mediante métodos explícitos, se prioriza el uso de Radau o BDF por sus mejores propiedades de estabilidad.

Además, el simulador contempla una estrategia de selección automática del integrador, junto con la posibilidad de selección manual por parte del usuario. Esta decisión permite mantener flexibilidad operativa sin perder coherencia con la naturaleza numérica del problema resuelto.

4.3.3. Control del Error y Tolerancias Numéricas

Con el fin de garantizar que los resultados obtenidos mediante el simulador sean confiables y reproducibles, se empleó un esquema de control adaptativo del error local basado en tolerancias relativas y absolutas. Estas tolerancias se definen como:

- rtol: tolerancia relativa, que controla el error proporcional al valor de la variable.
- atol: tolerancia absoluta, que controla el error mínimo permitido cuando el valor de la variable es pequeño.

El integrador ajusta automáticamente el tamaño del paso temporal para mantener el error local dentro de los límites establecidos por dichas tolerancias. Este enfoque es ampliamente recomendado en la literatura de métodos numéricos para sistemas dinámicos no lineales, debido a que permite equilibrar precisión y eficiencia computacional (Hairer et al., 1993; Ascher y Petzold, 1998).

En el simulador, estas tolerancias forman parte de los parámetros configurables del caso y permiten adaptar la precisión de la integración al tipo de problema analizado. Su adecuada selección resulta especialmente importante en escenarios no isotérmicos, donde pequeñas variaciones de temperatura pueden producir cambios significativos en la velocidad de reacción y, por ende, en la dinámica general del sistema.

Tabla 3.
Criterio práctico de selección del solver y tolerancias

Condición observada en la integración	***Interpretación probable***	***Acción recomendada***
Integración estable, pasos moderados, sin advertencias	Problema no rígido	Utilizar RK45 con tolerancia estándar
Pasos muy pequeños, elevado número de evaluaciones o integración lenta	Posible rigidez o fuerte acoplamiento térmico-cinético	Utilizar Radau o BDF
Necesidad de comparación fina con casos de referencia	Mayor exigencia de precisión numérica	Ajustar rtol y atol a valores más estrictos

Nota. Elaboración propia.

4.3.4. Discretización Temporal y Gestión de Salidas

Aunque el integrador numérico opera internamente con un tamaño de paso variable, el simulador define explícitamente un intervalo de simulación $[t_0, t_f]$ y un conjunto de puntos de salida t_{eval}, en los cuales se reportan los resultados al usuario. Estos puntos de evaluación se distribuyen de manera uniforme dentro del intervalo total de simulación.

Esta estrategia permite diferenciar dos niveles del proceso numérico:

- **La integración interna**, realizada mediante pasos adaptativos que dependen del método seleccionado y del control del error.
- **La presentación de resultados**, realizada mediante puntos de salida consistentes para visualización, comparación y análisis.

La separación entre integración interna y presentación de resultados mejora la estabilidad del cálculo y, al mismo tiempo, facilita la generación de perfiles temporales comparables de concentración, conversión y temperatura. Desde el punto de vista metodológico, esto resulta especialmente útil para contrastar resultados entre distintos casos de estudio o frente a soluciones de referencia.

4.3.5. Consistencia Física y Validaciones Internas

Dado el carácter técnico del simulador, se incorporaron verificaciones internas destinadas a mejorar la robustez del programa y a reducir la posibilidad de interpretaciones erróneas derivadas de datos de entrada inconsistentes. Estas validaciones no sustituyen la validación formal del simulador, pero constituyen un control preliminar de calidad antes y durante la resolución numérica.

Entre las verificaciones consideradas se incluyen:

- Coherencia dimensional de parámetros principales como $U\ A$, ΔH_r, ρ y C_p.
- Validación de signos, por ejemplo, $\Delta H_r < 0$ para reacciones exotérmicas y $\Delta H_r > 0$ para reacciones endotérmicas.
- Prevención de estados no físicos, como la aparición de concentraciones negativas debido a tolerancias numéricas inadecuadas o a una mala parametrización del problema.
- Revisión de consistencia entre el modo de operación seleccionado y los parámetros requeridos por el modelo.

Estas validaciones internas contribuyen a que la solución numérica conserve significado físico y a que el simulador opere de forma segura y consistente con el alcance del modelo adoptado.

4.4. Diseño Algorítmico del Simulador, Casos de Uso, Parámetros de Entrada y Sistema de Unidades

La presente sección describe la lógica algorítmica con la que opera ReactSim Studio, el alcance funcional de los casos soportados, el sistema interno de unidades adoptado por el simulador y los parámetros mínimos requeridos para ejecutar una simulación. Su propósito fue mantener trazabilidad entre el modelo físico-matemático, la implementación computacional y los resultados generados por el software, de manera que el comportamiento del programa fuese consistente con los fundamentos de la Ingeniería de las Reacciones Químicas y con los casos de referencia utilizados para validación (Levenspiel, 1999; Fogler, 2016).

Desde esta perspectiva, el simulador no fue concebido como una calculadora genérica, sino como una herramienta estructurada por tipos de caso, con reglas explícitas de entrada, normalización dimensional, resolución numérica, validación interna y postproceso. Asimismo, el código fuente del sistema se dispuso en un repositorio público bajo licencia MIT, acompañado de documentación técnica, instrucciones de instalación y una descripción básica de la arquitectura del programa, con el fin de favorecer su trazabilidad, reproducibilidad y posibilidad de extensión futura.

4.4.1. Taxonomía de Casos Soportados por el Simulador (Alcance Funcional)

Con el fin de mantener coherencia entre teoría, metodología y software, ReactSim Studio organiza sus casos de simulación según el régimen térmico de operación. No obstante, la versión desarrollada en este trabajo delimita de forma explícita su alcance funcional a reactores batch ideales homogéneos, en fase líquida, de volumen constante y con una sola reacción química irreversible.

Bajo estas condiciones, el simulador soporta los siguientes tipos de caso:

- **Reactor batch isotérmico.** La temperatura del sistema se mantiene constante durante toda la simulación. En este modo se resuelven únicamente los balances de materia y la cinética puede formularse mediante ley de potencia o mediante una expresión cinética personalizada validada.
- **Reactor batch no isotérmico adiabático.** La temperatura evoluciona únicamente como consecuencia del calor liberado o absorbido por la reacción química, sin intercambio térmico con el entorno.
- **Reactor batch no isotérmico con chaqueta a temperatura constante.** El sistema incorpora un término de transferencia de calor del tipo $UA(T - T_c)$, donde T_c representa una temperatura externa constante durante la simulación.

Esta taxonomía responde al alcance real del modelo implementado y excluye, en la versión 1.0 del simulador, casos como sistemas gaseosos, volumen variable, presión variable, múltiples reacciones simultáneas, reacciones reversibles o temperatura de chaqueta dependiente del tiempo. De esta forma, la herramienta conserva consistencia entre el problema planteado y las capacidades efectivas del software.

4.4.2. Sistema Fijo de Unidades y Normalización de Entradas

Para evitar errores de consistencia dimensional durante la simulación numérica, el simulador adoptó un sistema fijo de unidades internas basado en el Sistema Internacional. Esta decisión resultó especialmente importante en la evaluación de la constante cinética dependiente de la temperatura, en los términos de transferencia de calor, en el calor de reacción y en las propiedades físicas del sistema.

Las unidades base empleadas corresponden a:

- Longitud: metro (m)
- Masa: kilogramo (kg)
- Tiempo: segundo (s)
- Temperatura: kelvin (K)
- Cantidad de sustancia: mol (mol)

A partir de estas magnitudes, el simulador utilizara internamente las convenciones resumidas en la **Tabla 4**.

Tabla 4.
Sistema interno de unidades adoptado por el simulador

Magnitud	*Símbolo*	*Unidad interna (SI)*	*Observaciones*
Tiempo	t	s	Variable independiente del sistema
Concentración	C_i	$\frac{mol}{m^3}$	Entradas en mol/L se convierten automáticamente
Temperatura	T	K	Las entradas pueden realizarse en K o °C y se convierten internamente a K
Volumen	V	m^3	Volumen constante en reactor batch
Velocidad de reacción	r_i	$\frac{mol}{m^3 s}$	Consistente con balances de materia
Constante cinética	k	Dependiente del orden	Ajustada a la forma cinética seleccionada
Factor preexponencial	k_0	Dependiente del modelo	Coherente con (k(T))
Energía	—	J	Unidad base para balances energéticos
Calor de reacción	ΔH_r	$\frac{J}{mol}$	Signo según convención exotérmica/endotérmica
Densidad	ρ	$\frac{kg}{m^3}$	Constante en el modelo base
Calor específico	C_p	$\frac{J}{kgK}$	Constante en el modelo base
Coeficiente de transferencia	U	$\frac{W}{m^2 K}$	Propiedad global
Área de transferencia	A	m^2	Geometría del reactor
Término de transferencia	UA	$\frac{W}{K}$	Usado directamente en el balance de energía
Constante de los gases	R	$8.314462618 \frac{J}{molK}$	Valor recomendado por CODATA

Nota. Elaboración propia.

Aunque el motor de cálculo opera exclusivamente en unidades del Sistema Internacional, la interfaz permite al usuario introducir datos en unidades de uso común en Ingeniería Química, tales como mol/L, L, min, h, °C y kJ/mol. Antes de iniciar la integración, todas las entradas son normalizadas automáticamente al

sistema interno. Este proceso reduce errores por inconsistencias dimensionales y fortalece la robustez numérica del simulador.

Adicionalmente, el módulo de entrada incorpora validaciones básicas de magnitud y compatibilidad entre parámetros, sin alterar la estructura del modelo matemático. Estas verificaciones actúan como controles preventivos para mejorar la confiabilidad del proceso de simulación.

4.4.3. Parámetros de Entrada por Tipo de Caso

Para evitar que el simulador opere como un formulario ambiguo o excesivamente genérico, se definió un contrato mínimo de entrada estructurado por bloques funcionales. Este contrato establece qué datos deben suministrarse para que cada caso pueda ensamblarse y resolverse de manera coherente.

Los bloques principales de entrada son los siguientes:

- **Datos generales del caso.** Incluyen el modo térmico, el intervalo temporal de simulación, el número de puntos de salida, la estrategia de selección del solver, las tolerancias numéricas y la especie de referencia para el cálculo de la conversión.
- **Definición de especies y estequiometría.** Se ingresan las especies participantes, sus coeficientes estequiométricos ν_i y sus concentraciones iniciales. En la versión actual del simulador, la formulación se realiza exclusivamente en términos de concentración.
- **Modelo cinético.** El usuario puede seleccionar una ley cinética de tipo potencia o ingresar una expresión cinética personalizada validada. En los casos isotérmicos puede emplearse una constante cinética fija o una formulación equivalente. En los casos no isotérmicos se utilizan k_0, E_a y la ecuación de Arrhenius para modelar la dependencia térmica de la constante cinética.
- **Propiedades térmicas.** Para los casos no isotérmicos se requiere de ρ, C_p y ΔH_r. En la versión 1.0 del simulador, tanto la densidad como el calor específico se consideran constantes, y su dependencia con la temperatura queda fuera del alcance actual.
- **Intercambio de calor con el entorno.** Cuando el caso corresponde a un reactor con chaqueta, se requieren U, A y T_c. En el modo adiabático, el intercambio térmico se anula imponiendo $UA = 0$.
- **Restricciones operativas opcionales.** El simulador permite definir una temperatura máxima permisible, la cual se utiliza como criterio de evaluación del caso una vez resuelto el modelo.

Esta estructura permitió que cada simulación se configure de forma clara y consistente con el régimen de operación seleccionado, evitando solicitudes de datos irrelevantes y reforzando la correspondencia entre el modelo teórico y la interfaz del software.

4.4.4. Algoritmo General del Simulador (Diseño y Proceso)

El funcionamiento de ReactSim Studio se organizó mediante una secuencia algorítmica modular y determinística, estructurada en bloques que reflejan la lógica completa del proceso de simulación. Esta secuencia fue diseñada para convertir los modelos formulados en la Sección 4.2 en un procedimiento computacional ejecutable: primero se reciben y validan los datos del caso, luego se normalizan las unidades, se construye el vector de estado, se define la función dinámica del reactor, se resuelve el problema de valor inicial y finalmente se generan las variables de salida para su interpretación técnica.

Bloque A. Configuración y validación inicial del caso.

El usuario selecciona el régimen térmico, define las especies, la cinética, los parámetros térmicos y el intervalo temporal. En esta etapa se verifican dominios físicos básicos, como $T > 0$, $C_i \geq 0$, coherencia de signos en ΔH_r y compatibilidad entre el modo térmico y los parámetros suministrados.

Bloque B. Normalización y ensamble del modelo.

Todas las entradas se convierten al Sistema Internacional y se construye el vector de estado $\mathrm{y}(t)$. Para el caso isotérmico se ensamblan únicamente los balances de materia, mientras que en el caso no isotérmico se incorpora también el balance de energía. En esta etapa se define la función $\mathrm{f}(t, \mathrm{y}, \theta)$ que es utilizada por el integrador numérico.

Bloque C. Resolución numérica.

El problema se plantea como un IVP de la forma $\dot{\mathrm{y}} = \mathrm{f}(t, \mathrm{y})$, con condición inicial $\mathrm{y}(t_0) = \mathrm{y}_0$. A continuación, se selecciona el integrador, de forma automática o manual, se fijan rtol, atol y los puntos de salida t_{eval}, y se ejecuta la integración temporal del sistema.

Bloque D. Postproceso y cálculo de variables derivadas.

Una vez resuelto el sistema, el simulador calcula variables derivadas de interés técnico, tales como conversión, especie de referencia, reactivo limitante, perfiles temporales de temperatura, velocidad de reacción, calor generado, calor removido cuando aplica, temperatura máxima alcanzada, verificación respecto a la temperatura máxima permisible y tiempo recomendado de operación o parada.

Bloque E. Presentación, persistencia y exportación de resultados.

Finalmente, el programa organiza la información en tablas, gráficas y mensajes de diagnóstico. Además, permite guardar o recuperar casos en formato .reactsim, así como exportar resultados y reportes para su análisis posterior.

Este esquema algorítmico mantiene correspondencia directa con la arquitectura del software y con la estructura del modelo matemático implementado, lo que facilita tanto la validación como la futura extensión del sistema.

4.4.5. Biblioteca de Problemas con Resultados Esperados para la Validación del Simulador

Como parte de la estrategia de verificación, el simulador incorpora una biblioteca mínima de casos de referencia integrada mediante archivos en formato .reactsim. Estos archivos permiten cargar configuraciones predefinidas y contrastar el comportamiento numérico del software con soluciones analíticas o con respuestas esperadas del modelo.

Los casos de referencia considerados fueron:

Caso 1. Batch isotérmico de primer orden.

Se utilizó como caso base de validación analítica, dado que admite solución cerrada para la conversión:

$$X(t) = 1 - e^{-kt}$$

Caso 2. Batch isotérmico de orden cero.

Permitió verificar la evolución lineal de la concentración y contrastar el comportamiento numérico frente a una solución simple.

Caso 3. Batch isotérmico de segundo orden.

Se empleó para validar el comportamiento del modelo cuando la ley cinética depende de forma no lineal de la concentración. En este caso, la relación analítica clásica se expresa como:

$$\frac{X_A}{1 - X_A} = kC_{A0}t$$

Caso 4. Batch no isotérmico adiabático con Arrhenius.

Permitió verificar el acoplamiento entre concentración y temperatura, así como la sensibilidad térmica del sistema.

Caso 5. Batch no isotérmico con chaqueta.

Se utilizó para analizar el efecto del intercambio térmico sobre la trayectoria de temperatura y sobre la evolución de la conversión.

Cada caso de referencia se acompañó de una plantilla de entradas, un conjunto de unidades coherente con el modelo, una expectativa cualitativa de la respuesta y un criterio de aceptación basado en concordancia analítica, consistencia numérica o comportamiento esperado del perfil temporal. Esta biblioteca cumple una doble función: apoyar la validación del simulador y ofrecer casos reproducibles dentro del propio software.

4.4.6. Métodos Numéricos Aplicados (Criterio Metodológico del Software)

En coherencia con la estrategia descrita en la Sección 6.3, la resolución numérica del simulador se implementó mediante la función solve_ivp de SciPy, utilizando los métodos RK45, Radau y BDF. La herramienta ofrece dos modos de selección: uno

automático, basado en la naturaleza del caso, y otro manual, en el que el usuario puede escoger explícitamente el integrador.

Desde el punto de vista metodológico, el criterio seguido por el software es el siguiente:

- RK45 se adopta como opción base para problemas isotérmicos o escenarios no rígidos.
- Radau y BDF se priorizan en casos con acoplamiento térmico, dependencia de Arrhenius o mayor sensibilidad numérica.
- Las tolerancias rtol y atol permiten ajustar la precisión requerida según la finalidad de la simulación.

En el modo automático, el simulador aplica una heurística de selección del integrador. Esta prioriza métodos implícitos en casos con acoplamiento térmico, dependencia de Arrhenius o cinética personalizada, y mantiene RK45 como primera opción en escenarios no rígidos o numéricamente regulares. Además de resolver el sistema, el programa registra diagnósticos del proceso de integración, tales como el método finalmente utilizado, los métodos evaluados, el número de evaluaciones realizadas y otros indicadores relevantes del solver. Esta información aporta trazabilidad al cálculo y facilita la interpretación del comportamiento numérico en cada caso.

4.5. Interfaz de Entrada, Gestión de Casos y Criterios de Aceptación del Simulador

La operación de ReactSim Studio requiere una gestión estructurada de entradas, de manera que el usuario pueda definir casos de reactores batch ideales homogéneos sin ambigüedades y con coherencia física. Para ello, el diseño del software se basó en un enfoque orientado a casos, en el cual los parámetros requeridos, las opciones disponibles y los resultados generados dependen del régimen térmico y de la formulación seleccionada para el problema.

Este enfoque permite que la interfaz mantenga correspondencia directa con el modelo físico-matemático implementado, evitando solicitudes de datos irrelevantes y reduciendo la probabilidad de errores de configuración. En consecuencia, el simulador no solo resuelve el sistema de ecuaciones planteado, sino que guía al usuario dentro de un flujo de trabajo consistente con la lógica de la Ingeniería de las Reacciones Químicas y con el alcance definido para la versión desarrollada.

4.5.1. Estructura de la Interfaz de Entrada por Tipo de Caso

La interfaz gráfica del simulador se organiza mediante una estructura jerárquica de pestañas, diseñada para conducir al usuario a través de una secuencia lógica de definición del caso. En la versión implementada, el panel de entrada se distribuye en bloques funcionales que agrupan la información requerida por el modelo:

4.5.1.1. Datos generales.

Incluye el título del caso, el modo térmico, la estrategia de selección del solver, la especie de referencia, el volumen del reactor, el intervalo temporal de simulación, el número de puntos de salida y las tolerancias numéricas.

4.5.1.2. Especies.

Permite ingresar las especies participantes, sus coeficientes estequiométricos y sus concentraciones iniciales.

4.5.1.3. Cinética.

Permite seleccionar una cinética de ley de potencia o una expresión cinética personalizada validada, así como definir los parámetros correspondientes, incluyendo constante cinética, órdenes de reacción o parámetros de Arrhenius cuando corresponda.

4.5.1.4. Energía.

Agrupa los parámetros térmicos del sistema, tales como calor de reacción, densidad, calor específico, coeficiente global de transferencia, área de intercambio, temperatura de chaqueta y temperatura máxima permisible.

4.5.1.5. Alcance.

Presenta al usuario una síntesis del marco de validez del simulador y de las hipótesis principales bajo las cuales opera el modelo.

Esta organización permite activar o desactivar campos según el régimen térmico seleccionado, de modo que solo se soliciten los datos pertinentes para el caso en cuestión. Así, por ejemplo, los parámetros de transferencia de calor solo se habilitan en el modo no isotérmico con chaqueta, mientras que en el modo isotérmico el balance de energía queda desactivado.

4.5.2. Gestión de Unidades y Consistencia Dimensional

Con el fin de preservar la coherencia física de los cálculos, ReactSim Studio adopta un sistema interno de unidades basado en el Sistema Internacional. Todas las ecuaciones del modelo matemático se resuelven con unidades coherentes, independientemente de las unidades en las que el usuario introduzca los datos desde la interfaz.

Para facilitar el uso del simulador, la interfaz permite el ingreso de magnitudes en unidades habituales en Ingeniería Química, tales como mol/L, mol/m^3, L, m^3, s, min, h, K, °C, J/mol y kJ/mol. Antes de iniciar la integración, el programa convierte automáticamente estas magnitudes al sistema interno del modelo.

Además de la conversión de unidades, el simulador incorpora verificaciones básicas de consistencia dimensional y de validez física, entre ellas:

- Compatibilidad entre magnitudes de concentración, volumen y tiempo.
- Coherencia entre parámetros energéticos y térmicos.
- Validación de signos esperados en ΔH_r.

- Verificación de temperaturas físicamente admisibles.
- Prevención de configuraciones incompatibles con el modo térmico seleccionado.

Estas validaciones no alteran el modelo matemático, sino que funcionan como un control preventivo orientado a mejorar la confiabilidad del proceso de simulación y a evitar configuraciones físicamente inconsistentes.

4.5.3. Gestión de Casos y Persistencia de Configuración

El simulador incorpora un mecanismo de gestión de casos que permite guardar, abrir y reutilizar configuraciones completas de simulación mediante un formato propio con extensión .reactsim. Este formato almacena de forma estructurada los bloques de entrada del caso, incluyendo especies, estequiometría, cinética, parámetros térmicos, configuración numérica y metadatos asociados.

- Conservar configuraciones reproducibles de simulación.
- Compartir casos entre usuarios.
- Cargar casos de referencia integrados desde la propia aplicación.
- Evitar la reintroducción manual de parámetros en escenarios repetitivos.

Adicionalmente, ReactSim Studio incluye una biblioteca integrada de casos de referencia en formato .reactsim, accesible desde la interfaz, la cual permite cargar configuraciones representativas del comportamiento isotérmico y no isotérmico del reactor batch ideal.

4.5.4. Resultados Generados por el Simulador

Una vez ejecutada la simulación, el programa genera un conjunto de resultados orientados tanto a la interpretación técnica como a la verificación numérica del caso. Entre los resultados estándar que produce la herramienta se encuentran:

- Perfiles temporales de concentración de las especies químicas.
- Evolución temporal de la conversión de la especie de referencia.
- Evolución temporal de la temperatura del reactor, cuando el caso es no isotérmico.
- Velocidad de reacción en función del tiempo.
- Calor generado por reacción.
- Calor removido o suministrado por la chaqueta, cuando corresponda.
- Identificación de la especie de referencia y del reactivo limitante.
- Temperatura máxima alcanzada y verificación respecto a la temperatura máxima permisible.
- Tiempo recomendado de operación o parada.
- Diagnóstico del solver utilizado y de los métodos evaluados.

La interfaz organiza estos resultados mediante varias vistas complementarias: resumen textual, gráficas interactivas, tabla de perfiles, mensajes del sistema y consola de cálculo. Esta disposición permite interpretar el caso desde perspectivas distintas sin perder trazabilidad entre modelo, integración y salida numérica.

4.5.5. Criterios de Aceptación y Validación Técnica del Simulador

Con el objetivo de verificar el correcto funcionamiento del simulador, se establecieron criterios de aceptación basados en la comparación de los resultados obtenidos con soluciones analíticas, respuestas esperadas del modelo y comportamientos numéricos consistentes entre métodos de integración. Para cada caso de validación se consideraron, de manera general, los siguientes criterios:

- Concordancia con soluciones analíticas cuando estas existen.
- Reproducción correcta de tendencias dinámicas esperadas en perfiles de concentración y temperatura.
- Consistencia de la respuesta frente a cambios razonables en tolerancias numéricas.
- Estabilidad del integrador durante el intervalo de simulación.
- Ausencia de estados físicamente inadmisibles, como concentraciones negativas persistentes o temperaturas no válidas.

El proceso de evaluación del simulador tuvo un alcance técnico y computacional. No se incluyeron pruebas formales con estudiantes ni estudios pedagógicos de usabilidad, dado que el trabajo se centró en el desarrollo, implementación y verificación del modelo físico-matemático, así como en su resolución numérica.

4.5.6. Validación del Simulador

ReactSim Studio se evaluó con el propósito de verificar la coherencia matemática, física y numérica del modelo implementado, en correspondencia con los supuestos del reactor batch ideal homogéneo y con los métodos de integración empleados. Dado que el trabajo se fundamentó en modelos clásicos ampliamente documentados en la literatura de Ingeniería de las Reacciones Químicas, la comprobación del desempeño del software se realizó mediante contrastes con soluciones de referencia.

Para la operación isotérmica, el simulador se contrastó con soluciones analíticas disponibles para reacciones de orden cero, primero y segundo. En estos casos, se compararon los perfiles temporales de concentración y conversión con las expresiones cerradas derivadas de la ecuación de diseño del reactor batch, verificando que la respuesta numérica reprodujera correctamente la forma funcional esperada y que las discrepancias permanecieran dentro de márgenes aceptables de error.

En la condición no isotérmica adiabática, la validación se basó en la revisión conjunta de los perfiles de concentración y temperatura, comprobando que el simulador reprodujera el acoplamiento térmico-cinético esperado para reacciones con dependencia de Arrhenius. En estos casos se evaluó la sensibilidad del sistema frente a la energía de activación, el calor de reacción y las propiedades térmicas.

En el modo no isotérmico con chaqueta, la validación consideró la evolución simultánea de la concentración, la temperatura y la transferencia de calor con el entorno. Se verificó que el simulador reprodujera comportamientos compatibles con

la teoría, tales como la estabilización térmica bajo una remoción adecuada de calor o la presencia de máximos térmicos en escenarios exotérmicos.

De forma complementaria, se realizó una validación numérica del integrador solve_ivp, comparando el comportamiento de los métodos RK45, Radau y BDF sobre los mismos casos de prueba. Para ello, se ejecutaron simulaciones con distintos valores de tolerancia relativa y absoluta, verificando que las soluciones convergieran hacia perfiles equivalentes al refinar los criterios de error. Asimismo, se examinaron casos con mayor sensibilidad térmica para comprobar que el uso de métodos implícitos preservara la estabilidad sin sacrificar consistencia física.

En consecuencia, el simulador se consideró técnicamente aceptado, dado que fue capaz de reproducir adecuadamente los casos isotérmicos con solución analítica, mantuvo tendencias dinámicas coherentes en los problemas no isotérmicos de referencia y demostró estabilidad y convergencia numérica bajo distintas configuraciones de integración.

4.5.7. Contrastación Externa con Cantera

Como complemento a la validación mediante soluciones analíticas y problemas de referencia de Fogler, se incorporó una contrastación externa utilizando Cantera 3.2.0 como herramienta abierta de simulación química. Cantera es una herramienta de código abierto orientada al análisis de cinética química, termodinámica y procesos de transporte, por lo que resulta adecuada como entorno externo de comparación para casos dinámicos simplificados de sistemas reaccionantes (Goodwin et al., 2025; Cantera Developers, 2025).

La comparación no se formuló como una equivalencia total entre ambos programas, debido a que Cantera posee un alcance más amplio orientado a cinética química detallada, termodinámica, transporte, redes de reactores y sistemas multiespecie. ReactSim Studio, por su parte, fue desarrollado específicamente para reactores batch ideales homogéneos en fase líquida, de volumen constante y con una sola reacción irreversible. Por ello, la contrastación se realizó mediante casos equivalentes simplificados, configurados en Cantera con pseudoespecies A y P, una reacción irreversible $A \rightarrow P$ y parámetros cinéticos coherentes con los casos evaluados en ReactSim Studio. Bajo este enfoque, Cantera se empleó como una herramienta externa de verificación numérica para comprobar que ReactSim Studio reproduce respuestas coherentes cuando ambos entornos se someten a condiciones de simulación equivalentes.

La contrastación se efectuó en dos escenarios. El primero correspondió a un reactor batch isotérmico de segundo orden, con cinética $r = kC_A^2$, temperatura constante y solución analítica cerrada. Este caso permitió comparar simultáneamente los resultados de ReactSim Studio, Cantera y la solución teórica del modelo. El segundo escenario correspondió a un caso adiabático basado en el Ejemplo 9-1 de Fogler, en el cual se configuró una cinética de primer orden con dependencia de Arrhenius y un balance energético equivalente. En este caso, la comparación permitió evaluar la consistencia del acoplamiento entre concentración, velocidad de reacción y temperatura.

En ambos casos, las variables de comparación fueron la concentración final del reactivo, la concentración final del producto, la conversión y la temperatura final. Los errores relativos se calcularon tomando como referencia los resultados de ReactSim Studio y comparándolos con los valores obtenidos en Cantera. Este procedimiento permitió incorporar una validación externa complementaria, orientada a verificar que la lógica de cálculo implementada en ReactSim Studio produce resultados coherentes frente a un software abierto independiente.

4.5.8. Relación con los Objetivos de Trabajo

La metodología aplicada mantuvo correspondencia directa con los objetivos específicos de la investigación, de modo que cada etapa del desarrollo respondió a una acción concreta dentro del proceso de construcción y validación del simulador.

El primer objetivo específico, relacionado con la formulación de modelos matemáticos, se cumplió mediante la definición del reactor batch ideal homogéneo, sus supuestos, variables de estado, condiciones iniciales, ley cinética, balance de materia y balance de energía. Esta etapa permitió establecer el sistema de ecuaciones diferenciales que describe la evolución temporal de las concentraciones y, en los casos no isotérmicos, de la temperatura.

El segundo objetivo específico, correspondiente a la implementación numérica, se cumplió al transformar el sistema de ecuaciones formulado en una función dinámica implementada en Python. En esta función, el vector de estado contiene las concentraciones de las especies y, cuando aplica, la temperatura del reactor. A partir de dicho vector y de los parámetros definidos por el usuario, el programa calcula las velocidades de reacción, los términos estequiométricos, el balance de energía y las derivadas requeridas por el integrador numérico.

El tercer objetivo específico, referido al desarrollo de la herramienta computacional, se cumplió mediante la construcción de una interfaz gráfica que permite configurar los casos de simulación, introducir parámetros cinéticos, térmicos y operativos, ejecutar el cálculo numérico, visualizar perfiles de concentración, conversión y temperatura, guardar archivos de caso y exportar resultados. De esta forma, el modelo matemático no quedó limitado a una formulación teórica, sino que fue integrado en una herramienta funcional para el análisis de reactores batch ideales.

El cuarto objetivo específico, asociado con la validación del software, se cumplió mediante la comparación de los resultados de ReactSim Studio con soluciones analíticas, soluciones semianalíticas, problemas de referencia de Fogler, pruebas de estabilidad numérica y casos equivalentes resueltos en Cantera. Esta etapa permitió verificar que el simulador produce resultados coherentes con el modelo formulado, con los métodos numéricos seleccionados y con referencias externas de comparación.

En conjunto, esta relación entre objetivos y metodología evidencia que el desarrollo de ReactSim Studio siguió una secuencia lógica: formulación del modelo, traducción del modelo a código, resolución numérica, construcción de la interfaz y validación técnica de los resultados.

4.6. Arquitectura Conceptual del Simulador y Criterios de Diseño

Desde el punto de vista metodológico, la arquitectura conceptual de ReactSim Studio responde a la necesidad de mantener una relación directa entre la formulación físico-matemática del reactor, la estrategia numérica empleada para resolver el modelo y la forma en que los resultados se presentan al usuario. Esta arquitectura no se definió únicamente con criterios de programación, sino con base en principios de coherencia científica, trazabilidad del cálculo, claridad operativa y posibilidad de extensión futura.

En primer lugar, el diseño del simulador buscó garantizar que el comportamiento computacional del sistema correspondiera de manera explícita a los balances de materia y energía formulados en el marco teórico. Esto implicó que la lógica interna del software no se apoyara en aproximaciones empíricas ajenas al modelo, sino en una implementación fiel de las ecuaciones que describen el comportamiento del reactor batch ideal homogéneo bajo el alcance adoptado en el presente trabajo.

En segundo lugar, la arquitectura se concibió para separar con claridad el modelo físico del método numérico empleado para su resolución. Esta distinción fue importante porque permitió comprender que el comportamiento del reactor está determinado por la formulación del problema, mientras que el integrador actúa como una herramienta matemática para aproximar la solución del sistema de ecuaciones diferenciales ordinarias. La separación entre ambos niveles mejoró la claridad conceptual del simulador y favoreció la extensibilidad del sistema.

Finalmente, el diseño arquitectónico también priorizó la organización de la información de entrada y la presentación de resultados, de manera que el usuario pudiera relacionar las variables ingresadas, el procedimiento de simulación y la respuesta del reactor sin perder trazabilidad entre modelo, integración y salida numérica. Por ello, la arquitectura del simulador se estructuró en tres niveles conceptuales interrelacionados: el nivel físico-matemático, el nivel numérico-computacional y el nivel de interfaz e interpretación visual.

4.6.1. Nivel Físico-Matemático del Modelo del Reactor

El primer nivel de la arquitectura corresponde al plano físico-matemático y se vincula directamente con el modelo desarrollado en la Sección 6.2. En este nivel se ubican los balances de materia y energía del reactor batch ideal homogéneo, la formulación cinética de la reacción, la dependencia térmica mediante la ecuación de Arrhenius y los supuestos de idealidad adoptados para delimitar el alcance del sistema.

Este nivel representa el núcleo conceptual del simulador, ya que define la estructura del problema que se desea resolver. En él quedan especificadas las variables de estado, las condiciones iniciales, los parámetros cinéticos y térmicos, así como las relaciones matemáticas que gobiernan la evolución temporal del reactor en sus modos isotérmico y no isotérmico.

Desde la perspectiva computacional, este nivel se tradujo en la construcción de una función dinámica general de la forma:

$$\frac{dy}{dt} = f(t, y, \theta)$$

Esta función representa el comportamiento del reactor a lo largo del tiempo. Sin embargo, la importancia metodológica de este nivel no radica únicamente en su implementación algebraica, sino en el hecho de que cada término de la función dinámica mantiene correspondencia directa con un principio físico o químico previamente justificado en el marco teórico.

La existencia de este nivel permitió que el simulador no operara como una caja negra, sino como una traducción computacional explícita del modelo clásico del reactor batch ideal. En consecuencia, cualquier resultado generado por el software puede interpretarse y rastrearse conceptualmente a partir de los balances fundamentales que le dan origen.

4.6.2. Nivel Numérico-computacional (Resolución del Modelo)

El segundo nivel corresponde al plano numérico-computacional y se articula con la estrategia de integración temporal descrita en la Sección 6.3. Su función principal consiste en resolver el sistema de ecuaciones diferenciales ordinarias definido en el nivel físico-matemático, sin alterar la estructura del modelo ni los supuestos que lo sustentan.

En este nivel se ubican los métodos numéricos empleados por el simulador, el control de tolerancias, la generación de puntos de salida, la construcción del problema de valor inicial y la lógica de selección del solver. ReactSim Studio implementa esta capa mediante solve_ivp de SciPy, utilizando los métodos RK45, Radau y BDF, según la naturaleza del caso y la estrategia de resolución seleccionada.

La separación entre este nivel y el nivel físico-matemático resulta especialmente importante, ya que permite distinguir entre:

- El modelo que describe el comportamiento del reactor.
- El procedimiento numérico que aproxima su solución.

Esta diferenciación evita confundir la formulación del reactor con la técnica de integración utilizada, y refuerza la idea de que la estabilidad, precisión y costo computacional dependen del método numérico, pero no modifican el significado físico del sistema modelado.

Además, este nivel incluye operaciones complementarias necesarias para la robustez del cálculo, tales como la normalización al Sistema Internacional, la validación preliminar de entradas, la prevención de estados no físicos y la recopilación de diagnósticos del integrador. Todo ello contribuye a que la solución numérica sea no solo computacionalmente viable, sino también físicamente interpretable.

4.6.3. Nivel de Interfaz, Gestión de Casos e Interpretación Visual

El tercer nivel de la arquitectura se orienta a la interacción con el usuario y a la interpretación de los resultados. En este nivel se agrupan la interfaz gráfica, la gestión de casos, la visualización de perfiles, la exportación de resultados y los mecanismos de apoyo a la lectura técnica de la simulación.

A diferencia de los niveles anteriores, aquí no se formula ni se resuelve el modelo matemático, sino que se organiza la información de forma que el usuario pueda definir el caso, ejecutar la simulación y analizar la respuesta del reactor con claridad. Este nivel incluye la estructura de pestañas para datos generales, especies, cinética, energía y alcance; la presentación de resultados en forma de resumen, gráficas, tablas, mensajes y consola de cálculo; así como la gestión de archivos en formato .reactsim para guardar, cargar y reutilizar configuraciones.

Desde el punto de vista metodológico, este nivel cumple una función importante porque transforma una solución matemática en una representación técnica interpretable. La visualización de perfiles temporales de concentración, conversión, temperatura, velocidad de reacción y transferencia de calor permite relacionar directamente los resultados numéricos con el comportamiento físico del reactor. Asimismo, herramientas como la identificación del reactivo limitante, el tiempo recomendado de operación y la verificación de temperatura máxima permisible refuerzan la utilidad del simulador como instrumento de análisis.

Por tanto, este nivel no debe entenderse únicamente como una capa visual del sistema, sino como una parte funcional de la arquitectura, encargada de vincular la formulación del problema con la lectura técnica de sus resultados.

4.6.4. Relación entre la Arquitectura y Metodología

La arquitectura conceptual descrita en esta sección cumple una función complementaria a la del algoritmo operativo presentado en la Sección 6.4. Mientras el algoritmo explica cómo se ejecuta el simulador paso a paso, la arquitectura establece por qué el sistema fue estructurado de esa manera desde una perspectiva metodológica y científica.

Esta distinción permite separar con claridad la dimensión operativa del software de su fundamentación conceptual. La metodología no solo describe procedimientos, sino que también justifica decisiones de diseño vinculadas con la fidelidad del modelo, la trazabilidad del cálculo, la robustez numérica y la claridad en la interpretación de resultados.

En este sentido, la arquitectura de ReactSim Studio actúa como un marco organizador que articula teoría, simulación e interfaz dentro de un mismo sistema coherente. Gracias a ello, el simulador puede entenderse no solo como una aplicación funcional, sino como una implementación estructurada del modelo del reactor batch ideal homogéneo, diseñada para mantener correspondencia entre el problema físico-matemático, su resolución numérica y su presentación final al usuario.

4.7. Síntesis y Alcance Metodológico del Trabajo

La metodología desarrollada en este capítulo establece de manera sistemática el marco conceptual, matemático, numérico y computacional que sustenta el diseño e implementación de ReactSim Studio como herramienta de simulación de reactores batch ideales homogéneos, isotérmicos y no isotérmicos. En conjunto, las secciones precedentes configuran un enfoque integrado que articula los fundamentos de la Ingeniería de las Reacciones Químicas con criterios de modelado, resolución numérica, diseño algorítmico e implementación de software.

En primer lugar, se definió un enfoque de investigación aplicada, cuantitativa, determinística y no experimental, coherente con la naturaleza del trabajo y con la ausencia de experimentación física directa. Este enfoque permitió centrar el desarrollo en la formulación rigurosa de modelos clásicos de reactores batch, en la selección de estrategias numéricas apropiadas para su resolución y en su posterior implementación computacional dentro de un entorno controlado y reproducible.

En segundo lugar, se estableció el modelo matemático del reactor batch ideal homogéneo bajo el alcance adoptado por el simulador, considerando una sola reacción química irreversible, fase líquida, volumen constante y operación isotérmica o no isotérmica. A partir de esta formulación, se definió la estrategia de resolución numérica mediante el planteamiento del problema como un problema de valor inicial y el uso de métodos de integración temporal adecuados para sistemas no lineales y potencialmente rígidos.

Asimismo, se describió el diseño algorítmico del simulador, la estructura de sus entradas, la normalización al Sistema Internacional, la organización de casos de uso y la incorporación de casos de referencia para validación. Esto permitió delimitar con claridad el alcance funcional del software y asegurar que el proceso de simulación conservara coherencia entre teoría, configuración de caso, resolución numérica y resultados obtenidos.

De igual manera, se estableció una arquitectura conceptual organizada en los niveles físico-matemático, numérico-computacional y de interfaz e interpretación visual. Esta estructura permite separar el modelo del reactor, el método de solución y la presentación de resultados, evitando ambigüedades entre fenómeno físico, algoritmo numérico y herramienta informática. Como consecuencia, el simulador mantiene trazabilidad entre las ecuaciones implementadas, la integración temporal aplicada y la información que finalmente se presenta al usuario.

En conjunto, la metodología adoptada demuestra que el desarrollo del simulador siguió una secuencia coherente con los objetivos de la investigación. La formulación matemática permitió definir el comportamiento del reactor; la implementación numérica convirtió dicho modelo en una función computacional resoluble; el desarrollo de la interfaz permitió operar el simulador a partir de datos ingresados por el usuario; y la validación técnica permitió comprobar la consistencia de los resultados obtenidos. De esta manera, ReactSim Studio se desarrolló como una extensión funcional del modelo teórico hacia una herramienta computacional aplicable al análisis de reactores batch ideales homogéneos.

Por tanto, el capítulo metodológico proporciona la base necesaria para proceder, en el capítulo siguiente, a la presentación de resultados, la validación del simulador y el análisis del comportamiento de los casos implementados.

4.8. Entregables del Proyecto

Como resultado del desarrollo de la presente investigación, se generaron los siguientes entregables, orientados a materializar el modelo propuesto en una herramienta computacional funcional, técnicamente sustentada y disponible para su análisis, uso y posible ampliación futura:

1. **Simulador computacional ReactSim Studio**, desarrollado en el lenguaje de programación Python, para la simulación de reactores batch ideales homogéneos, isotérmicos y no isotérmicos, en fase líquida, de volumen constante y con una sola reacción química irreversible, mediante la resolución numérica de balances de materia y energía.
2. **Código fuente completo del simulador**, organizado de forma modular y documentada, dispuesto en un repositorio público en GitHub bajo licencia MIT, lo que permite su libre uso, modificación, análisis y distribución.
3. **Motor de simulación numérica**, implementado con base en la integración de sistemas de ecuaciones diferenciales ordinarias mediante la función solve_ivp de la biblioteca SciPy, con soporte para los métodos RK45, Radau y BDF, según la naturaleza del problema y la estrategia de resolución definida.
4. **Interfaz gráfica de usuario**, diseñada para permitir la configuración estructurada de casos de simulación, la introducción de parámetros cinéticos, térmicos y operativos, la gestión de archivos de caso y la visualización de resultados en forma de resúmenes, tablas, gráficas y mensajes de diagnóstico.
5. **Biblioteca de benchmarks y validación**, integrada al simulador mediante archivos en formato .reactsim, incluyendo problemas isotérmicos de orden cero, primero y segundo, así como casos no isotérmicos adiabáticos y con chaqueta, utilizados para verificar la correcta implementación del modelo matemático y del esquema numérico.
6. **Documentación técnica del software**, en la que se describen la formulación matemática implementada, los supuestos adoptados, la estrategia de resolución numérica, la lógica algorítmica del simulador, su arquitectura conceptual, sus alcances funcionales y sus limitaciones actuales.
7. **Manual de usuario y guía de operación**, con instrucciones para la instalación, ejecución y uso básico del simulador, así como lineamientos para la carga y guardado de casos, la interpretación de resultados y el empleo de casos de referencia integrados.
8. **Versión distribuible del software para entorno Windows**, generada a partir del proyecto mediante herramientas de empaquetado, con el propósito de facilitar su ejecución fuera del entorno de desarrollo.

En conjunto, estos entregables materializan los objetivos del trabajo, al integrar formulación teórica, resolución numérica, desarrollo de software y validación técnica dentro de una misma plataforma computacional.

V. Resultados, Implementación y Discusión

5.1. Implementación del Algoritmo Interno de ReactSim Studio

El resultado principal del presente trabajo no se limitó a la formulación teórica del reactor batch ideal homogéneo, sino que se materializó en la implementación de un sistema computacional capaz de traducir dicha formulación a un proceso de simulación funcional, estructurado y trazable. En este sentido, ReactSim Studio fue desarrollado como una herramienta que integra la definición del caso, la validación física y numérica de las entradas, la construcción del modelo matemático, la resolución del sistema de ecuaciones diferenciales ordinarias y la presentación organizada de resultados dentro de una misma plataforma.

Desde una perspectiva algorítmica, el funcionamiento interno del simulador responde a una secuencia lógica bien definida, en la que cada etapa cumple una función específica dentro del flujo general de simulación. Este flujo comienza con la captura de datos del caso a través de la interfaz gráfica y continúa con la transformación de dichas entradas a una representación estructurada del problema. Posteriormente, el sistema verifica la coherencia de los datos suministrados, normaliza las magnitudes al Sistema Internacional, construye la función dinámica del reactor, resuelve numéricamente el problema de valor inicial y finalmente genera las variables derivadas y los elementos visuales requeridos para la interpretación de los resultados.

La implementación de este algoritmo se diseñó con el propósito de mantener correspondencia directa entre la lógica del software y la formulación físico-matemática del modelo. Por ello, ReactSim Studio no opera como un entorno de cálculo opaco, sino como una herramienta cuya secuencia de ejecución puede rastrearse desde los datos de entrada hasta las salidas finales del simulador. Esta trazabilidad constituye uno de los rasgos metodológicos más importantes del sistema desarrollado, ya que permite relacionar de manera explícita la teoría del reactor batch con su resolución computacional.

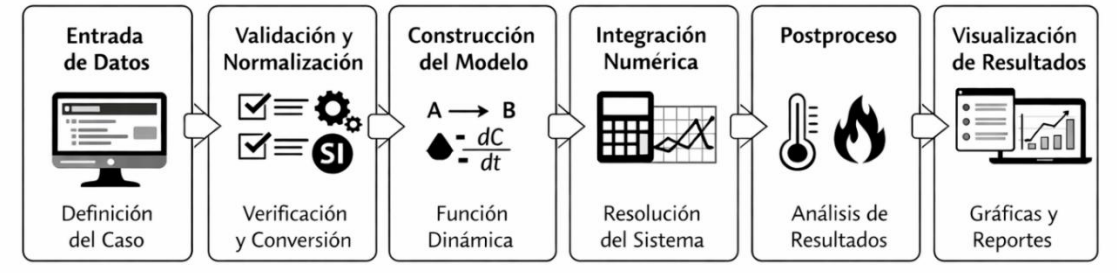
1. **Construcción del caso de simulación**, a partir de los datos suministrados por el usuario en la interfaz gráfica.
2. **Validación y normalización del caso**, verificando coherencia física, compatibilidad entre parámetros y conversión de unidades al Sistema Internacional.
3. **Construcción de la función cinética y del sistema dinámico**, según el tipo de ley de velocidad definido para el problema.
4. **Resolución numérica del sistema**, mediante el integrador seleccionado y bajo los criterios de control de error establecidos.
5. **Postproceso y generación de resultados**, incluyendo conversión, temperatura, velocidad de reacción, calor generado, calor removido, diagnósticos del solver y criterios adicionales de evaluación.
6. **Presentación de resultados en la interfaz**, mediante gráficas, tablas, mensajes y reportes exportables.

Este encadenamiento de etapas asegura que la simulación no sea tratada como un cálculo aislado, sino como un proceso integral de modelado, solución e interpretación. En consecuencia, la implementación algorítmica de ReactSim Studio constituye la base operativa del simulador y representa el punto de convergencia entre la formulación matemática desarrollada en capítulos anteriores y los resultados que se presentan en el presente capítulo.

Asimismo, la estructura del algoritmo fue organizada de manera modular, lo que permitió separar funciones de entrada, validación, cinética, integración, postproceso y visualización. Esta decisión de diseño no solo facilitó el desarrollo del software, sino que también contribuyó a mejorar su mantenibilidad, extensibilidad y capacidad de verificación. Gracias a ello, el simulador puede ser analizado no solo como una aplicación funcional, sino como una implementación estructurada de la lógica de simulación de reactores batch ideales homogéneos.

Por tanto, la implementación del algoritmo interno de ReactSim Studio constituye un resultado central de la investigación, ya que demuestra la viabilidad de traducir un modelo clásico de Ingeniería de las Reacciones Químicas a una herramienta computacional abierta, coherente y técnicamente sustentada. A partir de esta base, las secciones siguientes profundizan en el flujo interno del simulador, la organización de sus módulos, las funcionalidades implementadas y la validación de sus resultados frente a soluciones de referencia.

Figura 5.
Flujo general de operación de ReactSim Studio



Nota. Elaboración Propia.

5.2. Descripción del Algoritmo de Simulación Implementado

El algoritmo de simulación implementado en ReactSim Studio fue desarrollado para ejecutar de forma ordenada el proceso completo de resolución de un reactor batch ideal homogéneo, desde la captura de datos hasta la construcción del resultado final. A diferencia de un esquema de cálculo monolítico, la lógica interna del programa se organizó en etapas diferenciadas, cada una de las cuales responde a una función específica dentro de la simulación. Esta organización permite mantener claridad estructural en el código, facilitar la verificación del funcionamiento del sistema y preservar la correspondencia entre la formulación del modelo y su implementación computacional.

Desde el punto de vista operativo, el algoritmo parte de la información ingresada por el usuario en la interfaz gráfica, la transforma en una representación estructurada del caso, valida su coherencia, construye la función dinámica correspondiente y resuelve el problema mediante integración numérica. Posteriormente, el sistema procesa los perfiles obtenidos, calcula variables derivadas de interés técnico y prepara la información para su presentación visual y exportación.

5.2.1. Construcción del Caso de Simulación

La primera etapa del algoritmo consiste en transformar la información capturada por la interfaz gráfica en una estructura de datos coherente con el problema que se desea resolver. Para ello, el simulador organiza las entradas del usuario en bloques funcionales que incluyen datos generales del caso, definición de especies, parámetros cinéticos, configuración térmica y opciones del solver.

Esta información se encapsula inicialmente en una estructura de entrada que reúne todos los elementos requeridos por el simulador antes de cualquier validación física o numérica. En esta fase todavía no se ha construido el caso definitivo de simulación, sino una representación preliminar que contiene los datos tal como fueron suministrados por el usuario. Este enfoque permite separar claramente la captura de información de la lógica posterior de validación y normalización.

La ventaja de este diseño radica en que el sistema conserva una trazabilidad clara entre los campos visibles en la interfaz y los parámetros internos del modelo. De este modo, cualquier error, inconsistencia o ajuste posterior puede rastrearse hasta el bloque de entrada correspondiente, lo que favorece la robustez del proceso de simulación.

5.2.2. Validación y Normalización del Caso

Una vez construido el bloque inicial de entradas, el algoritmo procede a validar y normalizar la información antes de generar el caso definitivo de simulación. Esta etapa constituye uno de los componentes más importantes del sistema, ya que garantiza que el problema a resolver sea físicamente coherente, matemáticamente interpretable y numéricamente tratable.

La validación incluye la revisión de nombres de especies, coeficientes estequiométricos, concentraciones iniciales, dominio de temperaturas, compatibilidad entre modo térmico y parámetros suministrados, consistencia entre las unidades utilizadas y verificación de que existan reactivos y productos definidos correctamente. En paralelo, el sistema convierte todas las magnitudes al Sistema Internacional, de forma que el motor numérico trabaje siempre con un conjunto único y coherente de unidades.

En esta misma etapa se verifica también la coherencia de la ley cinética definida, ya sea en su forma de ley de potencia o mediante una expresión personalizada. Cuando se emplea una expresión personalizada, esta es compilada y evaluada de manera segura con un conjunto inicial de símbolos, a fin de detectar errores de sintaxis, dominios inválidos o respuestas no físicas antes de iniciar la simulación.

Como resultado de esta fase, el sistema genera una representación normalizada del problema, apta para ser utilizada por el motor numérico. Si la validación detecta inconsistencias, el flujo de simulación se interrumpe y la interfaz presenta mensajes de error específicos, evitando así que el programa intente resolver casos mal definidos o físicamente imposibles.

5.2.3. Construcción de la Función Cinética

Superada la etapa de validación, el algoritmo construye internamente la función que representa la velocidad de reacción. Esta función constituye uno de los elementos centrales de la simulación, ya que enlaza las concentraciones y, cuando corresponde, la temperatura con el avance temporal del sistema.

En ReactSim Studio se implementan dos modos principales de formulación cinética. El primero corresponde a la ley de potencia, en la que la velocidad se calcula a partir de una constante cinética y de las concentraciones elevadas a sus respectivos órdenes de reacción. El segundo corresponde a una expresión cinética personalizada, evaluada en un entorno restringido y seguro, en el que únicamente se habilitan símbolos físicos explícitos, como las concentraciones de las especies, el tiempo, la temperatura y la constante universal de los gases.

Cuando el caso es no isotérmico y la opción de Arrhenius está activada, la constante cinética deja de ser fija y pasa a calcularse dinámicamente en cada paso de la integración como función de la temperatura. De esta forma, la construcción de la cinética no se limita a definir una expresión algebraica, sino que establece el mecanismo mediante el cual la reacción responde al estado instantáneo del sistema.

Esta etapa permite que la formulación de la velocidad conserve coherencia con el modelo matemático adoptado y permite que el simulador trate de manera uniforme tanto los casos clásicos de ley de potencia como escenarios más flexibles definidos por el usuario.

5.2.4. Resolución Numérica del Sistema Dinámico

Una vez definida la función cinética, el algoritmo construye el sistema dinámico del reactor y procede a su resolución numérica. Para ello, se genera el vector de estado inicial, el intervalo temporal de simulación y el conjunto de puntos de evaluación en los cuales se desea reportar la solución.

A continuación, el sistema define la función de derivadas del problema, la cual calcula en cada instante las velocidades de variación de las concentraciones y, en el caso no isotérmico, la variación de temperatura a partir del balance de energía. En esta fase, el simulador aplica recortes numéricos mínimos a las concentraciones para evitar que errores de integración den lugar a evaluaciones cinéticas físicamente inconsistentes, sin alterar la lógica esencial del modelo.

La integración se realiza mediante la función solve_ivp de SciPy. Dependiendo de la estrategia de resolución seleccionada, el algoritmo puede utilizar un método definido manualmente por el usuario o aplicar una heurística de selección

automática. En esta heurística, se priorizan métodos implícitos cuando el problema presenta acoplamiento térmico, dependencia de Arrhenius o mayor sensibilidad numérica, mientras que para casos más suaves se mantiene RK45 como primera opción.

Cuando la simulación se ejecuta en modo automático, el algoritmo también evalúa el comportamiento del integrador durante la resolución y puede repetir la simulación con un método implícito si detecta señales prácticas de posible rigidez, como un número excesivo de evaluaciones de la función dinámica. Este comportamiento mejora la robustez del sistema sin ocultar al usuario la lógica seguida por el solver, ya que los métodos intentados y los diagnósticos básicos del proceso son conservados en la salida del simulador.

5.2.5. Postproceso y Generación de Resultado Final

Una vez obtenida la solución numérica del sistema, el algoritmo entra en la etapa de postproceso. En esta fase se reorganizan los perfiles calculados y se generan las variables derivadas que serán presentadas al usuario como resultado de la simulación.

Entre las variables calculadas en esta etapa se encuentran las concentraciones temporales de todas las especies, la conversión de la especie de referencia, la temperatura del reactor, la velocidad de reacción, el calor generado por reacción y, cuando existe chaqueta, el calor removido o suministrado al sistema. Asimismo, el algoritmo determina la especie de referencia efectiva, identifica el reactivo limitante y calcula indicadores adicionales como el tiempo recomendado de operación o parada y el cumplimiento o incumplimiento de la temperatura máxima permisible definida por el usuario.

Esta etapa también incorpora el ensamblaje de diagnósticos del solver, incluyendo el método finalmente utilizado, los métodos intentados y el número de evaluaciones principales del sistema. Como resultado, la salida del simulador no se limita a los perfiles fisicoquímicos del reactor, sino que incluye también información útil para interpretar el comportamiento numérico de la resolución.

El producto final de esta etapa es una estructura de resultado que contiene de manera ordenada toda la información requerida para la visualización, exportación y análisis posterior del caso simulado.

5.2.6. Presentación y Trazabilidad de Resultados en la Interfaz

La última etapa del algoritmo consiste en convertir el resultado interno del simulador en una salida interpretable para el usuario. Para ello, el programa adapta las variables calculadas a las unidades de presentación seleccionadas, actualiza las métricas principales en la interfaz y genera los elementos visuales asociados al caso.

En esta fase se dibujan los perfiles temporales, se actualizan las tablas de datos, se muestran mensajes de advertencia o confirmación, y se construye la traza interna de cálculo que permite seguir el proceso de simulación de forma más transparente.

Cuando corresponde, la interfaz marca el tiempo recomendado de operación, la temperatura máxima alcanzada y el umbral de temperatura máxima permisible, reforzando así la interpretación técnica de los resultados.

Desde el punto de vista metodológico, esta etapa es importante porque conecta la resolución numérica con la comprensión del fenómeno físico. No se trata únicamente de presentar valores calculados, sino de organizarlos de forma que el usuario pueda relacionar directamente la respuesta del reactor con el caso definido, la cinética seleccionada y el modo térmico empleado.

En conjunto, el algoritmo de simulación implementado en ReactSim Studio constituye una secuencia coherente que articula entrada, validación, modelado, integración, postproceso y visualización. Esta estructura no solo garantiza la funcionalidad del simulador, sino que también aporta claridad conceptual al proceso completo de simulación, permitiendo que la herramienta mantenga correspondencia explícita entre teoría, cálculo y resultado.

5.3. Flujograma Interno del Simulador

Con el propósito de representar de manera clara la lógica de ejecución de ReactSim Studio, se elaboraron flujogramas internos a partir de la estructura funcional real del software implementado. Estos diagramas complementan la descripción textual del algoritmo presentada en la Sección 7.2 y permiten visualizar, de forma sintética, la secuencia de operaciones que sigue el programa desde la configuración inicial del caso hasta la presentación final de los resultados.

Desde el punto de vista metodológico, la inclusión de flujogramas en este capítulo resulta pertinente porque facilita la interpretación del comportamiento interno del simulador y refuerza la trazabilidad entre la arquitectura lógica del programa y su implementación efectiva. En particular, estos diagramas permiten identificar con claridad la transición entre la entrada de datos, la validación del caso, la resolución numérica del sistema dinámico y la generación del resultado final, en coherencia con la estructura modular descrita en el capítulo metodológico.

Dado que el flujo completo del simulador contiene múltiples etapas y relaciones entre componentes, su representación en un único diagrama puede dificultar la lectura en formato de tesis. Por esta razón, se optó por dividir el flujo interno en tres representaciones complementarias: un flujograma general de operación, un flujograma específico de validación y normalización del caso, y un flujograma centrado en la integración numérica y el postproceso. Esta decisión permite mantener claridad visual sin perder fidelidad respecto al comportamiento real del software.

Adicionalmente, con el fin de preservar la trazabilidad técnica del desarrollo, el flujo interno completo del simulador podrá incorporarse en los anexos no solo como figura final, sino también mediante su código fuente en lenguaje DOT, de manera que pueda ser reutilizado, editado o renderizado nuevamente en un editor compatible con Graphviz. Esta inclusión resulta útil como respaldo documental del

software, ya que permite reconstruir el diagrama completo a partir de su definición textual original.

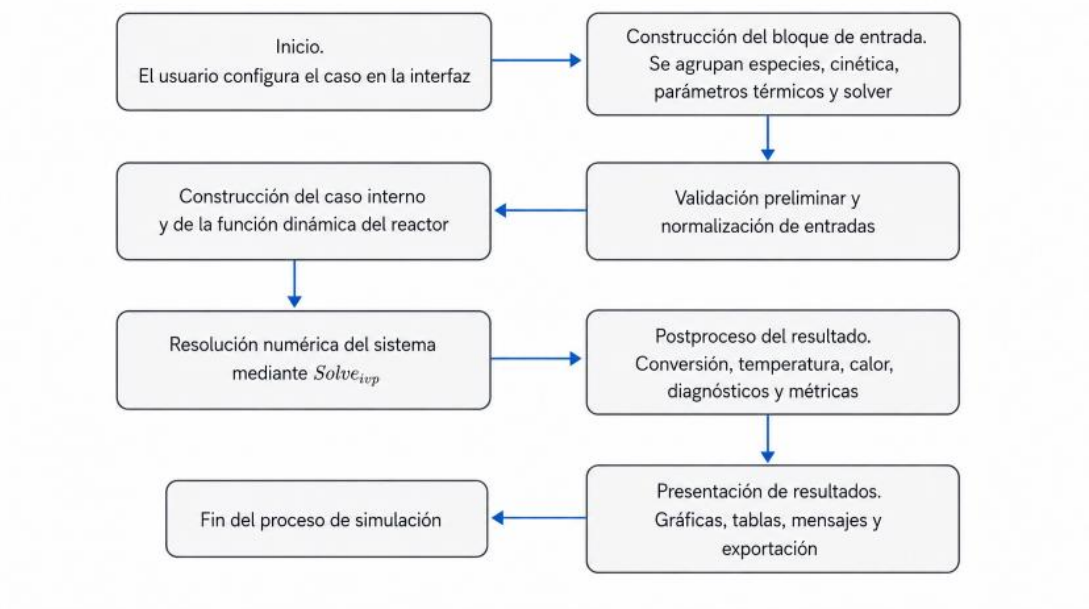
5.3.1. Flujograma General de Operación

El flujograma general resume la secuencia principal de operación de ReactSim Studio, desde la interacción inicial del usuario con la interfaz hasta la obtención del resultado final. Como se muestra en la **Figura 6**, el proceso comienza con la configuración del caso de simulación, continúa con la validación preliminar y la normalización de entradas, pasa por la construcción del modelo y la integración numérica, y concluye con el postproceso y la actualización de la interfaz.

Esta representación permite apreciar que la simulación no se ejecuta inmediatamente después de ingresar los datos, sino que depende de una cadena previa de validaciones y transformaciones que garantizan coherencia física y consistencia numérica. Asimismo, la **Figura 6** evidencia que la salida del simulador no consiste únicamente en perfiles calculados, sino en un conjunto estructurado de resultados, diagnósticos y elementos visuales preparados para su interpretación.

En consecuencia, el flujograma general sintetiza el funcionamiento de ReactSim Studio como un proceso secuencial integrado, en el que cada etapa depende de la anterior y aporta información necesaria para la siguiente. Esta visión global resulta útil para introducir los subflujos específicos que se presentan en las secciones posteriores.

Figura 6.
Flujograma general de operación de ReactSim Studio



Nota. Elaboración propia.

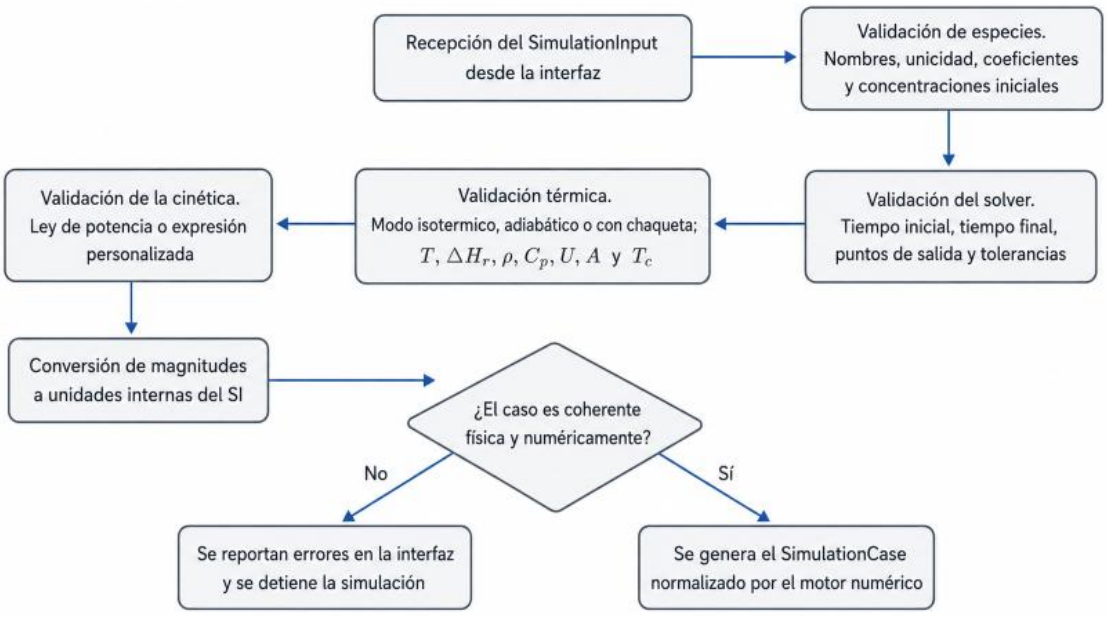
5.3.2. Flujograma de Validación y Normalización del Caso

Una vez definido el flujo general, resulta necesario aislar la etapa de validación y normalización, debido a que esta constituye una de las capas más importantes del simulador. Como se observa en la **Figura 7**, esta fase incluye la verificación de especies, la revisión de parámetros del solver, la validación térmica, el análisis de la cinética y la conversión sistemática de magnitudes al Sistema Internacional antes de construir el caso definitivo de simulación.

La **Figura 7** permite identificar que ReactSim Studio no opera directamente con los datos ingresados por el usuario, sino que primero los transforma en una representación interna consistente. Esta etapa resulta fundamental porque reduce la posibilidad de errores físicos y numéricos, detecta configuraciones incompatibles con el modelo y evita que el motor de integración reciba casos mal definidos.

Además, el flujograma muestra que la validación no se limita a controles de formato o presencia de datos, sino que incluye comprobaciones de coherencia fisicoquímica, como la existencia de reactivos y productos, la compatibilidad entre modo térmico y parámetros requeridos, y la validez de la ley cinética seleccionada. De este modo, la **Figura 7** refuerza la idea de que la robustez del simulador depende en gran medida de esta etapa intermedia de depuración y normalización.

Figura 7.
Flujograma de validación y normalización del caso de simulación



Nota. Elaboración propia.

5.3.3. Flujograma de Integración Numérica y Postproceso

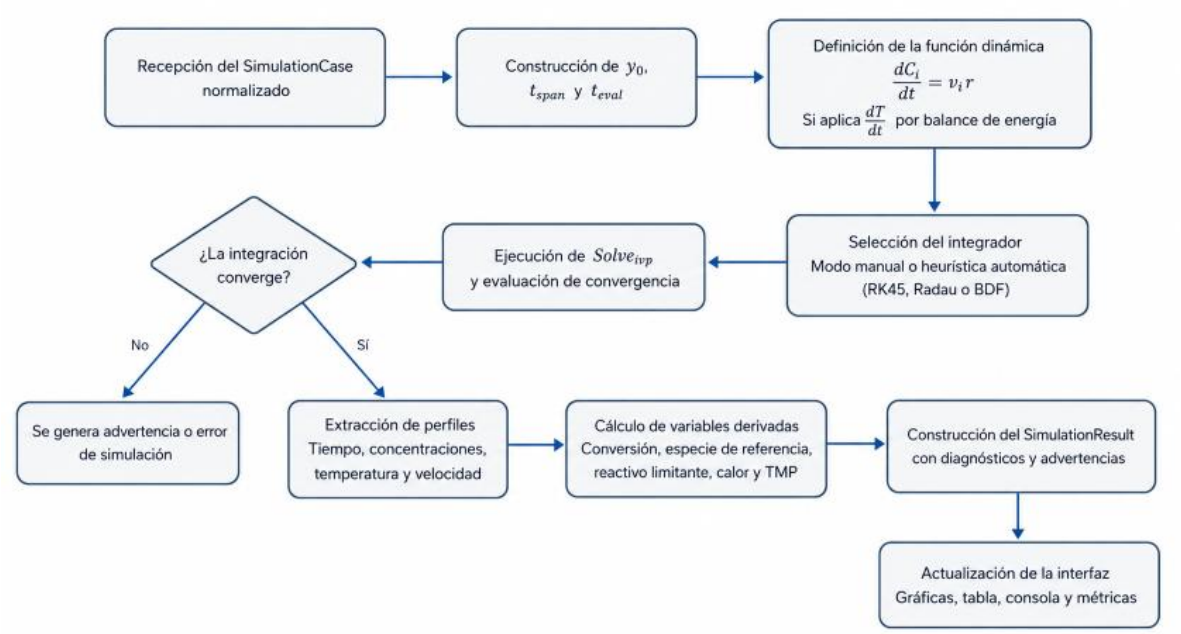
El tercer diagrama se enfoca en la fase de resolución del modelo y en la construcción de la salida del simulador. Como se muestra en la **Figura 8**, este subflujo parte de la definición del vector de estado inicial y de la función dinámica

del sistema, continúa con la selección del solver y la integración temporal mediante solve_ivp, y culmina con la generación del objeto de resultados y su traducción a elementos interpretables por la interfaz.

La **Figura 8** permite distinguir claramente entre la etapa de integración numérica propiamente dicha y el proceso posterior de postproceso. Esta separación resulta importante porque la simulación no concluye al obtener una solución del sistema de ecuaciones diferenciales, sino cuando dicha solución es reorganizada en perfiles de concentración, conversión, temperatura, calor generado, calor removido, diagnósticos del solver y criterios adicionales de evaluación, tales como el tiempo recomendado de operación o la verificación de temperatura máxima permisible.

Asimismo, el diagrama pone en evidencia el papel de la heurística de selección del solver y de los diagnósticos asociados a la convergencia del método. Por ello, la **Figura 8** no solo documenta la resolución numérica del caso, sino que también muestra cómo ReactSim Studio transforma la salida del integrador en un resultado técnico completo y visualmente interpretable.

Figura 8. Flujograma de integración numérica y postproceso en ReactSim Studio



Nota. Elaboración propia.

5.3.4. Consideraciones Sobre la Inclusión del Flujo Completo

Aunque las Figuras **6**, **7** y **8** permiten documentar adecuadamente la lógica funcional del simulador dentro del cuerpo principal de la tesis, el flujo interno completo de ReactSim Studio posee un nivel de detalle mayor que resulta útil como respaldo técnico. No obstante, su inclusión íntegra en esta sección afectaría la legibilidad del documento debido al tamaño y densidad del diagrama.

Por esta razón, el flujograma completo fue incorporado en el **Anexo A**, mientras que en el capítulo principal se conservaron las versiones resumidas y segmentadas presentadas en las figuras anteriores. Esta estrategia mantiene un equilibrio entre claridad expositiva y exhaustividad documental, y permite que el lector comprenda el comportamiento general del simulador sin perder la posibilidad de consultar posteriormente la representación integral de su flujo interno.

En consecuencia, las figuras incluidas en esta sección deben entenderse como una síntesis funcional del comportamiento de ReactSim Studio, mientras que la versión completa del flujo se presenta como material complementario en el anexo. En particular, el **Anexo A** contiene el código fuente en lenguaje DOT correspondiente al flujograma interno completo del simulador, junto con la indicación de que dicho código debe ser colocado en un editor o visor compatible con Graphviz para generar su representación gráfica integral. De esta manera, el anexo funciona no solo como evidencia documental del flujo interno del software, sino también como un recurso reproducible para la reconstrucción y consulta técnica del diagrama original.

5.4. Estructura Modular del Software

La implementación de ReactSim Studio se desarrolló bajo una organización modular, con el propósito de separar responsabilidades funcionales, facilitar el mantenimiento del código y conservar trazabilidad entre la formulación del modelo, la resolución numérica y la presentación de resultados. Esta estructura modular constituye uno de los resultados más relevantes del desarrollo del software, ya que permitió traducir el problema de simulación a un conjunto de componentes claramente diferenciados, pero articulados entre sí dentro de una misma arquitectura de trabajo.

Como se resume en la **Tabla 5**, el software se organiza en tres bloques principales: el módulo de simulación, el módulo de interfaz gráfica y el bloque de recursos y persistencia. Esta distribución responde a una lógica de diseño en la que el núcleo matemático y numérico se mantiene separado de la capa visual, mientras que los recursos auxiliares, como benchmarks y archivos de caso, se gestionan de forma independiente. A su vez, la relación entre estos bloques se representa de manera general en la **Figura 9**, donde puede apreciarse cómo la información fluye desde la configuración del caso hasta la construcción del resultado final y su visualización en pantalla.

5.4.1. Módulo de Simulación

El módulo de simulación constituye el núcleo técnico del sistema, ya que concentra la representación estructurada del caso, la validación física y dimensional, la construcción de la cinética, la integración numérica y el postproceso de resultados. En este bloque se agrupan los archivos responsables de convertir la formulación teórica del reactor batch ideal homogéneo en una secuencia operativa de cálculo.

Dentro de este módulo, el archivo models.py define las estructuras de datos que representan tanto las entradas del problema como los objetos normalizados y los resultados de simulación. Estas estructuras permiten encapsular de forma

coherente información relativa a especies, cinética, condiciones térmicas, solver y salidas del programa. Como se observa en la **Tabla 5**, este archivo cumple una función estructural, ya que sirve de base para la comunicación entre los distintos componentes del sistema.

Por su parte, validation.py y units.py conforman la capa de validación y normalización. El primero verifica la coherencia física y lógica del caso, mientras que el segundo se encarga de convertir las magnitudes suministradas por el usuario al sistema interno de unidades del simulador. Esta separación mejora la claridad del diseño, ya que distingue explícitamente entre la comprobación del caso y la transformación dimensional requerida para la integración numérica.

El archivo expression.py se ocupa de la compilación y validación segura de expresiones cinéticas personalizadas. Su incorporación amplía la flexibilidad del simulador sin comprometer la estabilidad del sistema, ya que restringe el entorno de evaluación a símbolos físicos explícitos y verifica previamente la validez de la expresión ingresada.

Finalmente, engine.py concentra la lógica de integración numérica y postproceso. En este archivo se construye la función dinámica del reactor, se selecciona el método de resolución, se ejecuta el solver y se calculan las variables derivadas del resultado. En consecuencia, este archivo actúa como el núcleo de ejecución del simulador, al enlazar el caso ya validado con la generación efectiva de perfiles de concentración, temperatura, conversión, calor y diagnósticos numéricos.

5.4.2. Módulo de Persistencia y Casos de Referencia

Además del núcleo de simulación, ReactSim Studio incorpora un bloque funcional orientado a la gestión de casos y recursos auxiliares. Este bloque tiene como finalidad asegurar reproducibilidad, reutilización de configuraciones y disponibilidad de ejemplos de referencia para validación interna.

El archivo case_files.py implementa la lectura y escritura de archivos en formato .reactsim, el cual actúa como formato propio del simulador para almacenar casos completos de simulación. Gracias a este componente, el programa puede guardar configuraciones, recuperar casos previamente construidos y cargar benchmarks integrados sin necesidad de reconstruir manualmente cada escenario desde la interfaz. Esta funcionalidad resulta particularmente importante dentro del enfoque de tesis, ya que facilita la repetición exacta de casos utilizados durante la validación del software.

A este mismo bloque pertenecen los recursos integrados localizados en el directorio resources, dentro del cual se incluyen los benchmarks de referencia del simulador. Como se sintetiza en la **Tabla 5**, estos casos permiten contrastar el comportamiento del motor de cálculo frente a soluciones analíticas o tendencias esperadas del modelo, constituyendo un soporte técnico para la validación presentada en las secciones posteriores del capítulo.

En consecuencia, el módulo de persistencia no solo cumple una función operativa, sino también metodológica, al permitir que los casos utilizados en el análisis sean almacenados, recuperados y comparados de manera consistente.

5.4.3. Módulo de Interfaz Gráfica

La capa de interfaz gráfica agrupa los componentes responsables de la interacción con el usuario y de la presentación visual de resultados. Esta capa fue desarrollada para permitir que la complejidad interna del modelo y del solver pudiera ser utilizada a través de una organización clara de pestañas, formularios, tablas, gráficos y mensajes de estado.

El archivo main.py actúa como punto de entrada de la aplicación, mientras que main_window.py organiza la construcción inicial de la ventana principal. A partir de allí, el subdirectorio ui/window concentra la mayor parte de la lógica de interacción, incluyendo componentes visuales, disposición de la ventana, diálogos auxiliares, campos de entrada, reportes, control del estado de la interfaz, trazabilidad del cálculo y gestión de la simulación.

Como se aprecia en la **Figura 9**, esta capa no realiza directamente los cálculos físico-matemáticos del reactor, sino que funciona como intermediaria entre el usuario y el núcleo de simulación. Su función principal consiste en capturar datos, invocar los procesos de validación y resolución, y traducir la salida interna del motor a una representación visual organizada, comprensible y útil para el análisis técnico del caso.

Esta separación entre interfaz y motor de simulación constituye una decisión importante de diseño, ya que permite modificar elementos visuales o de interacción sin alterar la lógica interna del cálculo. Asimismo, refuerza la mantenibilidad del programa y facilita futuras extensiones funcionales de la herramienta.

5.4.4. Integración entre Módulos

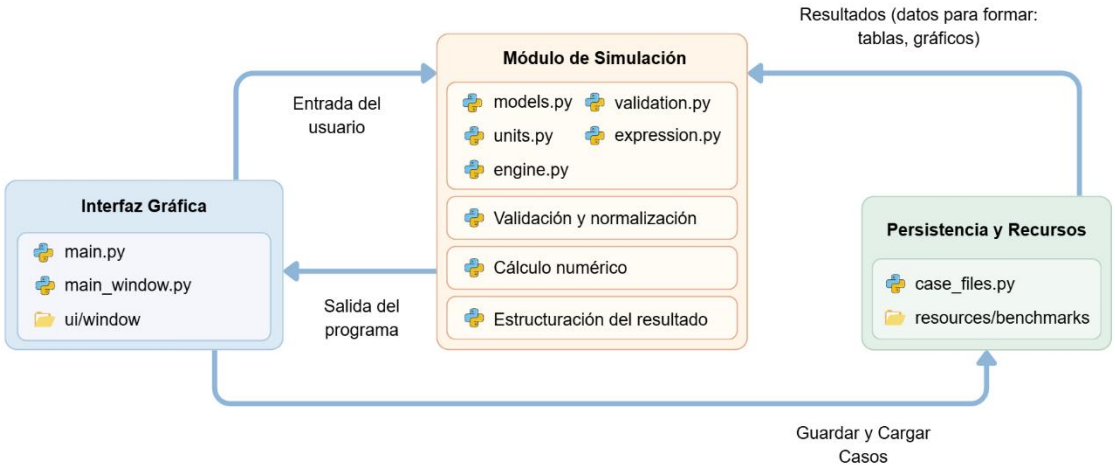
Aunque los módulos descritos anteriormente cumplen funciones diferenciadas, el funcionamiento de ReactSim Studio depende de la integración coordinada entre ellos. El usuario interactúa inicialmente con la interfaz gráfica, la cual construye una representación preliminar del caso; posteriormente, el módulo de simulación valida, normaliza y resuelve el problema, y finalmente la interfaz recibe el resultado estructurado para presentarlo en forma de gráficos, tablas y mensajes de salida. En paralelo, el módulo de persistencia permite guardar o recuperar configuraciones y casos de referencia sin alterar el flujo principal de simulación.

La **Figura 9** sintetiza esta relación entre módulos y muestra que la arquitectura del software no se basa en componentes aislados, sino en una interacción organizada por responsabilidades. De manera complementaria, la **Tabla 5** identifica los archivos principales asociados a cada bloque y resume la función específica de cada uno dentro del sistema.

En conjunto, esta estructura modular permitió desarrollar un simulador coherente, mantenible y técnicamente interpretable, capaz de integrar formulación matemática,

resolución numérica y presentación visual dentro de una única herramienta de trabajo.

Figura 9.
Organización modular de ReactSim Studio y relación funcional entre sus componentes



Nota. Elaboración propia.

Tabla 5.
Organización modular del software y función principal de sus componentes

Módulo	***Archivo o bloque principal***	***Función principal***
Simulación	models.py	Define estructuras de datos para entradas, casos normalizados y resultados.
Simulación	validation.py	Valida coherencia física y lógica del caso de simulación.
Simulación	units.py	Convierte magnitudes a unidades internas del Sistema Internacional.
Simulación	expression.py	Compila y verifica expresiones cinéticas personalizadas en un entorno seguro.
Simulación	engine.py	Ejecuta la integración numérica y el postproceso de resultados.
Persistencia	case_files.py	Guarda y carga casos en formato .reactsim.
Recursos	resources/benchmarks	Proporciona casos de referencia para validación y prueba.

Interfaz gráfica	main.py y main_window.py	Inicializan y organizan la aplicación principal.
Interfaz gráfica	ui/window	Gestiona campos de entrada, resultados, trazas, diálogos y reportes.

Nota. Elaboración propia.

5.5. Interfaz Desarrollada y Funcionalidades Implementadas

La interfaz de ReactSim Studio fue desarrollada para traducir la lógica interna del simulador a un entorno de trabajo claro, ordenado y utilizable en contextos de análisis técnico. Su diseño no se limitó a ofrecer una capa visual sobre el motor numérico, sino que se concibió como un componente funcional del software, encargado de organizar la entrada de datos, orientar la configuración del caso y presentar los resultados de forma interpretable.

En términos operativos, la interfaz reúne en una misma ventana los bloques de definición del caso, la ejecución de la simulación y la consulta de resultados. Esta integración permite que el usuario recorra el flujo completo del análisis sin cambiar de entorno de trabajo y sin perder trazabilidad entre los parámetros suministrados y la respuesta generada por el sistema. La organización general de esta estructura puede observarse en la **Figura 10**.

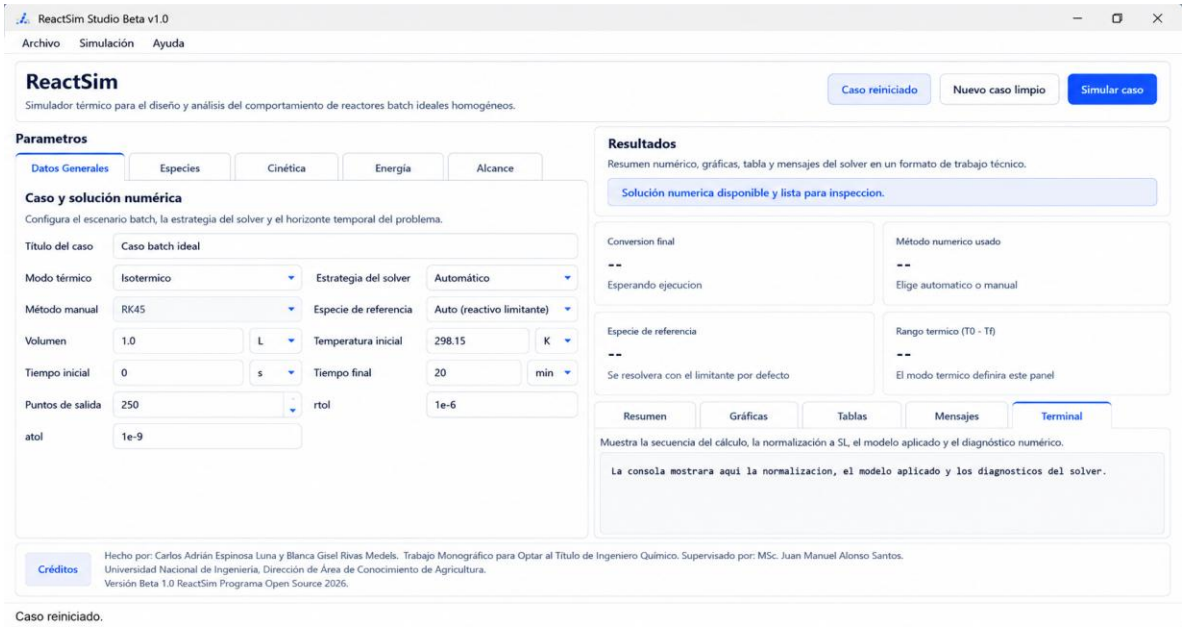
5.5.1. Organización General de la Interfaz

La ventana principal del simulador se encuentra dividida en dos áreas funcionales. La primera corresponde al panel de entrada, donde se definen los parámetros del caso; la segunda corresponde al panel de resultados, donde se muestran los indicadores principales, las gráficas, la tabla de perfiles, los mensajes del sistema y la terminal de cálculo.

En el panel de entrada, la información se agrupa por pestañas, lo que permite separar los distintos bloques del problema sin sobrecargar visualmente la configuración del caso. Esta organización mejora la legibilidad de la interfaz y reduce la probabilidad de errores durante la introducción de datos, ya que cada conjunto de parámetros se presenta en el contexto operativo que le corresponde. Como se aprecia en la **Figura 10**, esta estructura incluye apartados para datos generales, especies, cinética, energía y alcance del modelo.

En el panel derecho se concentran los resultados de simulación. Allí se ubican tarjetas métricas de resumen, un conjunto de pestañas para consultar gráficas, perfiles tabulados, mensajes del sistema y la terminal interna del cálculo. Esta distribución permite alternar entre una visión sintética del caso y una inspección más detallada de la solución numérica obtenida.

Figura 10.
Interfaz principal de ReactSim Studio



Nota. Elaboración propia.

5.5.2. Configuración del Caso y Control de Simulación

La interfaz permite construir el caso de simulación a partir de una secuencia estructurada de entradas. En la pestaña de datos generales se definen el título del caso, el modo térmico, la estrategia del solver, la especie de referencia, el volumen del reactor, la temperatura inicial, el intervalo temporal y las tolerancias numéricas. A partir de esta configuración base, las pestañas restantes habilitan o restringen campos según el tipo de problema seleccionado.

La pestaña de especies permite introducir nombres, coeficientes estequiométricos y concentraciones iniciales. La pestaña de cinética admite tanto leyes de potencia como expresiones personalizadas, junto con la posibilidad de activar la dependencia de Arrhenius. La pestaña de energía concentra los parámetros térmicos requeridos para los casos no isotérmicos, incluyendo calor de reacción, densidad, calor específico, coeficiente global de transferencia, área de intercambio, temperatura de chaqueta y temperatura máxima permisible.

El control operativo del caso se realiza desde la barra superior de la aplicación, donde se disponen acciones para ejecutar la simulación y reiniciar el formulario. Durante este proceso, la interfaz no solo transmite los datos al motor numérico, sino que también informa el estado del caso mediante mensajes de validación, advertencias y confirmaciones de ejecución. De este modo, la interfaz cumple una función activa en la gestión del flujo de trabajo del usuario.

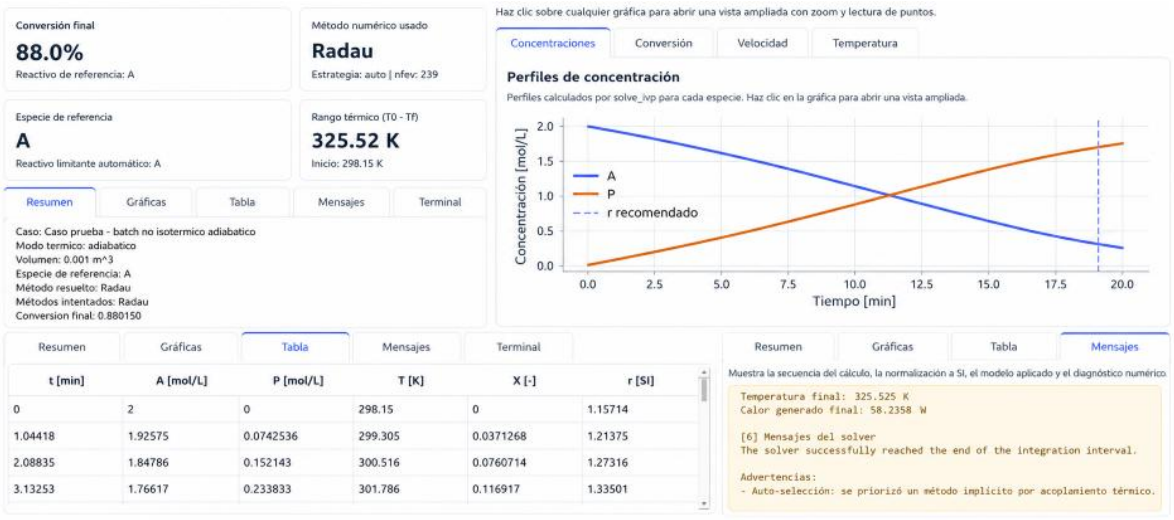
5.5.3. Visualización e Interpretación de Resultados

Una vez ejecutada la simulación, la interfaz presenta la salida del modelo en varios formatos complementarios. El primero corresponde a un resumen textual y numérico del caso, en el que se destacan la conversión final, el método numérico utilizado, la especie de referencia y el comportamiento térmico general del sistema. Esta información permite obtener una lectura rápida del resultado sin necesidad de inspeccionar inmediatamente las curvas o la tabla completa.

El segundo formato corresponde a la visualización gráfica. ReactSim Studio genera perfiles de concentración, conversión, velocidad de reacción y temperatura, permitiendo una lectura dinámica del comportamiento del reactor. Como se observa en la **Figura 11**, estas gráficas se organizan por pestañas y pueden complementarse con referencias visuales, como marcadores del tiempo recomendado de operación o líneas asociadas a la temperatura máxima permisible cuando esta ha sido definida.

El tercer formato corresponde a la tabla de perfiles y a la sección de mensajes del sistema. La tabla permite consultar de forma numérica los valores calculados en cada punto de salida, mientras que el panel de mensajes resume advertencias, observaciones del solver y condiciones relevantes detectadas durante la simulación. A esto se añade la terminal interna, que presenta una traza resumida del cálculo y mejora la transparencia del proceso de resolución.

Figura 11.
Vista de resultados de simulación en ReactSim Studio



Nota. Elaboración propia.

5.5.4. Gestión de Casos, Benchmarks y Exportación

Además de la configuración y visualización del caso, la interfaz incorpora herramientas para la gestión de archivos y la reutilización de configuraciones. El simulador permite guardar y abrir casos en formato .reactsim, lo cual facilita la repetición de simulaciones, la construcción de bibliotecas de ejemplos y la

comparación entre escenarios sin necesidad de reingresar manualmente los parámetros.

Dentro de esta misma lógica, ReactSim Studio integra casos de referencia predefinidos que pueden cargarse como casos de referencia. Esta funcionalidad resulta especialmente útil para fines de validación y documentación, ya que permite reproducir escenarios representativos y contrastar el comportamiento del simulador con soluciones analíticas o respuestas esperadas del modelo.

La interfaz también dispone de mecanismos de exportación de resultados, orientados a preservar trazabilidad del caso analizado y a facilitar su incorporación en reportes técnicos o documentos de tesis. En conjunto, estas funciones convierten a la interfaz en algo más que una capa de interacción: la posicionan como una parte activa de la experiencia de simulación, al integrar configuración, ejecución, consulta y almacenamiento dentro de un mismo entorno.

5.6. Validación Mediante Soluciones Analíticas y Resultados de ReactSim Studio

La validación de ReactSim Studio se realizó mediante la comparación de los resultados generados por el simulador con soluciones analíticas y semianalíticas de referencia, seleccionadas por su correspondencia con el alcance funcional del programa y por su utilidad para verificar la correcta implementación del modelo batch ideal homogéneo. Este enfoque permitió evaluar simultáneamente la consistencia del motor numérico, la coherencia del postproceso y la fidelidad del software respecto a resultados teóricos esperados.

Con el fin de mantener esta sección centrada en el análisis comparativo, los desarrollos matemáticos completos no se presentan aquí de forma detallada. Dichos procedimientos se reúnen en el **Anexo B**, donde se incluyen las deducciones analíticas paso a paso de los casos isotérmicos, los benchmarks no isotérmicos adaptados y el desarrollo semianalítico del Ejemplo 9-1 de Fogler. En el presente capítulo se muestran únicamente los resultados consolidados y su interpretación.

La estrategia de validación se estructuró en tres niveles. En primer lugar, se emplearon casos isotérmicos con solución analítica cerrada, correspondientes a cinéticas de orden cero, primero y segundo. En segundo lugar, se construyeron dos benchmarks no isotérmicos con solución analítica cerrada, uno adiabático y otro con chaqueta, a partir de la formulación general del reactor batch no isotérmico presentada por Fogler. Finalmente, se incorporó como caso de referencia bibliográfica el **Ejemplo 9-1** de Fogler, resuelto mediante un procedimiento semianalítico y comparado con el resultado obtenido en ReactSim Studio.

5.6.1. Casos Isotérmicos con Solución Analítica Cerrada

Como primer bloque de validación se seleccionaron tres casos batch isotérmicos, debido a que estos permiten contrastar directamente la respuesta numérica del simulador con expresiones analíticas exactas derivadas de la ecuación de diseño del reactor por lotes ideal. Tales casos constituyen una referencia clásica en

Ingeniería de las Reacciones Químicas y permiten verificar, sin interferencia del balance de energía, la correcta implementación del balance de materia, la ley de velocidad y la conversión de unidades.

Los desarrollos completos de estos tres casos se presentan en el **Anexo B**. En todos ellos se utilizó una única reacción irreversible $A \rightarrow P$, volumen y temperatura constantes, de forma consistente con el modelo implementado en ReactSim Studio. Los resultados comparativos obtenidos se resumen en la **Tabla 6**.

Tabla 6.
Comparación entre soluciones analíticas isotérmicas y resultados de ReactSim Studio

Caso	***Parámetros principales***	***Resultado analítico***	***Resultado ReactSim Studio***	***Error relativo máximo***
Batch isotérmico, orden cero	$C_{A0} = 2.0 \frac{mol}{L}$ $k = 0.05 \frac{mol}{L\,min}$ $t_f = 20\,min$	$C_A = 1.00 \frac{mol}{L}$ $X = 0.50$	$C_A = 1.00 \frac{mol}{L}$ $X = 0.50$	$<10^{-12}\%$
Batch isotérmico, primer orden	$C_{A0} = 2.0 \frac{mol}{L}$ $k = 0.15\,min^{-1}$ $t_f = 25\,min$	$C_A = 0.0470 \frac{mol}{L}$ $X = 0.97648$	$C_A = 0.0470 \frac{mol}{L}$ $X = 0.97648$	$1.16 \times 10^{-4}\%$
Batch isotérmico, segundo orden	$C_{A0} = 1.0 \frac{mol}{L}$ $k = 0.20 \frac{L}{mol\,min}$ $t_f = 10\,min$	$C_A = 0.3333 \frac{mol}{L}$ $X = 0.6667$	$C_A = 0.3333 \frac{mol}{L}$ $X = 0.6667$	$3.66 \times 10^{-7}\%$

Nota. Elaboración propia.

Como se aprecia en la **Tabla 6**, la concordancia entre las soluciones analíticas y las respuestas del simulador es prácticamente exacta en los tres casos isotérmicos. Esto confirma que el motor del programa reproduce correctamente la ecuación de diseño batch, la implementación de la cinética de ley de potencia y el tratamiento interno de unidades. Asimismo, esta primera validación establece una base sólida antes de examinar escenarios no isotérmicos de mayor complejidad.

5.6.2. Casos no Isotérmicos con Solución Analítica Adaptada a Partir de Fogler

Con el fin de extender la validación a los modos no isotérmicos del simulador, se construyeron dos casos de referencia con solución analítica cerrada a partir de la formulación general del balance de energía batch presentada por Fogler para reactores por lotes no isotérmicos. En ambos casos se adoptó una reacción irreversible de primer orden con constante cinética k independiente de la temperatura, lo que permitió conservar la estructura física del problema y, al mismo tiempo, obtener expresiones analíticas explícitas para la evolución de la concentración, la conversión y la temperatura.

El primer caso de referencia corresponde a un reactor batch no isotérmico adiabático, para el cual la solución analítica conduce a una relación lineal entre temperatura y conversión. El segundo corresponde a un reactor batch no isotérmico con chaqueta a temperatura constante, donde el balance de energía incorpora el término de transferencia $UA(T - T_c)$ y conduce a una ecuación diferencial lineal de primer orden con solución cerrada. Ambos desarrollos se presentan detalladamente en el **Anexo B**.

Los parámetros usados en estos dos casos fueron consistentes con el alcance del simulador y con un escenario representativo de validación:

- $C_{A0} = 2.0 \frac{mol}{L}$
- $k = 0.15\, min^{-1}$
- $T_0 = 298.15\ K$
- $\Delta H_r = -65000 \frac{J}{mol}$
- $\rho = 1000 \frac{kg}{m^3}$
- $C_p = 4180 \frac{J}{kgK}$
- $V = 1.0L$

Para el caso con chaqueta se añadieron:

- $U = 250 \frac{W}{m^2 K}$
- $A = 1.80\ m^2$
- $T_c = 290.0\ K$

Los resultados comparativos obtenidos para ambos casos de referencia se resumen en la **Tabla 7**.

Tabla 7.
Comparación entre soluciones analíticas no isotérmicas adaptadas y resultados en ReactSim Studio

Caso	*Magnitud comparada*	*Solución analítica*	*ReactSim Studio*	*Error relativo*
Batch no isotérmico adiabático	$C_A(20\ min)$	$0.09957 \frac{mol}{L}$	$0.09957 \frac{mol}{L}$	$9.01 \times 10^{-8}\ \%$
Batch no isotérmico adiabático	$\mathrm{X}(20\ min)$	0.95021	0.95021	$4.72 \times 10^{-9}\%$

Batch no isotérmico adiabático	$T(20\ min)$	$327.70\ K$	$327.702\ K$	$4.26 \times 10^{-10}\%$
Batch no isotérmico con chaqueta	$C_A(20\ min)$	$0.09957\ \frac{mol}{L}$	$0.09957\ \frac{mol}{L}$	$8.75 \times 10^{-8}\%$
Batch no isotérmico con chaqueta	$\mathrm{X}(20\ min)$	0.95021	0.95021	$4.58 \times 10^{-9}\%$
Batch no isotérmico con chaqueta	$T(20\ min)$	$290.04\ K$	$290.037\ K$	$1.11 \times 10^{-11}\%$

Nota. Elaboración propia con base en la formulación general del reactor batch no isotérmico desarrollada por Fogler.

La **Tabla 7** muestra concordancia prácticamente exacta entre las soluciones analíticas y las respuestas numéricas de ReactSim Studio. Este resultado es especialmente importante porque valida de manera explícita los dos modos no isotérmicos del simulador: el caso adiabático, donde la temperatura evoluciona exclusivamente por el calor de reacción, y el caso con chaqueta, donde la temperatura está influida además por la remoción térmica impuesta por el término $UA(T - T_c)$. En consecuencia, esta comparación confirma que el simulador implementa correctamente el balance de energía en ambos escenarios.

5.6.3. Caso de Referencia Bibliográfica: Ejemplo 9-1 Fogler

Como validación complementaria se empleó el **Ejemplo 9-1** de Fogler, correspondiente a un reactor por lotes adiabático con dependencia térmica de la cinética. A diferencia de los casos anteriores, este caso de referencia no conduce a una solución cerrada simple, ya que la constante de velocidad depende de la temperatura y la temperatura depende a su vez de la conversión. Por ello, su tratamiento es de carácter semianalítico: parte de una relación analítica entre temperatura y conversión, pero requiere la evaluación numérica de la integral de diseño.

El desarrollo completo de este caso se presenta en el **Anexo B**, incluyendo la reconstrucción de la relación $T(X)$, la expresión de $k(T)$ y la aplicación de la regla de Simpson utilizada por Fogler. En esta sección se muestran únicamente los resultados comparativos finales. La **Tabla 8** resume la comparación entre la solución reportada por Fogler y la obtenida en ReactSim Studio mediante el archivo de caso equivalente cargado en formato .reactsim.

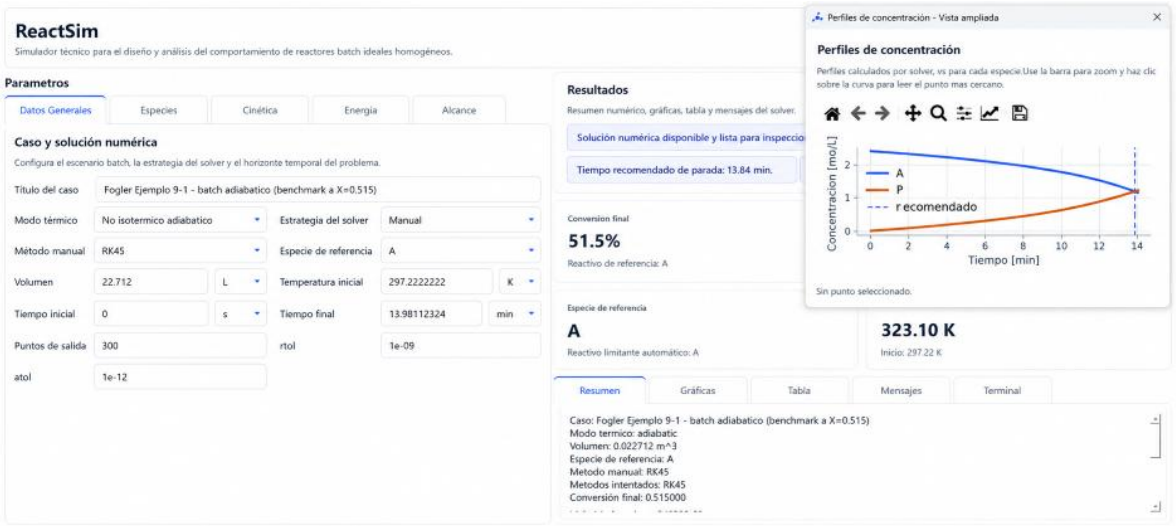
Tabla 8.
Comparación entre la solución de Fogler y el resultado de ReactSim Studio para el Ejemplo 9-1.

Magnitud	*Referencia de Fogler*	*ReactSim Studio*	*Comentario*
Conversión objetivo, X	0.515	0.5147	Diferencia mínima por discretización e interpolación
Tiempo, min	13.9	13.98	Concordancia alta
Temperatura final, °F	122.0	121.91	Concordancia alta

Nota. Elaboración propia con base en Fogler.

La concordancia observada en la **Tabla 8** confirma que el simulador reproduce adecuadamente el comportamiento esperado del benchmark no isotérmico tomado del texto de Fogler. Las pequeñas diferencias registradas se explican por el hecho de que la solución manual del libro se basa en tabulación y aplicación de la regla de Simpson, mientras que ReactSim Studio resuelve el problema mediante integración numérica directa con control de error.

Figura 12.
Resultado del caso Fogler Ejemplo 9-1 en ReactSim Studio7



Nota. Elaboración propia.

5.6.4. Contrastación Externa de Resultados con Cantera

Con el propósito de complementar la validación técnica de ReactSim Studio, se realizó una contrastación externa con Cantera 3.2.0. Esta comparación permitió evaluar la consistencia de los resultados obtenidos por el simulador frente a una

herramienta abierta de simulación química ampliamente utilizada para problemas de cinética, termodinámica y reactores. La comparación se planteó mediante casos equivalentes, debido a que Cantera y ReactSim Studio no poseen el mismo alcance físico ni la misma estructura de modelado.

Para asegurar una comparación controlada, los casos fueron configurados en Cantera mediante pseudoespecies A y P, una reacción irreversible $A \rightarrow P$ y parámetros cinéticos equivalentes a los utilizados en ReactSim Studio. En el caso isotérmico, se desactivó la ecuación de energía en Cantera para mantener temperatura constante. En el caso adiabático, se incorporó una configuración energética equivalente para reproducir el acoplamiento térmico-cinético del caso de referencia. Este procedimiento permitió comparar concentración final, conversión y temperatura bajo condiciones equivalentes de simulación.

El primer caso correspondió a un reactor batch isotérmico de segundo orden. En ReactSim Studio, el caso se definió con $C_{A0} = 1\,\frac{mol}{L}$, $C_{P0} = 0$, $V = 1L$, $T = 298.15K$, $t_f = 600s$, $k = 3.33333 \times 10^{-6}\,\frac{m^3}{mol \cdot s}$. El reporte del simulador indicó una conversión final de 0.666667, concentración final de A de $0.333333\,\frac{mol}{L}$ y concentración de P de $0.666667\,\frac{mol}{L}$.

Tabla 9.
Comparación Externa entre ReactSim Studio y Cantera para el Caso Isotérmico de Segundo Orden

Variable	*ReactSim Studio*	*Cantera*	*Error relativo*
C_A *final*	0.333333 mol/L	0.333334 mol/L	0.000168 %
C_P *final*	0.666667 mol/L	0.666666 mol/L	0.000084 %
X_A *final*	0.666667	0.666666	0.000084 %
T_f	298.15 K	298.15 K	0.000000 %

Nota. Elaboración propia.

Los resultados de la **Tabla 9** muestran una concordancia numérica prácticamente total entre ReactSim Studio y Cantera para el caso isotérmico de segundo orden. Las diferencias observadas se mantienen por debajo de 0.001 %, lo cual indica que, bajo las condiciones equivalentes definidas para la comparación, ambos entornos reproducen de manera consistente el comportamiento esperado del balance de materia y de la ley cinética de segundo orden. Además, al tratarse de un caso con solución analítica cerrada, la comparación también respalda la correcta implementación del modelo isotérmico en ReactSim Studio.

El segundo caso correspondió a un reactor batch adiabático basado en el Ejemplo 9-1 de Fogler. En ReactSim Studio, el caso fue configurado con $C_{A0} = 2.380239521\,\frac{mol}{L}$, $C_{P0} = 0, V = 22.712\,L$, $T_0 = 297.222\,K$, $t_f = 838.867\,s$, $k_0 = 4.7077 \times 10^9$, $E_a = 75319.8\,\frac{J}{mol}$ y una cinética de primer orden con dependencia de Arrhenius. El reporte indicó una concentración final de A de $1.15442\,\frac{mol}{L}$ y concentración final de P de $1.22582\,\frac{mol}{L}$.

Tabla 10.
Comparación Externa entre ReactSim Studio y Cantera para el Caso Adiabático Basado en Fogler

Variable	*ReactSim Studio*	*Cantera*	*Error relativo*
C_A *final*	1.154420 mol/L	1.154467 mol/L	0.004101 %
C_P *final*	1.225820 mol/L	1.225772 mol/L	0.003901 %
X_A *final*	0.515000	0.514979	0.004175 %
T_f	323.101 K	323.099921 K	0.000334 %

Nota. Elaboración propia.

Los resultados de la **Tabla 10** evidencian una alta concordancia entre ReactSim Studio y Cantera para el caso adiabático evaluado. El error relativo máximo fue inferior a 0.005%, lo que indica que la implementación del balance de materia, la ecuación de Arrhenius y el balance de energía en ReactSim Studio produce una respuesta numérica coherente frente a una herramienta externa de simulación. La diferencia observada se atribuye principalmente a redondeos, al uso de pseudoespecies en Cantera y a las diferencias internas en la representación del sistema termodinámico.

En conjunto, ambos casos muestran que ReactSim Studio reproduce de forma consistente la evolución de concentración, conversión y temperatura frente a casos equivalentes configurados en Cantera. Esta contrastación externa complementa la validación realizada mediante soluciones analíticas, soluciones semianalíticas y problemas de referencia tomados de Fogler, fortaleciendo la evidencia de que el simulador desarrollado implementa correctamente la lógica físico-matemática definida para reactores batch ideales homogéneos. No obstante, esta comparación debe interpretarse dentro del alcance del presente trabajo, ya que no pretende demostrar equivalencia funcional total entre ReactSim Studio y Cantera, sino verificar la coherencia de los resultados bajo condiciones controladas de simulación.

5.7. Desempeño Numérico del Simulador

Además de la validación frente a soluciones de referencia, resulta necesario examinar el comportamiento numérico de ReactSim Studio desde la perspectiva del proceso de integración. Este análisis permite verificar que el simulador no solo reproduce resultados físicamente coherentes, sino que también utiliza de forma estable y consistente los métodos de resolución implementados en su motor numérico.

El desempeño numérico se evaluó considerando tres aspectos: la respuesta de los métodos RK45, Radau y BDF en casos representativos, el criterio de selección automática del solver y la sensibilidad de la solución frente al ajuste de tolerancias numéricas. Para ello se utilizaron casos de referencia integrados del propio simulador, correspondientes a un caso isotérmico de primer orden, un caso no isotérmico adiabático y un caso no isotérmico con chaqueta.

Conviene señalar que las diferencias entre RK45, Radau y BDF no se deben a que resuelvan modelos distintos, sino al tipo de esquema numérico empleado por cada método. RK45 es un integrador explícito y, por ello, suele ser conveniente en problemas no rígidos o moderadamente suaves. En cambio, Radau y BDF son métodos implícitos, lo que implica evaluaciones adicionales del jacobiano y factorizaciones internas, pero les confiere mayor robustez en problemas con acoplamiento térmico o indicios de rigidez numérica. En consecuencia, las diferencias entre métodos deben interpretarse principalmente en términos de costo computacional y estabilidad, y no como discrepancias en la validez física de la solución.

5.7.1. Comparación Entre los Métodos de Integración

La primera comparación se realizó ejecutando los mismos casos con los tres métodos disponibles en ReactSim Studio: RK45, Radau y BDF. Para cada simulación se revisaron los diagnósticos internos del solver, en particular el número de evaluaciones de la función (nfev), las evaluaciones del jacobiano (njev) y las factorizaciones internas (nlu) cuando estas aplicaban. También se verificó la concordancia de los resultados finales en términos de conversión y temperatura.

Los resultados se resumen en la **Tabla 11** y se representan de forma comparativa en la **Figura 13**. En el caso isotérmico, RK45 mostró un comportamiento eficiente y estable, lo cual es consistente con la naturaleza no rígida del problema. En los casos no isotérmicos, Radau y BDF mantuvieron resultados finales prácticamente equivalentes, aunque con un perfil numérico más apropiado para sistemas con acoplamiento energético.

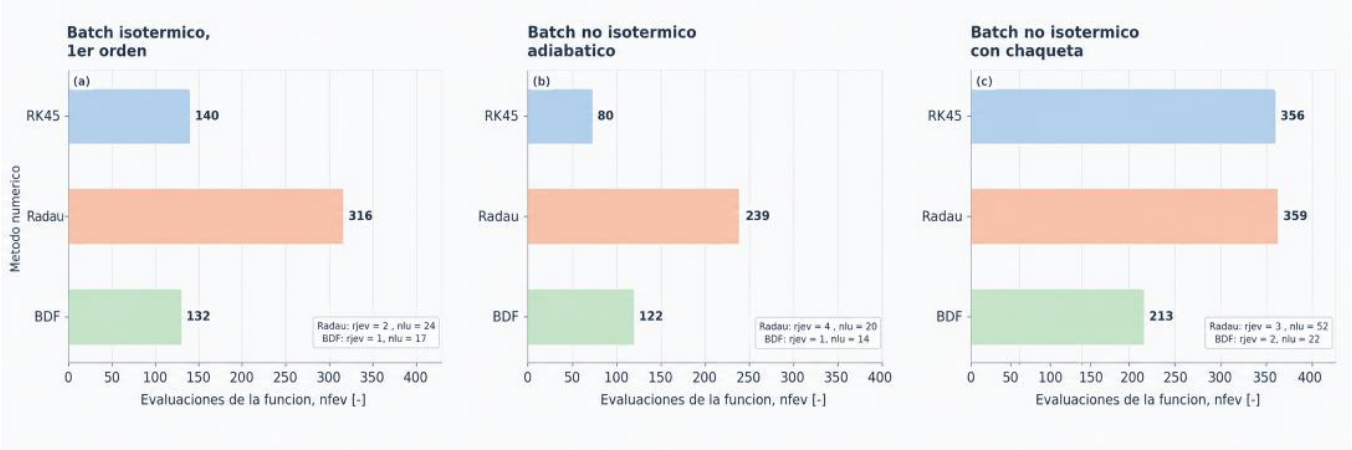
La **Figura 13** permite apreciar con mayor claridad que la diferencia principal entre métodos no radica en la solución física obtenida, sino en el esfuerzo computacional requerido para alcanzarla. En particular, RK45 evita el cálculo de jacobianos y factorizaciones, mientras que Radau y BDF incorporan estos pasos como parte de su esquema implícito. Por ello, los valores de njev y nlu aparecen únicamente en los métodos implícitos y constituyen un indicador de su costo interno de resolución.

Tabla 11.
Comparación de métodos numéricos implementados en ReactSim Studio

Caso	*Método*	*nfev*	*njev*	*nlu*	X_f	T_f
Batch isotérmico, 1er orden	RK45	140	0	0	0.97648	298.15 K
Batch isotérmico, 1er orden	Radau	316	2	24	0.97648	298.15 K
Batch isotérmico, 1er orden	BDF	132	1	17	0.97648	298.15 K
Batch no isotérmico adiabático	RK45	80	0	0	0.88015	325.52 K
Batch no isotérmico adiabático	Radau	239	4	28	0.88015	325.52 K
Batch no isotérmico adiabático	BDF	122	1	14	0.88015	325.52 K
Batch no isotérmico con chaqueta	RK45	356	0	0	0.31377	290.06 K
Batch no isotérmico con chaqueta	Radau	359	3	54	0.31377	290.06 K
Batch no isotérmico con chaqueta	BDF	213	2	22	0.31377	290.06 K

Nota. Elaboración propia.

Figura 13.
Comparación del costo numérico de RK45, Radau y BDF en casos representativos de ReactSim Studio



Nota. Elaboración propia.

La **Tabla 11** muestra que los tres métodos implementados en ReactSim Studio convergen correctamente en los casos estudiados y producen resultados finales prácticamente idénticos. En consecuencia, la diferencia entre ellos no radica en la validez del resultado físico, sino en el costo computacional y en la conveniencia de su uso según la naturaleza del problema.

5.7.2. Comportamiento de la Selección Automática del Solver

Uno de los elementos distintivos del simulador es la incorporación de una estrategia de selección automática del solver. Esta lógica busca asignar un método numérico

apropiado según la naturaleza del caso antes de iniciar la simulación, manteniendo RK45 como opción preferente en problemas suaves y priorizando métodos implícitos en escenarios con acoplamiento térmico o mayor sensibilidad numérica.

La respuesta de esta heurística se verificó utilizando los mismos tres casos de referencia anteriores. Los resultados se presentan en la **Tabla 12**.

Tabla 12.
Comportamiento de la selección automática del solver en ReactSim Studio

Caso	***Método seleccionado automáticamente***	***Métodos intentados***	***Observación generada por el simulador***
Batch isotérmico, 1er orden	RK45	RK45	Se priorizó RK45 por tratarse de un caso no rígido
Batch no isotérmico adiabático	Radau	Radau	Se priorizó un método implícito por acoplamiento térmico
Batch no isotérmico con chaqueta	Radau	Radau	Se priorizó un método implícito por acoplamiento térmico

Nota. Elaboración propia.

Como se aprecia en la **Tabla 12**, la selección automática se comportó de forma coherente con la naturaleza de los casos evaluados. En el problema isotérmico se mantuvo RK45 como opción principal, mientras que en los dos problemas no isotérmicos se priorizó Radau, lo cual es consistente con la lógica del motor y con la conveniencia de emplear métodos implícitos en sistemas con balance de energía acoplado.

5.7.3. Sensibilidad a Tolerancias Numéricas

El tercer aspecto evaluado fue la sensibilidad de la solución frente al ajuste de las tolerancias rtol y atol, tomando como referencia el benchmark no isotérmico adiabático resuelto con Radau. El objetivo de esta prueba fue verificar que, al modificar el criterio de error admitido por el integrador, la solución convergiera hacia perfiles equivalentes y que las diferencias principales se manifestaran en el costo numérico de la simulación.

Para interpretar correctamente esta comparación, debe recordarse que valores más grandes de rtol y atol relajan el criterio de error local permitido por el integrador, mientras que valores más pequeños lo hacen más estricto. En consecuencia, al aumentar rtol y atol se espera una disminución del costo computacional, aunque

con menor exigencia numérica; por el contrario, al reducir estas tolerancias se incrementa el esfuerzo de integración, pero la solución tiende a estabilizarse hacia un valor convergente.

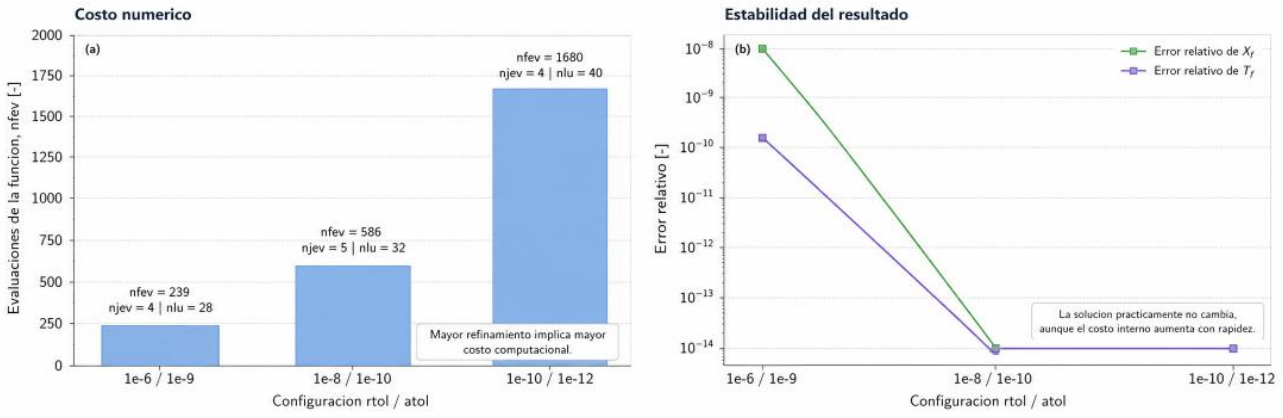
Los resultados se resumen en la **Tabla 13** y se complementan con la **Figura 14**. Como se observa, al refinar las tolerancias, es decir, al reducir rtol y atol, aumenta el número de evaluaciones y factorizaciones internas del solver, mientras que las variaciones en la conversión final y en la temperatura final permanecen prácticamente despreciables. Esto indica que la solución presenta convergencia práctica y que el incremento del esfuerzo numérico no se traduce en cambios relevantes a escala de ingeniería para el caso analizado.

Tabla 13.
Sensibilidad del caso no isotérmico adiabático frente a distintas tolerancias numéricas

rtol	***atol***	***nfev***	***njev***	***nlu***	X_f	$T_f\ (K)$
10^{-6}	10^{-9}	239	4	28	0.88015010	325.52308910
10^{-8}	10^{-10}	586	5	32	0.88015009	325.52308904
10^{-10}	10^{-12}	1680	4	40	0.88015009	325.52308904

Nota. Elaboración propia.

Figura 14.
Efecto del ajuste de rtol y atol sobre el costo numérico y la estabilidad del resultado



Nota. Elaboración propia.

La **Tabla 13** evidencia que el refinamiento de tolerancias aumenta el costo numérico del problema, pero no produce cambios significativos en la solución final. La **Figura 14** permite visualizar que la sensibilidad del resultado es baja frente al ajuste de

tolerancias dentro del rango estudiado. En consecuencia, las tolerancias por defecto implementadas en ReactSim Studio resultan suficientes para obtener resultados confiables dentro del nivel de precisión requerido en esta tesis, sin introducir un costo computacional innecesario.

5.7.4. Discusión del Desempeño Numérico

En conjunto, el análisis realizado permite afirmar que ReactSim Studio presenta un desempeño numérico consistente con el tipo de problemas que resuelve. Los métodos implementados convergen adecuadamente en los casos de referencia considerados, la selección automática del solver responde de forma coherente a la naturaleza térmica del caso y la solución mantiene estabilidad frente al refinamiento de tolerancias.

Desde una perspectiva práctica, estos resultados indican que el simulador no depende de un único método de integración para producir resultados aceptables, sino que dispone de una base numérica suficientemente robusta para trabajar tanto en escenarios isotérmicos como en casos con acoplamiento energético. Al mismo tiempo, el registro de diagnósticos internos aporta transparencia adicional al proceso de simulación, permitiendo interpretar no solo la respuesta física del reactor, sino también el comportamiento del solver utilizado para resolverlo.

5.8. Discusión General de Resultados y Limitaciones Actuales

Los resultados presentados en este capítulo evidencian que ReactSim Studio permitió integrar de manera coherente la formulación clásica de reactores batch ideales homogéneos con una implementación computacional funcional, numéricamente estable y visualmente interpretable. La relación entre el algoritmo interno, la estructura modular, la interfaz gráfica y los casos de validación mostró que el software no opera como una herramienta cerrada de cálculo, sino como una plataforma trazable, en la cual cada resultado puede relacionarse con el modelo matemático implementado, los parámetros de entrada, el método numérico utilizado y las salidas generadas por el sistema.

Desde el punto de vista funcional, el desarrollo del simulador permitió consolidar en una misma aplicación la definición del caso de simulación, la validación física y dimensional de entradas, la normalización al Sistema Internacional, la resolución de balances de materia y energía, la selección del solver, el cálculo de variables derivadas y la presentación organizada de resultados. Esta integración constituye uno de los principales aportes del trabajo, ya que demuestra la viabilidad de trasladar un modelo batch ideal homogéneo a una herramienta computacional operativa sin perder correspondencia con los fundamentos teóricos expuestos en el marco teórico.

La validación mediante soluciones analíticas y semianalíticas mostró una alta concordancia entre los resultados de referencia y los obtenidos con ReactSim Studio. En los casos isotérmicos, la coincidencia con las expresiones cerradas para cinéticas de orden cero, primer orden y segundo orden confirmó la correcta implementación del balance de materia y de las leyes cinéticas de tipo potencia. En

los casos no isotérmicos adaptados, la concordancia con soluciones derivadas del balance de energía permitió verificar que el simulador reproduce adecuadamente tanto el comportamiento adiabático como el efecto del término de transferencia de calor con chaqueta. Asimismo, el contraste con el Ejemplo 9-1 de Fogler permitió extender la validación a un escenario con dependencia térmica de la cinética mediante la ecuación de Arrhenius, reforzando la consistencia del simulador frente a una referencia bibliográfica clásica de la Ingeniería de las Reacciones Químicas.

La contrastación externa con Cantera complementó el proceso de validación al comparar los resultados de ReactSim Studio con una herramienta abierta reconocida en el ámbito de la simulación química. Esta comparación se realizó mediante casos equivalentes, configurados de forma controlada para reproducir el mismo comportamiento dinámico evaluado en ReactSim Studio. En el caso isotérmico de segundo orden, los resultados de concentración final, conversión y temperatura presentaron errores relativos inferiores a 0.001 %, lo cual respalda la implementación del balance de materia y de la cinética de segundo orden. En el caso adiabático basado en Fogler, la comparación también mostró diferencias reducidas en concentración, conversión y temperatura final, con errores relativos inferiores a 0.005 %, lo que respalda la implementación del acoplamiento entre cinética, balance de materia y balance de energía.

No obstante, la comparación con Cantera debe interpretarse dentro del alcance metodológico del trabajo. Cantera es una herramienta general de simulación química, orientada a problemas de cinética, termodinámica, transporte, mecanismos detallados, múltiples especies y redes de reactores. ReactSim Studio, en cambio, fue diseñado como una herramienta especializada para reactores batch ideales homogéneos en fase líquida, de volumen constante y con una sola reacción irreversible. Por tanto, la comparación no busca establecer equivalencia total entre ambos programas, sino comprobar que, bajo condiciones equivalentes, ReactSim Studio produce respuestas coherentes frente a un entorno externo independiente.

Desde esta perspectiva, Cantera ofrece mayor generalidad, mientras que ReactSim Studio ofrece mayor especificidad y trazabilidad para el modelo abordado en esta investigación. Esta diferencia no representa una desventaja del simulador desarrollado, sino una delimitación técnica del alcance propuesto. ReactSim Studio no pretende reemplazar a una suite general de simulación química, sino proporcionar una herramienta clara, abierta y técnicamente estructurada para estudiar un tipo específico de reactor desde los fundamentos de la Ingeniería de las Reacciones Químicas. En ese sentido, su valor radica en que el usuario puede identificar de forma directa la relación entre balance de materia, balance de energía, ley cinética, selección del método numérico y resultados obtenidos.

En el plano numérico, los resultados también fueron satisfactorios. La comparación entre RK45, Radau y BDF mostró que el simulador genera respuestas consistentes con diferentes métodos de integración, mientras que el análisis de tolerancias evidenció convergencia práctica de la solución al refinar los criterios de error. Además, la estrategia de selección automática del solver mostró un comportamiento coherente con la naturaleza de los casos evaluados, al priorizar métodos explícitos

en escenarios numéricamente suaves y métodos implícitos en problemas con mayor sensibilidad o acoplamiento térmico. En conjunto, estos resultados indican que el núcleo numérico de ReactSim Studio es suficientemente robusto para los problemas contemplados en la versión desarrollada.

A pesar de los resultados obtenidos, el trabajo presenta limitaciones que deben reconocerse con claridad. ReactSim Studio fue construido bajo un alcance deliberadamente acotado, restringido a reactores batch ideales homogéneos, en fase líquida, de volumen constante y con una sola reacción química irreversible. En consecuencia, el simulador no incorpora en su estado actual sistemas de volumen variable, fase gaseosa, múltiples reacciones simultáneas, reacciones reversibles, fenómenos de transferencia de masa entre fases ni propiedades térmicas dependientes de la temperatura, más allá de la dependencia cinética representada mediante la ecuación de Arrhenius. Estas limitaciones no constituyen fallas del sistema, sino decisiones de delimitación necesarias para mantener coherencia entre los objetivos, el modelo matemático y la viabilidad de implementación.

Asimismo, la validación desarrollada en esta investigación tuvo un carácter técnico, matemático y computacional. Los resultados obtenidos respaldan la consistencia del simulador frente a soluciones analíticas, casos bibliográficos y comparaciones externas equivalentes, pero no deben interpretarse como una validación experimental de un reactor industrial real. De igual forma, el trabajo no incluye todavía estudios formales de usabilidad, evaluación pedagógica con usuarios ni análisis de desempeño en contextos de enseñanza, aspectos que podrían abordarse en investigaciones posteriores.

A pesar de estas restricciones, los resultados alcanzados muestran que ReactSim Studio constituye una base técnica estable y extensible. La organización modular del software, la separación entre interfaz gráfica y motor de simulación, la gestión de archivos de caso, la incorporación de benchmarks y la posibilidad de emplear configuraciones cinéticas definidas por el usuario permiten proyectar futuras ampliaciones sin necesidad de reestructurar completamente el sistema. Entre las extensiones naturales del trabajo se encuentran la incorporación de reacciones reversibles, múltiples reacciones, propiedades dependientes de la temperatura, nuevos modos de operación térmica, comparación con más herramientas externas y ampliación de la biblioteca de casos de validación.

En síntesis, la discusión de resultados permite afirmar que ReactSim Studio cumple de manera satisfactoria con el alcance propuesto para esta investigación. El simulador resolvió los casos evaluados de forma coherente con la teoría de reactores batch ideales homogéneos, mostró estabilidad numérica bajo distintos métodos de integración, reprodujo soluciones de referencia y presentó concordancia frente a casos equivalentes configurados en Cantera. De esta manera, el trabajo deja establecida una plataforma abierta, trazable y técnicamente consistente para el análisis, validación continua y ampliación futura de modelos de reactores batch en el ámbito de la Ingeniería Química.

VI. Conclusiones

La presente investigación permitió desarrollar ReactSim Studio como un software de código abierto orientado a la simulación de reactores batch ideales homogéneos en fase líquida, bajo condiciones isotérmicas y no isotérmicas, de volumen constante y con una sola reacción química irreversible. La herramienta integra modelos de cinética química, balances de materia y balances de energía dentro de una plataforma computacional funcional, permitiendo representar la evolución temporal de variables como concentración, conversión, temperatura, velocidad de reacción, calor generado y calor removido cuando corresponde. De esta manera, el trabajo logró vincular la teoría clásica de la Ingeniería de las Reacciones Químicas con una implementación computacional trazable, reproducible y técnicamente consistente.

El modelo matemático formulado permitió describir el comportamiento dinámico del reactor batch ideal homogéneo a partir de los principios de conservación de la materia y la energía. Para el régimen isotérmico, el sistema se representó mediante balances de materia acoplados a una ley cinética de tipo potencia; mientras que, para el régimen no isotérmico, se incorporó el balance de energía y la dependencia de la constante cinética con la temperatura mediante la ecuación de Arrhenius. Esta formulación permitió representar el consumo del reactivo, la formación del producto, el avance de la conversión y el efecto térmico asociado a la reacción química dentro del alcance definido para el simulador.

La implementación numérica de los modelos se realizó mediante la transformación del sistema físico-matemático en un problema de valor inicial compuesto por ecuaciones diferenciales ordinarias. El motor de simulación fue desarrollado en Python utilizando la función solve_ivp de SciPy, con soporte para los métodos RK45, Radau y BDF. La incorporación de tolerancias relativas y absolutas, selección manual o automática del integrador, normalización de unidades al Sistema Internacional y diagnósticos numéricos permitió obtener soluciones estables y coherentes para los casos evaluados, tanto en escenarios no rígidos como en problemas con posible acoplamiento térmico-cinético.

La herramienta computacional desarrollada permitió integrar el modelo matemático y el motor numérico en una interfaz gráfica funcional, orientada a la configuración estructurada de casos de simulación. ReactSim Studio permite ingresar parámetros cinéticos, térmicos y operativos, ejecutar simulaciones, visualizar perfiles de concentración, conversión y temperatura, gestionar archivos de caso y exportar resultados. Esta estructura permitió que el software no operara únicamente como un resolvedor numérico, sino como una plataforma organizada para el análisis técnico de reactores batch ideales homogéneos, manteniendo correspondencia entre datos de entrada, modelo implementado y resultados generados.

La validación del software desarrollado permitió comprobar la consistencia técnica de ReactSim Studio frente a soluciones analíticas, soluciones semianalíticas, problemas de referencia tomados de Fogler y casos equivalentes configurados en Cantera. En los casos isotérmicos, los resultados coincidieron con las soluciones

cerradas esperadas para cinéticas de orden cero, primer orden y segundo orden. En los casos no isotérmicos, el simulador reprodujo adecuadamente el comportamiento térmico asociado al balance de energía y a la dependencia de la constante cinética con la temperatura mediante la ecuación de Arrhenius. Adicionalmente, la contrastación externa con Cantera mostró errores relativos bajos en concentración, conversión y temperatura final, lo que respalda la coherencia numérica del simulador dentro del alcance definido para reactores batch ideales homogéneos.

.

VII. Recomendaciones

A partir de los resultados obtenidos en la presente investigación, se recomienda continuar el desarrollo de ReactSim Studio mediante la ampliación progresiva de su alcance físico-matemático. En particular, una línea natural de trabajo futuro consiste en incorporar reacciones reversibles, múltiples reacciones simultáneas, variación de volumen y modelos de propiedades termofísicas dependientes de la temperatura. Estas extensiones permitirían representar sistemas reaccionantes más complejos y ampliar la aplicabilidad del simulador a problemas de mayor interés técnico y académico.

Asimismo, se recomienda profundizar la validación del software mediante la incorporación de nuevos casos de referencia y, cuando sea posible, mediante la comparación con datos experimentales o con resultados obtenidos en plataformas consolidadas de simulación de procesos. Aunque la validación desarrollada en esta tesis permitió confirmar la coherencia matemática y numérica del simulador dentro de su dominio de aplicación, una validación más amplia fortalecería su confiabilidad y su utilidad como herramienta de análisis.

Desde el punto de vista computacional, se recomienda continuar evaluando el desempeño del motor numérico en escenarios de mayor rigidez, así como ampliar las estrategias automáticas de selección del solver y diagnóstico de convergencia. De igual manera, conviene incorporar herramientas adicionales de análisis, tales como estudios de sensibilidad paramétrica, análisis de escenarios operativos y rutinas de estimación o ajuste de parámetros cinéticos, con el fin de enriquecer la capacidad analítica del programa.

En relación con la interfaz y la experiencia de uso, se recomienda conservar el enfoque de claridad y trazabilidad adoptado en esta versión, pero complementarlo con mejoras orientadas a la documentación interna, la exportación de reportes y la organización de bibliotecas de casos de referencia. También sería conveniente realizar pruebas de uso con estudiantes o usuarios del área de Ingeniería Química, a fin de identificar dificultades de interpretación, mejorar la claridad de los mensajes del sistema y evaluar su utilidad como herramienta de apoyo académico.

Finalmente, se recomienda preservar el carácter abierto y modular de ReactSim Studio como criterio central para su evolución futura. La arquitectura desarrollada en esta tesis constituye una base adecuada para ampliar el simulador sin perder coherencia entre teoría, algoritmo e implementación. En este sentido, el trabajo realizado no debe entenderse como un punto final, sino como una plataforma inicial sólida para desarrollos posteriores en simulación de reactores químicos.

VIII. Bibliografía

Ascher, U. M., & Petzold, L. R. (1998). *Computer methods for ordinary differential equations and differential-algebraic equations*. Society for Industrial and Applied Mathematics. Recuperado de: https://epubs.siam.org/doi/book/10.1137/1.9781611971392

Bird, R. B., Stewart, W. E., & Lightfoot, E. N. (2007). *Transport phenomena* (2.ª ed.). John Wiley & Sons.

Cantera Developers. (2025). *Cantera 3.2.0 documentation*. Recuperado de: https://cantera.org/3.2/

DWSIM. (s. f.). *DWSIM: The open source chemical process simulator*. Recuperado de: https://dwsim.org/

Espinosa Luna, C. A., & Rivas Medal, B. G. (2026). *ReactSim Studio* (versión 1.0) [Software de computadora]. GitHub. Recuperado de: https://github.com/DTCarlox77/ReactSimStudio

Espinosa Luna, C. A., & Rivas Medal, B. G. (2026). *ReactSim Studio* [Página web]. Recuperado de: https://react-sim-landing.vercel.app/

Fogler, H. S. (2016). *Elements of chemical reaction engineering* (5.ª ed.). Pearson Education.

Goodwin, D. G., Moffat, H. K., Schoegl, I., Speth, R. L., & Weber, B. W. (2025). *Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes* (Version 3.2.0) [Software de computadora]. Zenodo. https://doi.org/10.5281/zenodo.17620923

Hairer, E., Nørsett, S. P., & Wanner, G. (1993). *Solving ordinary differential equations I: Nonstiff problems* (2.ª ed. rev.). Springer. Recuperado de: https://link.springer.com/book/10.1007/978-3-540-78862-1

Hairer, E., & Wanner, G. (1996). *Solving ordinary differential equations II: Stiff and differential-algebraic problems* (2.ª ed. rev.). Springer. Recuperado de: https://link.springer.com/book/10.1007/978-3-642-05221-7

Incropera, F. P., DeWitt, D. P., Bergman, T. L., & Lavine, A. S. (2007). *Fundamentals of heat and mass transfer* (6.ª ed.). John Wiley & Sons.

Kumar, R., Medlin, J. W., & Vlachos, D. G. (2021). TAPsolver: A Python package for the simulation and analysis of TAP reactor experiments. *Chemical Engineering Journal, 420*, 129377. Recuperado de: https://doi.org/10.1016/j.cej.2021.129377

Levenspiel, O. (1999). *Chemical reaction engineering* (3.ª ed.). John Wiley & Sons.

National Institute of Standards and Technology. (2008). *Guide for the use of the International System of Units (SI)* (NIST Special Publication 811, 2008 ed.).

U.S. Department of Commerce. Recuperado de: https://physics.nist.gov/cuu/pdf/sp811.pdf

NIST/CODATA Task Group on Fundamental Constants. (2018). CODATA recommended values of the fundamental physical constants: 2018. *Reviews of Modern Physics, 93*(2), 025010. Recuperado de: https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.93.025010

Qt Company. (s. f.). *Qt for Python: PySide6 documentation*. Recuperado de: https://doc.qt.io/qtforpython-6/

SciPy Developers. (2023). *scipy.integrate.solve_ivp — SciPy reference guide*. Recuperado de: https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html

Smith, J. M. (1981). *Chemical engineering kinetics* (3.ª ed.). McGraw-Hill.

University of Colorado Boulder, Department of Chemical and Biological Engineering. (s. f.). *LearnChemE: Isothermal batch reactors – Example problems*. Recuperado de: https://learncheme.com/quiz-yourself/interactive-self-study-modules/isothermal-batch-reactors/isothermal-batch-reactors-example-problems

University of Colorado Boulder, Department of Chemical and Biological Engineering. (s. f.). *LearnChemE: Nonisothermal batch reactor simulations*. Recuperado de: https://learncheme.com/quiz-yourself/interactive-self-study-modules/isothermal-batch-reactors/isothermal-batch-reactors-simulation/

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., & SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods, 17*(3), 261–272. Recuperado de: https://www.nature.com/articles/s41592-019-0686-2

IX. Anexos

Anexo A. Código DOT del Flujograma Interno de ReactSim Studio

El presente anexo contiene el código fuente en lenguaje DOT correspondiente al flujograma interno completo de ReactSim Studio, elaborado a partir de la estructura funcional real del software implementado. Este material complementa los flujogramas resumidos presentados en el Capítulo VII y ofrece una representación más detallada de la lógica interna del simulador.

Listado A.1.Código fuente en lenguaje DOT

```
digraph ReactSimStudioFlow {
    rankdir=TB;
    graph [
        fontname="Segoe UI",
        fontsize=11,
        labelloc="t",
        label="ReactSim Studio - Flujo interno de simulación",
        bgcolor="white",
        pad="0.3",
        nodesep="0.35",
        ranksep="0.55"
    ];
    node [
        shape=box,
        style="rounded,filled",
        fillcolor="#ffffff",
        color="#cfd8e3",
        fontname="Segoe UI",
        fontsize=10,
        margin="0.12,0.08"
    ];
    edge [
        color="#5f6b7a",
        arrowsize=0.8,
        fontname="Segoe UI",
        fontsize=9
    ];

    start [label="Inicio.\nUsuario configura un caso en la GUI.", fillcolor="#eef4ff",
color="#bdd0ff"];
    build_input [label="UI -> _build_simulation_input().\nSe arma un SimulationInput con
especies,\ncinética, térmica y solver."];
    validate_fields [label="Validación sintáctica primaria.\nSe detectan vacíos, números
inválidos\ny tablas incompletas."];
    invalid_input [shape=diamond, label="¿Hay errores de entrada?", fillcolor="#fff7e6",
color="#f0d18a"];
    show_ui_error [label="Se marca la GUI.\nLabels en rojo, consola de validación,\nalerta y enfoque
del primer campo inválido.", fillcolor="#fff5f5", color="#f1b4ad"];

    subgraph cluster_validation {
        label="Normalización y validación del caso.";
        color="#dbe4ee";
        style="rounded";

        normalize_case [label="normalize_case().\nConvierte el SimulationInput en SimulationCase."];
        norm_species [label="_normalize_species().\nValida nombres, unicidad,
signos,\nestequiometría y C0 >= 0."];
```

```
        norm_solver [label="_normalize_solver().\nConvierte tiempo a SI, valida t_end >
t_start,\npuntos de salida y tolerancias."];
        norm_thermal [label="_normalize_thermal().\nConvierte T, ΔHr, ρ, Cp, U, A, Tc y TMP.\nDefine
modo isotérmico, adiabático o con chaqueta."];
        norm_kinetics [label="_normalize_kinetics().\nResuelve ley de potencia o expresión
personalizada.\nConvierte k o k0 según el orden total."];
        custom_expr [label="_validate_custom_expression().\nCompila AST seguro y prueba la
expresión\ncon el estado inicial.", fillcolor="#f7fbff"];
        valid_case [shape=diamond, label="¿El caso es físicamente coherente?", fillcolor="#eefaf2",
color="#b7e3c4"];
    }

    subgraph cluster_rate {
        label="Construcción de la cinética.";
        color="#dbe4ee";
        style="rounded";

        build_rate [label="_build_rate_function().\nDevuelve una función r(t, C, T)."];
        rate_custom [label="Modo custom.\nEvalúa símbolos {especies, t, T, R}\nen entorno
restringido.", fillcolor="#f7fbff"];
        rate_power [label="Modo power_law.\nCalcula r = k * Π Ci^αi\no r = k0 * exp(-Ea / RT) * Π
Ci^αi.", fillcolor="#f7fbff"];
    }

    subgraph cluster_solver {
        label="Integración numérica.";
        color="#dbe4ee";
        style="rounded";

        solve_entry [label="_solve().\nConstruye y0, t_span y t_eval."];
        auto_methods [label="_auto_methods().\nHeurística para priorizar RK45,\nRadau o BDF.",
fillcolor="#f7fbff"];
        dynamics [label="dynamics(t, y).\nRecorta C >= 0 para evaluar la cinética y calcula:\n
dCi/dt = vi * r.\n Si aplica, dT/dt por balance de energía.", fillcolor="#f7fbff"];
        solve_loop [label="Bucle de solve_ivp().\nIntenta métodos según estrategia\nauto o
manual."];
        success_method [shape=diamond, label="¿El método convergió?", fillcolor="#eefaf2",
color="#b7e3c4"];
        stiff_check [shape=diamond, label="¿RK45 parece rígido\npor nfev elevado?",
fillcolor="#fff7e6", color="#f0d18a"];
        retry_implicit [label="Se agrega advertencia y se repite\ncon método implícito.",
fillcolor="#fff9e8", color="#f2d58a"];
        solver_fail [label="Se acumula advertencia.\nSi todos fallan, se reporta error de
simulación.", fillcolor="#fff5f5", color="#f1b4ad"];
    }

    subgraph cluster_post {
        label="Postproceso y salida.";
        color="#dbe4ee";
        style="rounded";

        assemble_result [label="simulate_case().\nExtrae tiempo, concentraciones,\ntemperatura y
velocidad."];
        reference_species [label="resolve_reference_species().\nSelecciona especie de referencia\ny
reactivo limitante.", fillcolor="#f7fbff"];
        conversion [label="_conversion_profile().\nCalcula X a partir de la especie\nde
referencia.", fillcolor="#f7fbff"];
        recommended_stop [label="_recommend_stop_time().\nBusca el primer instante en que el
cambio\nremanente del sistema cae por debajo del umbral.", fillcolor="#f7fbff"];
        heat_terms [label="Si el caso es no isotérmico,\ncalcula calor generado y retirado.",
fillcolor="#f7fbff"];
        tmp_check [label="Si existe TMP,\ncompara Tmax con el límite definido\ny marca
incumplimiento si corresponde.", fillcolor="#fff9e8", color="#f2d58a"];
        result_object [label="Se crea SimulationResult.\nIncluye perfiles, diagnósticos,\nmensajes y
advertencias.", fillcolor="#eefaf2", color="#b7e3c4"];
    }

    subgraph cluster_ui {
        label="Presentación en la interfaz.";
```

```
        color="#dbe4ee";
        style="rounded";

        apply_result [label="_apply_result().\nConvierte salidas a unidades de pantalla,\nactualiza
métricas, gráficas, tabla y alertas."];
        graph_annotations [label="La GUI dibuja:\n- punto de tiempo recomendado,\n- línea vertical
de referencia,\n- línea horizontal de TMP si existe.", fillcolor="#f7fbff"];
        trace_build [label="build_calculation_trace().\nGenera la terminal interna paso a paso.",
fillcolor="#f7fbff"];
        success_alert [label="Se muestra estado final.\nAlerta de éxito, mensajes y sonido.",
fillcolor="#eefaf2", color="#b7e3c4"];
        export_path [label="Opcional.\nGuardar caso .reactsim o exportar PDF.",
fillcolor="#f7fbff"];
        end [label="Fin.\nCaso listo para inspección,\nexportación o reutilización.",
fillcolor="#eef4ff", color="#bdd0ff"];
    }

    start -> build_input -> validate_fields -> invalid_input;
    invalid_input -> show_ui_error [label="Sí"];
    invalid_input -> normalize_case [label="No"];
    show_ui_error -> end;

    normalize_case -> norm_species -> norm_solver -> norm_thermal -> norm_kinetics;
    norm_kinetics -> custom_expr [label="Si la cinética es custom."];
    norm_kinetics -> valid_case [label="Si la cinética es ley de potencia."];
    custom_expr -> valid_case;
    valid_case -> show_ui_error [label="No"];
    valid_case -> build_rate [label="Sí"];

    build_rate -> rate_custom [label="Modo custom"];
    build_rate -> rate_power [label="Modo power_law"];
    rate_custom -> solve_entry;
    rate_power -> solve_entry;

    solve_entry -> auto_methods -> dynamics -> solve_loop -> success_method;
    success_method -> solver_fail [label="No"];
    solver_fail -> end;
    success_method -> stiff_check [label="Sí"];
    stiff_check -> retry_implicit [label="Sí"];
    retry_implicit -> solve_loop;
    stiff_check -> assemble_result [label="No"];

    assemble_result -> reference_species -> conversion -> recommended_stop -> heat_terms ->
tmp_check -> result_object;
    result_object -> apply_result -> graph_annotations -> trace_build -> success_alert ->
export_path -> end;
}
```

Nota. *El código DOT incluido en este anexo debe colocarse en un editor o visor compatible con Graphviz para generar la representación gráfica completa del flujograma. De esta manera, el anexo constituye no solo una evidencia documental del flujo interno del simulador, sino también un recurso reproducible que permite reconstruir visualmente la lógica de operación de ReactSim Studio.*

Anexo B. Soluciones Analíticas y Semianalíticas de Referencia para la Validación de ReactSim Studio

El presente anexo reúne los desarrollos analíticos y semianalíticos utilizados como base de comparación para la validación de ReactSim Studio. En todos los casos se parte de la ecuación de diseño del reactor batch ideal homogéneo a volumen constante, en concordancia con la formulación clásica expuesta por Fogler (2016). Los resultados obtenidos en este anexo sustentan las comparaciones reportadas en la Sección 5.6 del capítulo de resultados.

Para mantener coherencia con el alcance del simulador, los casos aquí presentados se restringen a una sola reacción irreversible, fase líquida y volumen constante. En los ejemplos isotérmicos y en los ejemplos no isotérmicos adaptados, se emplean parámetros consistentes con los benchmarks implementados en ReactSim Studio. En el último caso, correspondiente al Ejemplo 9-1 de Fogler, se sigue el procedimiento semianalítico del propio texto.

B.1. Reactor Batch Isotérmico de Orden Cero

Considérese una reacción irreversible de orden cero:

$$A \rightarrow P$$

Bajo condiciones isotérmicas y volumen constante. Para un reactor batch ideal, el balance de materia se expresa como:

$$\frac{dC_A}{dt} = r_A$$

Y, para una reacción de orden cero la ley de velocidad es:

$$-r_A = k$$

Por ende:

$$\frac{dC_A}{dt} = -k$$

Integrando entre $t = 0$, cuando $C_A = C_{A0}$, y un tiempo cualquiera t, cuando $C_A = C_A(t)$, se obtiene:

$$\int_{C_{A0}}^{C_A} dC_A = -\int_0^t k\, dt$$

$$C_A - C_{A0} = -kt$$

De donde resulta la solución analítica de concentración, presentada en la Ecuación (B.1):

$$C_A(t) = C_{A0} - kt \qquad (\mathrm{B.1})$$

La conversión de la especie A se define como:

$$X_A = \frac{C_{A0} - C_A}{C_{A0}}$$

Sustituyendo la expresión anterior, la conversión correspondiente se obtiene mediante la Ecuación (B.2):

$$X_A(t) = \frac{kt}{C_{A0}} \tag{B.2}$$

Sustitución para el caso benchmark

Para el caso implementado en ReactSim Studio:

- $C_{A0} = 2.0 \frac{mol}{L}$
- $k = 0.05 \frac{mol}{Lmin}$
- $t_f = 20\ min$

Entonces,

$$C_A(20) = 2.0 - (0.05)(20) = 1 \frac{mol}{L}$$

Y

$$X_A(20) = \frac{(0.05)(20)}{2.0} = 0.50$$

Por tanto, la solución analítica de referencia es:

$$C_A(20\ min) = 1.0 \frac{mol}{L}$$

$$X_A(20\ min) = 0.50 \rightarrow 50\%$$

B.2. Reactor Batch Isotérmico de Primer Orden

Para una reacción irreversible de primer orden:

$$A \rightarrow P$$

El balance de materia para un reactor batch ideal es:

$$\frac{dC_A}{dt} = -kC_A$$

Separando variables:

$$\frac{dC_A}{C_A} = -k\ dt$$

Integrando entre C_{A0} y $C_A(t)$, entre 0 y t:

$$\int_{C_{A0}}^{C_A} \frac{dC_A}{C_A} = -\int_0^t k\, dt$$

$$\ln\left(\frac{C_A}{C_{A0}}\right) = -kt$$

La solución analítica de concentración para el caso de primer orden se expresa mediante la Ecuación (B.3):

$$C_A(t) = C_{A0}e^{-kt} \tag{B.3}$$

La conversión correspondiente se presenta en la Ecuación (B.4):

$$X_A(t) = \frac{C_{A0} - C_A}{C_{A0}} = 1 - e^{-kt} \tag{B.4}$$

Sustitución para el caso benchmark

Para el caso implementado en ReactSim Studio:

- $C_{A0} = 2.0\,\frac{mol}{L}$
- $k = 0.15\,min^{-1}$
- $t_f = 25\,min$

Entonces,

$$C_A(25\,min) = 2.0e^{-(0.15)(25)} = 0.0470\,\frac{mol}{L}$$

$$X_A(25\,min) = 1 - e^{-3.75} = 0.97648$$

Por tanto, la solución analítica de referencia es:

$$C_A(25\,min) = 0.0470\,\frac{mol}{L}$$

$$X_A(25\,min) = 0.97648 \rightarrow 97.64\%$$

B.3. Reactor Batch Isotérmico de Segundo Orden

Para una reacción irreversible de segundo orden:

$$A \rightarrow P$$

El balance de materia queda:

$$\frac{dC_A}{dt} = -kC_A^2$$

Separando variables:

$$\frac{dC_A}{dt} = -kC_A^2$$

Integrando:

$$\int_{C_{A0}}^{C_A} C_A^{-2}\, dC_A = -\int_0^t k\, dt$$

$$\left[-\frac{1}{C_A}\right]_{C_{A0}}^{C_A} = -kt$$

$$-\frac{1}{C_A} + \frac{1}{C_{A0}} = -kt$$

Reordenando:

$$\frac{1}{C_A} - \frac{1}{C_{A0}} = kt$$

La solución analítica de concentración para el caso de segundo orden se presenta en la Ecuación (B.5):

$$C_A(t) = \frac{C_{A0}}{1 + kC_{A0}t} \tag{B.5}$$

La conversión se expresa como:

$$X_A(t) = 1 - \frac{C_A}{C_{A0}}$$

Sustituyendo:

$$X_A(t) = 1 - \frac{1}{1 + kC_{A0}t}$$

La conversión correspondiente se expresa mediante la Ecuación (B.6):

$$X_A(t) = \frac{kC_{A0}t}{1 + kC_{A0}t} \tag{B.6}$$

Sustitución para el caso de benchmark

Para el caso implementado en ReactSim Studio:

- $C_{A0} = 1.0\,\frac{mol}{L}$
- $k = 0.20\,\frac{L}{mol\ min}$
- $t_f = 10\ min$

Entonces,

$$C_A(10) = \frac{1.0}{1 + (0.20)(1.0)(10)} = \frac{1}{3} = 0.3333 \frac{mol}{L}$$

Y

$$X_A(10) = \frac{(0.20)(1.0)(10)}{1 + (0.20)(1.0)(10)} = \frac{2}{3} = 0.6667$$

Por tanto, la solución analítica de referencia es:

$$C_A(10\ min) = \ 0.3333 \frac{mol}{L}$$

$$X_A(10\ min) = 0.6667 \rightarrow 66.67\%$$

B.4. Reactor Batch No Isotérmico Adiabático con Cinética de Primer Orden y k Constante

Siguiendo la formulación general de Fogler para reactores batch no isotérmicos, considérese una reacción irreversible de primer orden:

$$A \rightarrow P$$

Con constante cinética independiente de la temperatura, es decir:

$$-r_A = kC_A$$

Balance de materia

Para el reactor batch ideal:

$$\frac{dC_A}{dt} = -kC_A$$

Cuya solución es:

$$C_A(t) = C_{A0}e^{-kt}$$

Y la conversión:

$$X_A(t) = 1 - e^{-kt}$$

Balance de energía adiabático

En condiciones adiabáticas, el balance de energía se escribe como:

$$\rho C_p V \frac{dT}{dt} = (-\Delta H_r)Vr$$

Donde $r = kC_A$. Como el volumen es constante, se simplifica a:

$$\rho C_p \frac{dT}{dt} = (-\Delta H_r) k C_A$$

Sustituyendo $C_A = C_{A0}(1 - X)$ o, de manera equivalente, dividiendo el balance de energía entre el balance de conversión, se obtiene:

$$\frac{dT}{dX} = \frac{(-\Delta H_r) C_{A0}}{\rho C_p}$$

Integrando desde $X = 0$, $T = T_0$, hasta $X = X(t)$, $T = T(t)$:

$$\int_{T_0}^{T} dT = \frac{(-\Delta H_r) C_{A0}}{\rho C_p} \int_{0}^{X} dX$$

$$T - T_0 = \frac{(-\Delta H_r) C_{A0}}{\rho C_p} X$$

La relación temperatura-conversión para el caso adiabático se expresa mediante la Ecuación (B.7):

$$T(t) = T_0 + \frac{(-\Delta H_r) C_{A0}}{\rho C_p} X_A(t) \tag{B.7}$$

Al sustituir la conversión de primer orden de la Ecuación (B.4), se obtiene la evolución térmica mostrada en la Ecuación (B.8):

$$T(t) = T_0 + \frac{(-\Delta H_r) C_{A0}}{\rho C_p} (1 - e^{-kt}) \tag{B.8}$$

Esta es la solución analítica cerrada del caso no isotérmico adiabático bajo cinética de primer orden con k constante.

Sustitución para el caso de benchmark

Para el caso implementado en ReactSim Studio:

- $C_{A0} = 2.0 \frac{mol}{L} \rightarrow 2000 \frac{mol}{m^3}$
- $k = 0.15\, min^{-1}$
- $T_0 = 298.15\, K$
- $\Delta H_r = -65.0 \frac{kJ}{mol} \rightarrow -65000 \frac{J}{mol}$
- $\rho = 1000 \frac{kg}{m^3}$
- $C_p = 4180 \frac{J}{kgK}$

Primero se calcula el factor térmico:

$$\beta = \frac{(-\Delta H_r)C_{A0}}{\rho C_p}$$

$$\beta = \frac{(65000)(2000)}{(1000)(4180)} = 31.1005\,K$$

A $t = 20\ min$:

$$X_A(20) = 1 - e^{-(0.15)(20)} = 1 - e^{-3} = 0.95021$$

Y

$$C_A(20) = 2.0e^{-3} = 0.09957\,\frac{mol}{L}$$

La temperatura se calcula como:

$$T(20) = 298.15 + 31.1005(0.95021) = 327.70\ \ K$$

Por tanto, la solución analítica de referencia es:

$$C_A(20\ min) = 0.09957\,\frac{mol}{L}$$

$$X_A(20\,min) = 0.95021 \rightarrow 95.02\%$$

$$T(20\ min) = 327.70\,K$$

B.5. Reactor Batch No Isotérmico con Chaqueta a Temperatura Constante, Cinética de Primer Orden y k Constante

Para el caso con intercambio térmico, se mantiene el mismo balance de materia:

$$\frac{dC_A}{dt} = -kC_A$$

Cuya solución es:

$$C_A(t) = C_{A0}e^{-kt}$$

Y la conversión:

$$X_A(t) = 1 - e^{-kt}$$

Balance de energía con chaqueta

Cuando el reactor intercambia calor con una chaqueta a temperatura constante T_c, el balance de energía es:

$$\rho C_p V\frac{dT}{dt} = (-\Delta H_r)VkC_A - UA(T - T_c)$$

Sustituyendo $C_A = C_{A0}e^{-kt}$

$$\rho C_p V \frac{dT}{dt} = (-\Delta H_r)VkC_{A0}e^{-kt} - UA(T - T_c)$$

Dividiendo entre $\rho C_p V$:

$$\frac{dT}{dt} + \frac{UA}{\rho C_p V}T = \frac{(-\Delta H_r)kC_{A0}}{\rho C_p}e^{-kt} + \frac{UA}{\rho C_p V}T_c$$

Los parámetros auxiliares del modelo térmico con chaqueta se definen mediante la Ecuación (B.9):

$$a = \frac{(-\Delta H_r)kC_{A0}}{\rho C_p} \quad ; \quad b = \frac{UA}{\rho C_p V} \tag{B.9}$$

La ecuación se escribe como:

$$\frac{dT}{dt} + bT = ae^{-kt} + bT_c$$

Resolución de la ecuación diferencial

Se trata de una ecuación diferencial lineal de primer orden. El factor integrante es:

$$\mu(t) = e^{bt}$$

Multiplicando toda la ecuación por e^{bt}:

$$e^{bt}\frac{dT}{dt} + be^{bt}T = ae^{(b-k)t} + bT_c e^{bt}$$

Lo que es equivalente a:

$$\frac{d}{dt}(Te^{bt}) = ae^{(b-k)t} + bT_c e^{bt}$$

Integrando entre 0 y t:

$$T(t)e^{bt} - T_0 = \frac{a}{b-k}\left(e^{(b-k)t} - 1\right) + T_c(e^{bt} - 1)$$

Con estas definiciones, la solución analítica cerrada del caso batch no isotérmico con chaqueta a temperatura constante se presenta en la Ecuación (B.10):

$$T(t) = T_c + (T_0 - T_c)e^{-bt} + \frac{a}{b-k}(e^{-kt} - e^{-bt}) \tag{B.10}$$

Esta expresión constituye la solución analítica cerrada del caso batch no isotérmico con chaqueta a temperatura constante, bajo cinética de primer orden con k constante.

Sustitución para el caso benchmark

Para el caso implementado en ReactSim Studio:

- $C_{A0} = 2.0\,\frac{mol}{L} \rightarrow 2000\,\frac{mol}{m^3}$
- $k = 0.15\,min^{-1}$
- $T_0 = 298.15\,K$
- $\Delta H_r = -65000\,\frac{J}{mol}$
- $\rho = 1000\,\frac{kg}{m^3}$
- $C_p = 4180\,\frac{J}{kgK}$
- $V = 1.0L \rightarrow 0.001\,m^3$
- $U = 250\,\frac{W}{m^2K}$
- $A = 1.80\,m^2$
- $T_c = 290.0\,K$

Como en esta derivación t se expresa en minutos, el término UA debe convertirse a J/min/K:

$$UA = (250)(1.80)(60) = 27000\,\frac{J}{minK}$$

Luego,

$$a = \frac{(65000)(0.15)(2000)}{(1000)(4180)} = 4.66507\,\frac{K}{min}$$

$$b = \frac{27000}{(1000)(4180)(0.001)} = 6.45933\,min^{-1}$$

A $t = 20\,min$:

$$C_A(20) = 2.0\,e^{-3} = 0.09957\,\frac{mol}{L}$$

$$X_A(20) = 1 - e^{-3} = 0.95021$$

Y

$$T(t) = 290 + (298.15 - 290)e^{-6.45933(20)} + \frac{4.66507}{6.45933 - 0.15}\left(e^{-0.15(20)} - e^{-6.45933(20)}\right)$$

$$T(20) = 290.04\,K$$

Por tanto, la solución analítica de referencia es:

$$C_A(20\,min) = 0.09957\,\frac{mol}{L}$$

$$X_A(20\,min) = 0.95021 \rightarrow 95.02\%$$

$$T(20\,min) = 290.04\,K$$

B.6. Resolución Semianalítica del Ejemplo 9-1 de Fogler: Reactor por Lotes Adiabático

El **Ejemplo 9-1** de Fogler corresponde a un reactor por lotes adiabático y constituye una referencia importante para la validación de ReactSim Studio en modo no isotérmico. A diferencia de los casos anteriores, este problema no conduce a una solución cerrada simple, ya que la constante cinética depende de la temperatura y la temperatura, a su vez, depende de la conversión. Por ello, su tratamiento es semianalítico: parte de relaciones analíticas del modelo, pero la integral de tiempo se evalúa numéricamente.

1. Ecuación de diseño

Para el reactor batch ideal:

$$N_{A0}\frac{dX}{dt} = -r_A V$$

Como el volumen se considera constante:

$$\frac{dX}{dt} = k(1 - X)$$

2. Ley de velocidad

De acuerdo con el ejemplo:

$$-r_A = kC_A$$

Y, por estequiometría en volumen constante,

$$C_A = C_{A0}(1 - X)$$

3. Relación adiabática entre temperatura y conversión

Fogler utiliza la relación lineal entre temperatura y conversión presentada en la Ecuación (B.11):

$$T = 535 + 90.45X \ (T\ en\ °R) \qquad (\mathrm{B.11})$$

Esta expresión proviene del balance adiabático aplicado a los datos del ejemplo 8-4. Es importante notar que, aunque los materiales de alimentación se reportan inicialmente a 58°F, el problema se resuelve a partir de una temperatura efectiva de mezcla de 75°F, equivalente a 535°R.

4. Dependencia de la constante cinética con la temperatura

La constante cinética dependiente de la temperatura se calcula mediante la Ecuación (B.12):

$$k = 0.000273 \exp\left[16306\left(\frac{1}{535} - \frac{1}{T}\right)\right] \quad (s^{-1}) \qquad (\mathrm{B.12})$$

Al sustituir la Ecuación (B.11) en la Ecuación (B.12), la ecuación de diseño se transforma en la integral de tiempo indicada en la Ecuación (B.13):

$$t = \int_0^{0.515} \frac{dX}{0.000273 \exp\left[16306\left(\frac{1}{535} - \frac{1}{535 + 90.45X}\right)\right](1 - X)} \qquad (B.13)$$

5. Evaluación por la regla de Simpson

Fogler resuelve esta integral mediante tabulación y regla de Simpson con cuatro subintervalos. Para ello se toma:

$$h = \frac{0.515}{4} = 0.12875$$

Los valores tabulados del problema son los siguientes:

Tabla B.1.

Valores intermedios para la evaluación del Ejemplo 9-1 de Fogler

$\boldsymbol{X}$	$\boldsymbol{T}(°\boldsymbol{R})$	$\boldsymbol{k}(\boldsymbol{s^{-1}})$	$\frac{\mathbf{1}}{[\boldsymbol{K}(\mathbf{1}-\boldsymbol{X})]}(\boldsymbol{s})$
0.0000	535	2.73×10^{-4}	3663
0.1288	547	5.33×10^{-4}	2154
0.2575	558	9.59×10^{-4}	1404
0.3863	570	17.7×10^{-4}	921
0.5150	582	32.0×10^{-4}	644

Nota. Adaptado de Fogler.

Aplicando Simpson:

$$t = \frac{0.12875}{3}[f_0 + 4f_1 + 2f_2 + 4f_3 + f_4]$$

$$t = \frac{0.12875}{3}[3663 + 4(2154) + 2(1404) + 4(921) + 644]$$

$$t = 833s \rightarrow 13.9\,min$$

La temperatura correspondiente a $X = 0.515$ es:

$$T = 535 + 90.45(0.515) = 581.58°R \rightarrow 582°R$$

Lo que equivale a:

$$T \approx 122°F$$

6. Resultado de referencia

Por lo tanto, la solución de referencia de Fogler para el Ejemplo 9-1 es:

$$X = 0.515 \rightarrow 51.5\%$$

$$t = 13.9\ min$$

$$T \approx 122°F$$

Estos valores se utilizaron como benchmark de comparación frente al caso equivalente implementado en ReactSim Studio.

B.7. Observación Final del Anexo

Las soluciones analíticas y semianalíticas presentadas en este anexo constituyen la base formal de comparación utilizada en el capítulo de resultados. En los casos isotérmicos y en los casos no isotérmicos adaptados, las expresiones cerradas permitieron contrastar directamente la salida de ReactSim Studio con resultados exactos. En el caso del Ejemplo 9-1 de Fogler, el procedimiento semianalítico permitió disponer de un benchmark confiable para evaluar la respuesta del simulador en un escenario no isotérmico con dependencia térmica de la cinética.

Anexo C. Evidencia del Repositorio Público y Licencia de ReactSim Studio

El presente anexo reúne la evidencia documental asociada al repositorio público utilizado para respaldar el desarrollo de ReactSim Studio. Su propósito es dejar constancia de la disponibilidad del código fuente, la licencia de distribución, la estructura principal del proyecto, la documentación técnica y el punto de referencia del control de versiones empleado en la presente investigación.

La incorporación de este anexo responde al carácter abierto del software desarrollado y permite verificar que el simulador no solo fue descrito metodológicamente en el cuerpo del documento, sino que también cuenta con un soporte técnico accesible para su revisión, ejecución, reproducción y posible ampliación futura.

C.1. Dirección del Repositorio y Fecha de Consulta

El código fuente de ReactSim Studio se encuentra disponible en un repositorio público alojado en GitHub. Para efectos de trazabilidad académica, se registró la dirección del repositorio, el usuario propietario y la fecha de consulta utilizada como referencia documental.

Dirección del repositorio:
https://github.com/DTCarlox77/ReactSimStudio

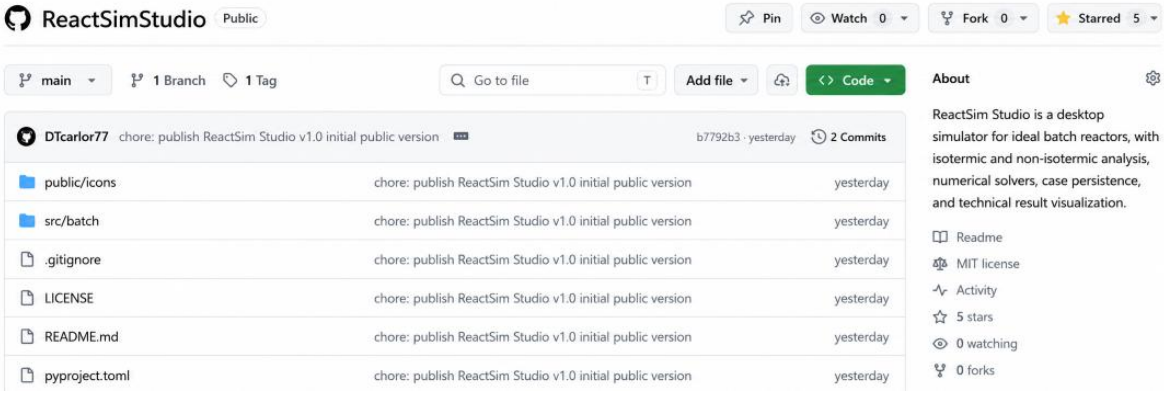
Usuario propietario:
DTCarlox77

Fecha de consulta:
28 de abril de 2026

El repositorio consultado contiene los archivos principales del proyecto, incluyendo el código fuente del simulador, la documentación de uso, los archivos de configuración del entorno Python, los recursos auxiliares y el archivo de licencia.

Figura C.1.

Página principal del repositorio público de ReactSim Studio en GitHub



Nota. Elaboración propia.

C.2. Archivo de Licencia MIT

El repositorio incluye un archivo denominado LICENSE, en el cual se establece que el software se distribuye bajo la **Licencia MIT**. Esta licencia permite el uso, copia, modificación, fusión, publicación, distribución, sublicenciamiento y venta de copias del software, siempre que se mantenga el aviso de copyright y el texto de la licencia en las copias o porciones sustanciales del programa.

La inclusión de esta licencia formaliza el carácter abierto de ReactSim Studio y respalda uno de los elementos centrales del objetivo general de la investigación: desarrollar un software de código abierto para la simulación de reactores batch ideales homogéneos.

- LICENSE

El encabezado del archivo contiene la denominación **MIT License**, seguida del aviso de copyright correspondiente y de las cláusulas de permiso y exención de responsabilidad propias de esta licencia.

Figura C.2. Archivo LICENSE del repositorio de ReactSim Studio

```
MIT License

Copyright (c) 2026 Carlos Adrián Espinosa Luna

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

Nota. Elaboración propia con base en el archivo README.md del repositorio público consultado.

C.3. Estructura Principal de Carpetas y Archivos

La organización del repositorio permite identificar una separación clara entre el código fuente, los recursos auxiliares, los archivos de configuración y la documentación base del proyecto. Esta estructura favorece la mantenibilidad del software y permite relacionar la implementación real con la arquitectura modular descrita en el cuerpo principal de la tesis.

Tabla C.1.

Estructura principal del repositorio ReactSim Studio

Directorio o archivo	***Descripción general***
public/	Agrupa recursos públicos complementarios.
src/	Contiene el código fuente principal del simulador.
LICENSE	Archivo de licencia MIT del proyecto.
README.md	Documento principal de presentación, instalación y uso general.
pyproject.toml	Archivo de configuración base del proyecto Python.
requirements.txt	Lista de dependencias necesarias para instalación y ejecución.
.gitignore	Archivo de configuración para excluir elementos locales del control de versiones.

Nota. Elaboración propia con base en el repositorio público consultado el 28 de abril de 2026.

A nivel de código fuente, el directorio src/ contiene el paquete principal del programa, dentro del cual se ubican el núcleo de simulación, la interfaz gráfica y los recursos necesarios para la ejecución de la aplicación. Esta organización permite separar la lógica físico-numérica del simulador, la interfaz de usuario y los archivos auxiliares utilizados por el sistema.

C.4. Documentación de Instalación y Ejecución

El repositorio incluye documentación de instalación y ejecución mediante el archivo README.md. Este documento cumple la función de guía inicial para comprender el propósito del proyecto, conocer sus capacidades actuales, revisar el alcance del modelo implementado y reproducir el entorno de ejecución del simulador.

Entre los elementos documentados se encuentran:

- Propósito general del programa.
- Capacidades actuales del software.
- Alcance físico-matemático del modelo.
- Supuestos no incluidos en la versión actual.
- Estructura general del repositorio.
- Requisitos mínimos del entorno.
- Pasos de instalación para desarrollo.
- Comandos de ejecución.
- Uso general de la aplicación.
- Gestión de archivos .reactsim.
- Benchmarks.

La secuencia de instalación descrita en la documentación se basa en el clonado del repositorio, la creación de un entorno virtual, la instalación de dependencias mediante pip y la ejecución del programa desde los comandos definidos o desde el módulo principal de la aplicación.

Tabla C.2.

Contenidos principales de la documentación de instalación y ejecución

Elemento documentado	***Contenido principal***
Descripción del proyecto	Presenta ReactSim Studio como simulador técnico de reactores batch ideales.
Capacidades actuales	Resume modos de simulación, métodos numéricos y funcionalidades principales.
Alcance del modelo	Define las hipótesis bajo las cuales opera la versión actual.
Requisitos	Indica versión de Python, uso de pip y entorno compatible.
Instalación	Describe clonado, entorno virtual e instalación de dependencias.
Ejecución	Incluye comandos para iniciar la aplicación.
Casos .reactsim	Explica el guardado y recuperación de casos de simulación.

Nota. Elaboración propia con base en el archivo README.md del repositorio público consultado.

C.5. Commit de Referencia Utilizado en la Tesis

Con el fin de asegurar la trazabilidad entre el software analizado y los resultados presentados en la tesis, se registró un commit de referencia del repositorio. Este punto de control permite identificar la versión específica del código fuente utilizada como base para la documentación, validación y análisis del simulador.

Commit de referencia:

b7792b35d7517dfb973c98f69bfc88d754cb4810

Fecha del commit:

27 de abril de 2026

Mensaje asociado:

chore: publish ReactSim Studio v1.0 initial public version.

Versión asociada:

ReactSim Studio v1.0

Tabla C.3.

Referencia de control de versiones utilizada en la tesis

Elemento	*Información registrada*
Repositorio	DTCarlox77/ReactSimStudio
Tipo de referencia	Commit
Hash del commit	b7792b35d7517dfb973c98f69bfc88d754cb4810
Fecha	27 de abril de 2026
Mensaje	chore: publish ReactSim Studio v1.0 initial public version.
Versión asociada	ReactSim Studio v1.0

Nota. Elaboración propia con base en el historial del repositorio utilizado para esta investigación.

C.6. Relación entre el Repositorio y los Componentes Descritos en la Tesis

La organización interna del repositorio mantiene correspondencia con la arquitectura modular descrita en el cuerpo principal del documento. En particular, se observa una separación entre el módulo de simulación, la interfaz gráfica, los recursos auxiliares y los archivos de configuración del proyecto.

Tabla C.4.

Correspondencia entre el repositorio y los componentes funcionales descritos en la tesis

Componente descrito en la tesis	*Ubicación principal*	*Contenido asociado*
Módulo de simulación	src/batch/simulation/	Modelos, validación, unidades, cinética e integración numérica.
Módulo de interfaz gráfica	src/batch/ui/	Arranque, ventana principal, tema visual e interacción con el usuario.
Lógica de ventana y resultados	src/batch/ui/window/	Formularios, estado, simulación, trazas, reportes y vistas.
Recursos y casos de referencia	src/batch/resources/	Íconos, recursos auxiliares y casos .reactsim.

Nota. Elaboración propia con base en la estructura del repositorio público consultado.

C.7. Evidencia del Carácter Abierto y Reproducible del Proyecto

La evidencia presentada en este anexo permite comprobar que ReactSim Studio fue desarrollado como un proyecto abierto, documentado y técnicamente reproducible. La existencia de un repositorio público, un archivo de licencia, documentación de instalación, una estructura modular de código y un commit de referencia proporciona elementos suficientes para relacionar el producto computacional con los objetivos planteados en la investigación.

En particular, el repositorio permite:

- Revisar el código fuente del simulador.
- Identificar la licencia de distribución.
- Reproducir el entorno de desarrollo.
- Ejecutar el programa a partir de sus instrucciones de instalación.
- Examinar la estructura modular del sistema.
- Verificar la existencia de casos de referencia y documentación base del proyecto.
- Asociar los resultados de la tesis con una versión específica del software.

De esta manera, el repositorio público no solo funciona como un medio de distribución del software, sino también como una evidencia técnica de trazabilidad, reproducibilidad y transparencia del desarrollo realizado.

C.8. Observación Final del Anexo

El repositorio público de ReactSim Studio constituye el respaldo documental del software desarrollado en la presente investigación. Su estructura, licencia, documentación y control de versiones permiten verificar que el simulador fue implementado como una herramienta abierta, modular y extensible, en correspondencia con el objetivo general de la tesis y con el enfoque metodológico adoptado para su desarrollo.

Por tanto, este anexo complementa la descripción técnica presentada en los capítulos metodológico y de resultados, al ofrecer evidencia directa del soporte computacional utilizado para materializar el modelo de simulación propuesto.

Anexo D. Recursos de Acceso, Instalación y Ejecución de ReactSim Studio

El presente anexo reúne la información técnica básica para acceder, instalar y ejecutar ReactSim Studio desde su repositorio público. Su propósito es proporcionar una referencia práctica para revisar el código fuente, reproducir el entorno de trabajo y ejecutar el simulador en modo de desarrollo.

D.1. Acceso al Repositorio

ReactSim Studio se encuentra disponible en un repositorio público alojado en GitHub. En este repositorio se conserva el código fuente principal del simulador, los archivos de configuración del proyecto, los recursos gráficos básicos y la licencia de distribución.

Repositorio público:

https://github.com/DTCarlox77/ReactSimStudio

El acceso al repositorio permite revisar la estructura interna del programa, instalar sus dependencias, ejecutar el simulador desde el código fuente y realizar modificaciones o ampliaciones futuras conforme al carácter abierto del proyecto.

D.2. Requisitos Básicos

Para ejecutar ReactSim Studio desde el código fuente se requiere un entorno compatible con Python y las bibliotecas utilizadas por el simulador. Los requisitos básicos son:

- Python 3.10 o superior.
- pip actualizado.
- Sistema operativo compatible con Python.
- Windows 10 u 11 para la versión distribuible actual.
- Bibliotecas principales: PySide6, numpy, scipy y matplotlib.

Estas dependencias permiten ejecutar la interfaz gráfica, resolver numéricamente el sistema de ecuaciones diferenciales, procesar los datos de entrada y visualizar los resultados generados por el simulador.

D.3. Archivos Principales del Proyecto

La estructura visible del repositorio contiene los archivos y directorios necesarios para comprender, instalar y ejecutar el simulador. Entre los elementos principales se encuentran:

- README.md: documento de presentación general del proyecto, con información básica de instalación, ejecución y uso.
- requirements.txt: archivo que contiene las dependencias necesarias para ejecutar el simulador.
- pyproject.toml: archivo de configuración base del proyecto Python.
- src/batch: directorio que contiene el código fuente principal del simulador.
- public/icons: directorio destinado a recursos gráficos e íconos del proyecto.

- LICENSE: archivo de licencia MIT del software.
- .gitignore: archivo de configuración para excluir del control de versiones elementos generados localmente.

Dentro de src/batch se concentra la implementación principal de ReactSim Studio, incluyendo el motor de simulación, la validación de entradas, la gestión de unidades, la interfaz gráfica y los recursos internos necesarios para el funcionamiento de la aplicación.

D.4. Instalación desde el Código Fuente

Para instalar ReactSim Studio desde el repositorio, se recomienda clonar el proyecto, crear un entorno virtual e instalar las dependencias declaradas.

```Bash Console
git clone https://github.com/DTCarlox77/ReactSimStudio
cd ReactSimStudio
```

En Windows, el entorno virtual puede crearse y activarse mediante:

```Bash Console
python -m venv venv
venv\Scripts\activate
```

En macOS o Linux, puede utilizarse:

```Bash Console
python -m venv venv
source venv/bin/activate
```

Una vez activo el entorno, las dependencias se instalan con:

```Bash Console
pip install -r requirements.txt
```

Este procedimiento prepara el entorno necesario para ejecutar el simulador desde el código fuente y permite trabajar sobre la versión de desarrollo del proyecto.

D.5. Ejecución del Programa

Con el entorno virtual activo y las dependencias instaladas, ReactSim Studio puede ejecutarse desde el módulo principal de la interfaz gráfica:

```bash
python src/batch/ui/main.py
```

Dependiendo de la configuración definida en el proyecto, también puede estar disponible la ejecución mediante comandos registrados en el entorno, tales como:

```bash
ReactSim
```

O bien:

```bash
reactsim
```

La ejecución directa desde main.py resulta especialmente útil para revisión técnica, pruebas locales o depuración del comportamiento del simulador.

D.6. Recursos Integrados

ReactSim Studio incorpora recursos internos que apoyan la operación del programa y la reproducción de casos de simulación. Entre estos recursos se incluyen archivos de configuración, recursos gráficos e íconos, además de los archivos propios de caso en formato .reactsim cuando son generados o utilizados por el simulador.

Estos recursos permiten conservar configuraciones de simulación, cargar casos previamente definidos, revisar el comportamiento del software y facilitar su uso en contextos técnicos o académicos.

D.7. Uso Inicial del Simulador

Una vez ejecutado el programa, el flujo básico de uso consiste en definir el caso de simulación, ingresar las especies químicas, establecer la estequiometría, configurar la cinética, seleccionar el modo térmico y ejecutar la simulación.

Posteriormente, el usuario puede revisar los resultados mediante resúmenes, gráficas, tablas, mensajes del sistema y consola de cálculo. El simulador también

permite guardar casos en formato .reactsim y exportar resultados, lo que favorece la trazabilidad del análisis y la reproducción posterior de simulaciones.

D.8. Cierre del Anexo

La información presentada en este anexo permite identificar los recursos mínimos necesarios para acceder, instalar, ejecutar y revisar ReactSim Studio desde su código fuente. De esta manera, se complementa la descripción metodológica y técnica del software, proporcionando una referencia práctica para reproducir el entorno de trabajo y utilizar el simulador en condiciones controladas.

Anexo E. Identidad Visual Básica de ReactSim Studio

El presente anexo muestra de forma breve los elementos visuales básicos de ReactSim Studio, utilizados para identificar el programa y mantener una interfaz clara, sobria y coherente con su finalidad técnica.

E.1. Logotipo

ReactSim Studio cuenta con un logotipo propio compuesto por el nombre del programa y un símbolo gráfico de apariencia molecular. El diseño utiliza una combinación de texto negro y azul, asociada a una imagen moderna y vinculada con el área de la Ingeniería Química.

Figura E.1.

Logotipo principal de ReactSim Studio



Nota. Elaboración propia.

E.2. Paleta Cromática Básica

La interfaz de ReactSim Studio emplea una paleta centrada en tonos azules, blanco y grises claros. El azul se utiliza como color principal para botones, pestañas activas, elementos destacados y componentes de identidad visual. Los tonos claros se emplean como fondo y apoyo visual, favoreciendo la legibilidad y la organización del entorno de trabajo.

Tabla E.1.

Paleta cromática básica empleada en ReactSim Studio

Uso principal	*Color*	*Código hexadecimal*	
Color primario	Azul principal	#5170FF	
Fondo suave primario	Azul claro	#EEF2FF	
Borde primario	Azul pálido	#CDD6FF	

Texto primario acentuado	Azul oscuro	#2F4DDB	
Selección de tablas	Azul suave de selección	#D9E4FF	
Texto general	Gris azulado oscuro	#1F2933	

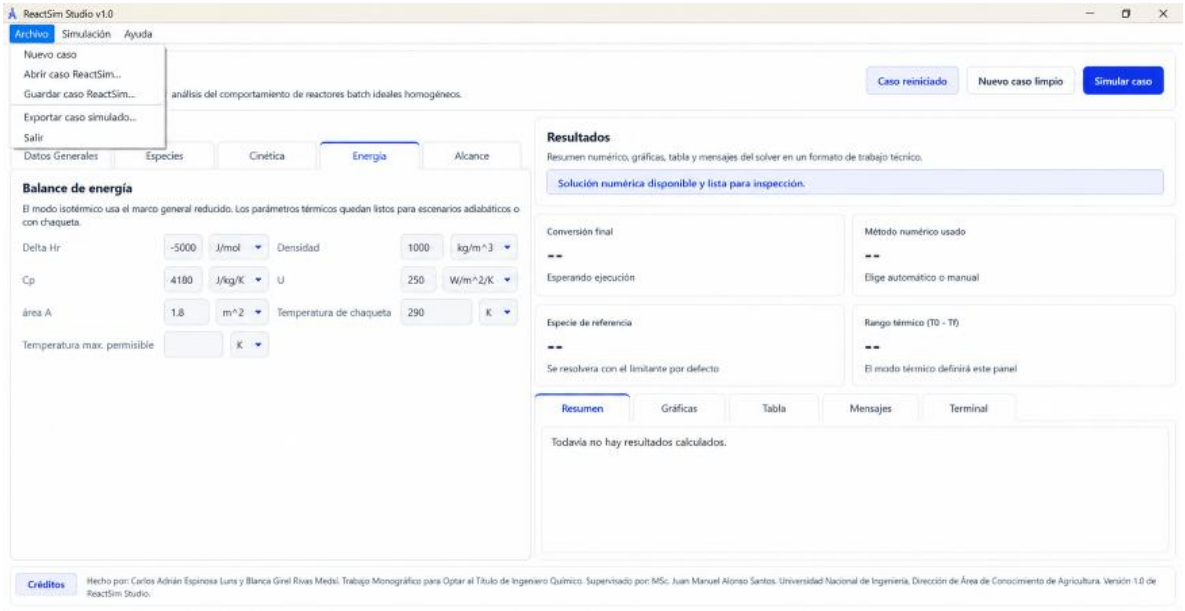
Nota. Elaboración propia con base en el tema visual aplicado en ReactSim Studio.

E.3. Diseño General de la Interfaz

El diseño de la interfaz separa el área de parámetros y el área de resultados, permitiendo que el usuario configure el caso, ejecute la simulación y revise la información generada de forma ordenada. Además, el uso de pestañas, tarjetas de resumen, tablas, gráficas y mensajes de estado contribuye a mantener una experiencia de trabajo clara y funcional.

Figura E.2.

Vista general de la interfaz de ReactSim Studio



Nota. Elaboración propia.

Anexo F. Guía de Usuario de ReactSim Studio

El presente anexo describe el uso básico de ReactSim Studio v1.0, con el propósito de orientar al usuario en la configuración de casos, ejecución de simulaciones, lectura de resultados y uso de funciones complementarias del programa.

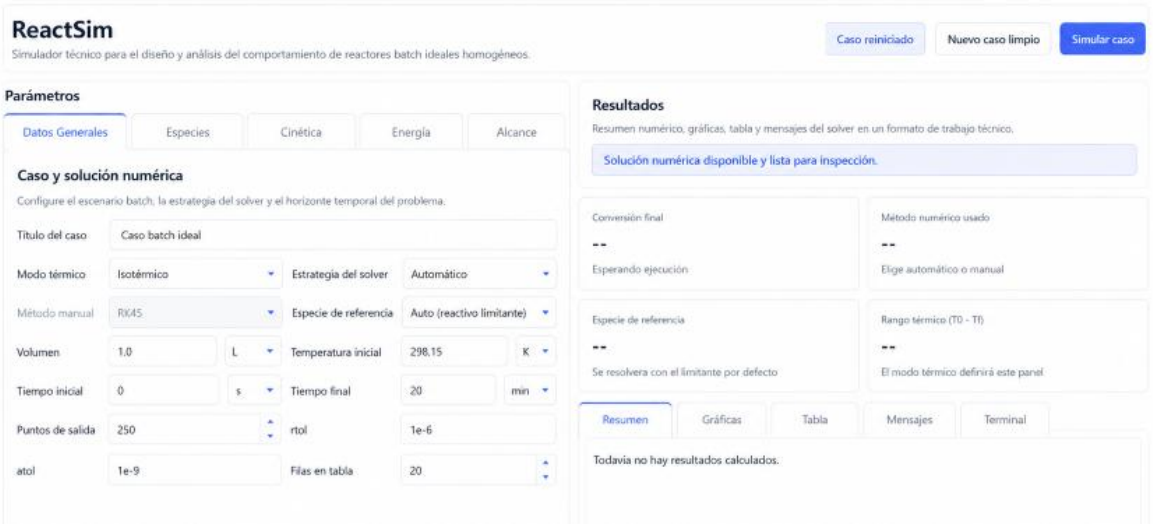
ReactSim Studio está diseñado para simular reactores batch ideales homogéneos en fase líquida, de volumen constante y con una sola reacción química irreversible. El programa permite trabajar con casos isotérmicos, no isotérmicos adiabáticos y no isotérmicos con chaqueta a temperatura constante.

F.1. Vista General de la Interfaz

La ventana principal se divide en dos áreas de trabajo. En el panel izquierdo se configuran los parámetros del caso mediante pestañas. En el panel derecho se presentan los resultados de la simulación, incluyendo resumen numérico, gráficas, tabla de datos, mensajes del sistema y terminal de cálculo.

Figura F.1.

Vista general de la interfaz de ReactSim Studio



Nota. Elaboración propia.

F.2. Organización de las Pestañas de Entrada

La configuración del caso se realiza mediante cinco pestañas principales.

Tabla F.1.

Configuración general de ReactSim Studio

Pestaña	***Función principal***
Datos Generales	Define el modo térmico, volumen, tiempo, solver y tolerancias.

Especies	Permite ingresar especies, coeficientes estequiométricos y concentraciones iniciales.
Cinética	Define la ley de velocidad, órdenes de reacción y parámetros cinéticos.
Energía	Configura parámetros térmicos para casos no isotérmicos.
Alcance	Presenta las hipótesis y límites principales del simulador.

Nota. Elaboración propia con base en la interfaz de ReactSim Studio v1.0.

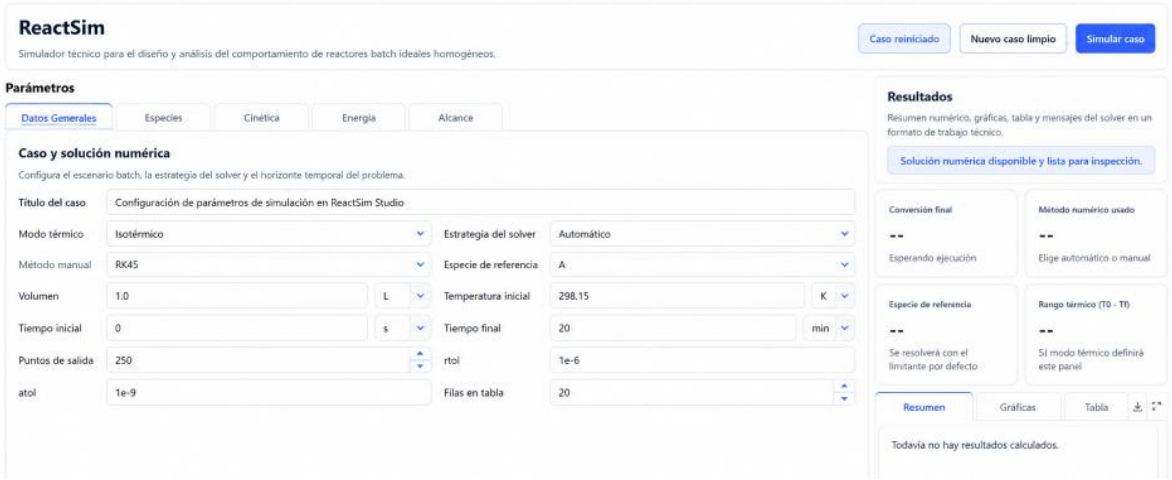
F.3. Creación de un Caso de Simulación

Para construir un caso de simulación, el usuario debe seguir el siguiente orden general:

1. En **Datos Generales**, asignar un título al caso, seleccionar el modo térmico, definir el volumen del reactor, la temperatura inicial, el intervalo de tiempo, los puntos de salida y las tolerancias numéricas.
2. En **Especies**, ingresar las especies químicas participantes. Los reactivos deben colocarse con coeficiente estequiométrico negativo y los productos con coeficiente positivo.
3. En **Cinética**, seleccionar el modelo cinético. El programa permite usar una ley de potencia guiada o una expresión cinética personalizada. Si se activa Arrhenius, deben definirse el factor preexponencial y la energía de activación.
4. En **Energía**, completar los parámetros térmicos cuando el caso sea no isotérmico. Para un caso adiabático se requieren calor de reacción, densidad y calor específico. Para un caso con chaqueta se agregan el coeficiente de transferencia, el área y la temperatura de chaqueta.
5. Ejecutar la simulación mediante el botón **Simular caso**.

Figura F.2.

Configuración de parámetros de simulación en ReactSim Studio



Nota. Elaboración propia.

F.4. Ejecución de la Simulación

Una vez completados los campos requeridos, la simulación se ejecuta con el botón **Simular caso**. Durante este proceso, ReactSim Studio valida las entradas, convierte las unidades al Sistema Internacional, construye el sistema dinámico y resuelve el modelo mediante el solver seleccionado.

Si el caso contiene errores de entrada, el programa no ejecuta la simulación. En su lugar, muestra mensajes de validación y señala los campos que deben corregirse.

F.5. Lectura de Resultados

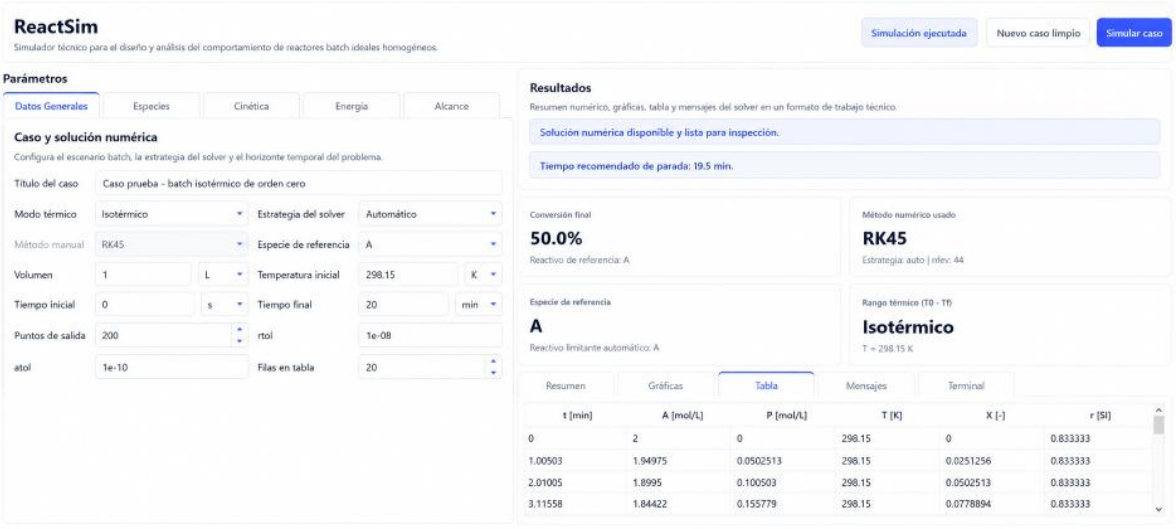
Después de ejecutar una simulación válida, el panel derecho muestra los resultados principales del caso. Las tarjetas superiores presentan información resumida, como conversión final, método numérico utilizado, especie de referencia y comportamiento térmico.

El usuario puede consultar los resultados mediante las siguientes vistas:

Vista	*Contenido*
Resumen	Datos principales del caso y resultados finales.
Gráficas	Perfiles temporales de concentración, conversión, temperatura y otras variables.
Tabla	Valores numéricos generados durante la simulación.
Mensajes	Advertencias, validaciones y estado del solver.
Terminal	Traza del proceso interno de cálculo.

Figura F.3.

Panel de resultados de ReactSim Studio



Nota. Elaboración propia.

F.6. Gestión de Casos y Benchmarks

ReactSim Studio permite guardar y abrir casos mediante archivos con extensión .reactsim. Esta función facilita conservar configuraciones, repetir simulaciones y compartir casos de trabajo.

El programa también incorpora casos de referencia integrados, accesibles desde el menú **Simulación**. Estos casos sirven como ejemplos de referencia para validar el funcionamiento del simulador, familiarizarse con la interfaz y comparar resultados esperados.

F.7. Exportación de Resultados

Cuando existe una simulación válida, el usuario puede exportar un reporte en formato PDF. Este reporte reúne la información principal del caso, los resultados obtenidos, los mensajes del solver y las gráficas disponibles.

Esta función permite conservar evidencia del análisis realizado y facilita la incorporación de resultados en documentos técnicos o académicos.

F.8. Recomendaciones Básicas de Uso

Para obtener resultados consistentes, se recomienda:

- Definir primero el modo térmico del caso.
- Ingresar correctamente la estequiometría, usando signo negativo para reactivos y positivo para productos.
- Usar el modo automático del solver, salvo que se desee comparar métodos numéricos.
- Mantener los valores por defecto de rtol y atol, excepto en estudios de sensibilidad.
- Revisar las pestañas **Mensajes** y **Terminal** después de cada simulación.
- Guardar los casos importantes en formato .reactsim para poder reproducirlos posteriormente.

F.9. Cierre del Anexo

Esta guía resume el flujo básico de uso de ReactSim Studio v1.0. Con ella, el usuario puede configurar un caso de reactor batch ideal, ejecutar la simulación, interpretar los resultados y utilizar las funciones de guardado, carga, benchmarks y exportaciones disponibles en el programa.

Anexo G. Códigos y Configuración de Casos Equivalentes en Cantera

El presente anexo documenta los scripts utilizados para realizar la contrastación externa entre ReactSim Studio y Cantera 3.2.0. Los casos fueron configurados mediante pseudoespecies Ay P, con el propósito de reproducir condiciones equivalentes a las simuladas en ReactSim Studio y comparar variables finales como concentración, conversión y temperatura.

Las pseudoespecies empleadas no representan sustancias químicas reales, sino una configuración simplificada para verificar la consistencia numérica de los resultados frente a un entorno externo de simulación.

G.1. Entorno de Ejecución

Las simulaciones externas fueron ejecutadas en un entorno Conda independiente con Cantera 3.2.0 instalado.

```
Bash                                                    Console

conda activate ct-env
python -c "import cantera as ct; print(ct.__version__)"
```

Resultado obtenido:

```
Bash                                                    Console

3.2.0
```

G.2. Caso Equivalente Isotérmico de Segundo Orden

El primer caso correspondió a una reacción irreversible:

$$A \rightarrow P$$

Con cinética:

$$r = kC_A^2$$

Tabla G.1.

Datos principales del caso

Parámetro	***Valor***
C_{A0}	$1.0\,\frac{mol}{L}$
C_{P0}	$0.0\,\frac{mol}{L}$

V	$1.0\ L$
T	$298.15\ K$
t_f	$600\ s$
k	$3.33333 \times 10^{-6}\ \frac{m^3}{mol \cdot s}$

Nota. Elaboración propia.

G.2.1. Archivo reactsim_second_order.yaml

```
description: |
  Pseudo-caso equivalente al caso ReactSim Studio:
  batch isotermico de segundo orden.
  Reaccion A => P con cinetica r = k * C_A^2.

units:
  length: m
  time: s
  quantity: mol
  activation-energy: J/mol

phases:
- name: gas
  thermo: ideal-gas
  kinetics: gas
  species: [A, P]
  reactions: all
  state:
    T: 298.15 K
    P: 2.478819e6 Pa

species:
- name: A
  composition: {H: 1}
  thermo:
    model: constant-cp
    T0: 298.15 K
    h0: 0.0 J/mol
    s0: 0.0 J/mol/K
    cp0: 30.0 J/mol/K

- name: P
  composition: {H: 1}
  thermo:
    model: constant-cp
    T0: 298.15 K
    h0: 0.0 J/mol
    s0: 0.0 J/mol/K
    cp0: 30.0 J/mol/K

reactions:
- equation: A => P
  rate-constant:
    A: 3.33333e-06 m^3/mol/s
    b: 0.0
    Ea: 0.0 J/mol
  orders:
    A: 2.0
```

G.2.2. **Script** run_cantera_second_order.py

```
import numpy as np
import cantera as ct

T0 = 298.15
CA0_mol_m3 = 1000.0
CP0_mol_m3 = 0.0
V = 0.001
tf = 600.0
output_points = 200

Ctot_kmol_m3 = (CA0_mol_m3 + CP0_mol_m3) / 1000.0
P0 = Ctot_kmol_m3 * ct.gas_constant * T0

gas = ct.Solution("reactsim_second_order.yaml")
gas.TPX = T0, P0, {"A": 1.0, "P": 0.0}

reactor = ct.IdealGasReactor(gas, energy="off", clone=False)
reactor.volume = V

net = ct.ReactorNet([reactor])

times = np.linspace(0.0, tf, output_points)

idx_A = gas.species_index("A")
idx_P = gas.species_index("P")

CA = []
CP = []
T = []

for t in times:
    net.advance(float(t))
    conc_mol_m3 = reactor.phase.concentrations * 1000.0

    CA.append(conc_mol_m3[idx_A])
    CP.append(conc_mol_m3[idx_P])
    T.append(reactor.T)

CA = np.array(CA)
CP = np.array(CP)
T = np.array(T)

X = (CA0_mol_m3 - CA) / CA0_mol_m3

print(f"Cantera version = {ct.__version__}")
print(f"P0 = {P0:.6f} Pa")
print(f"Tiempo final = {tf:.6f} s")
print(f"CA final = {CA[-1]:.6f} mol/m3 = {CA[-1]/1000:.6f} mol/L")
print(f"CP final = {CP[-1]:.6f} mol/m3 = {CP[-1]/1000:.6f} mol/L")
print(f"X final = {X[-1]:.6f}")
print(f"T final = {T[-1]:.6f} K")
```

Comando de ejecución:

Bash Console

```
python run_cantera_second_order.py
```

Salida final obtenida:

```
Bash                                                    Console

Cantera version = 3.2.0
CA final = 333.333559 mol/m3 = 0.333334 mol/L
CP final = 666.666441 mol/m3 = 0.666666 mol/L
X final = 0.666666
T final = 298.150000 K
```

G.3. Caso Equivalente Adiabático Basado en Fogler

El segundo caso correspondió a una reacción irreversible:

$$A \rightarrow B$$

Con cinética dependiente de la temperatura:

$$r = k_0 \exp\left(-\frac{E_a}{RT}\right) C_A$$

Tabla G.2.

Datos principales del caso

Parámetro	***Valor***
C_{A0}	$2.380239521 \frac{mol}{L}$
C_{P0}	$0.0 \frac{mol}{L}$
V	$22.712\ L$
T	$297.222\ K$
t_f	$838.867\ s$
k	$4.7077 \times 10^9\ s^{-1}$
E_a	$75319.8 \frac{J}{mol}$
Condición térmica	Adiabática

Nota. Elaboración propia.

G.3.1. **Archivo** reactsim_fogler.yaml

```
description: |
  Pseudo-caso equivalente al benchmark Fogler 9-1 de ReactSim Studio.
  Reaccion A => P, adiabatico, volumen constante, primer orden.

units:
  length: m
  time: s
  quantity: mol
  activation-energy: J/mol

phases:
- name: gas
  thermo: ideal-gas
  kinetics: gas
  species: [A, P]
  reactions: all
  state:
    T: 297.222 K
    P: 5.884e6 Pa

species:
- name: A
  composition: {H: 1}
  thermo:
    model: constant-cp
    T0: 298.15 K
    h0: 0.0 J/mol
    s0: 0.0 J/mol/K
    cp0: 1764.4221867 J/mol/K

- name: P
  composition: {H: 1}
  thermo:
    model: constant-cp
    T0: 298.15 K
    h0: -88245.27 J/mol
    s0: 0.0 J/mol/K
    cp0: 1764.4221867 J/mol/K

reactions:
- equation: A => P
  rate-constant:
    A: 4.7077e9 1/s
    b: 0.0
    Ea: 75319.8 J/mol
```

G.3.2. **Script** run_cantera_fogler.py

```
import numpy as np
import cantera as ct

T0 = 297.222
CA0_mol_m3 = 2380.239521
CP0_mol_m3 = 0.0
V = 0.022712
tf = 838.867
output_points = 300

Ctot_kmol_m3 = (CA0_mol_m3 + CP0_mol_m3) / 1000.0
P0 = Ctot_kmol_m3 * ct.gas_constant * T0

gas = ct.Solution("reactsim_fogler.yaml")
gas.TPX = T0, P0, {"A": 1.0, "P": 0.0}

reactor = ct.IdealGasReactor(gas, energy="on", clone=False)
```

```
reactor.volume = V

net = ct.ReactorNet([reactor])

times = np.linspace(0.0, tf, output_points)

idx_A = gas.species_index("A")
idx_P = gas.species_index("P")

CA = []
CP = []
T = []

for t in times:
    net.advance(float(t))
    conc_mol_m3 = reactor.phase.concentrations * 1000.0

    CA.append(conc_mol_m3[idx_A])
    CP.append(conc_mol_m3[idx_P])
    T.append(reactor.T)

CA = np.array(CA)
CP = np.array(CP)
T = np.array(T)

X = (CA0_mol_m3 - CA) / CA0_mol_m3

print(f"Cantera version = {ct.__version__}")
print(f"P0 = {P0:.6f} Pa")
print(f"Tiempo final = {tf:.6f} s")
print(f"CA final = {CA[-1]:.6f} mol/m3 = {CA[-1]/1000:.6f} mol/L")
print(f"CP final = {CP[-1]:.6f} mol/m3 = {CP[-1]/1000:.6f} mol/L")
print(f"X final = {X[-1]:.6f}")
print(f"T final = {T[-1]:.6f} K")
```

Comando de ejecución:

Bash Console

```
python run_cantera_fogler.py
```

Salida final obtenida:

Bash Console

```
Cantera version = 3.2.0
CA final = 1154.467340 mol/m3 = 1.154467 mol/L
CP final = 1225.772181 mol/m3 = 1.225772 mol/L
X final = 0.514979
T final = 323.099921 K
```

G.4. Cierre del Anexo

La documentación presentada en este anexo permite respaldar el procedimiento seguido para la contrastación externa entre ReactSim Studio y Cantera. Los archivos de configuración, scripts de ejecución y salidas obtenidas evidencian que los casos comparativos fueron definidos bajo condiciones equivalentes a las

simuladas en ReactSim Studio, utilizando pseudoespecies y parámetros cinéticos ajustados al modelo evaluado. De esta manera, el anexo complementa los resultados presentados en el cuerpo del documento y fortalece la reproducibilidad del proceso de validación externa.

Asimismo, los resultados obtenidos en Cantera muestran concordancia con los valores reportados por ReactSim Studio para las variables finales de concentración, conversión y temperatura. Por tanto, la información aquí presentada constituye un respaldo técnico adicional para la discusión de resultados, al demostrar que el simulador desarrollado produce respuestas coherentes frente a un entorno externo de simulación química, dentro del alcance definido para los casos equivalentes analizados.