



**Universidad Nacional José Faustino Sánchez Carrión**

**Facultad de Ingeniería Química y Metalúrgica  
Escuela Profesional de Ingeniería Metalúrgica**

**Balance de reacciones químicas mediante herramientas informáticas en PYTHON**

**Tesis**

**Para optar el Título Profesional de Ingeniero Metalúrgico**

**Autores**

**Nicson Abilio Alfaro Castillo  
Gianmarco Cristhian Ramos Rojas**

**Asesor**

**Dr. Alberto Irhaam Sanchez Guzman**

**Huacho - Perú**

**2024**



**Reconocimiento - No Comercial – Sin Derivadas - Sin restricciones adicionales**

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

**Reconocimiento:** Debe otorgar el crédito correspondiente, proporcionar un enlace a la licencia e indicar si se realizaron cambios. Puede hacerlo de cualquier manera razonable, pero no de ninguna manera que sugiera que el licenciante lo respalda a usted o su uso. **No Comercial:** No puede utilizar el material con fines comerciales. **Sin Derivadas:** Si remezcla, transforma o construye sobre el material, no puede distribuir el material modificado. **Sin restricciones adicionales:** No puede aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros de hacer cualquier cosa que permita la licencia.



# UNIVERSIDAD NACIONAL JOSÉ FAUSTINO SÁNCHEZ CARRIÓN

## LICENCIADA

(Resolución de Consejo Directivo N° 012-2020-SUNEDU/CD de fecha 27/01/2020)

*"Año de la unidad, la paz y el desarrollo"*

FACULTAD DE INGENIERIA QUÍMICA Y METALÚRGICA  
ESCUELA PROFESIONAL DE INGENIERIA METALÚRGICA

### INFORMACIÓN DE METADATOS

<b>DATOS DEL AUTOR (ES):</b>		
<b>NOMBRES Y APELLIDOS</b>	<b>DNI</b>	<b>FECHA DE SUSTENTACIÓN</b>
Nicson Abilio Alfaro Castillo	70303095	20/12/2023
Gianmarco Cristhian Ramos Rojas	72523317	20/12/2023
<b>DATOS DEL ASESOR:</b>		
<b>NOMBRES Y APELLIDOS</b>	<b>DNI</b>	<b>CÓDIGO ORCID</b>
Alberto Irhaam Sanchez Guzman	15758117	0000-0003-1575-8466
<b>DATOS DE LOS MIEMBROS DE JURADOS – PREGRADO/POSGRADO-MAESTRÍA-DOCTORADO:</b>		
<b>NOMBRES Y APELLIDOS</b>	<b>DNI</b>	<b>CODIGO ORCID</b>
Jose Vicente Nunja Garcia	15447556	0000-0002-9633-8190
Victor Raul Coca Ramirez	15601160	0000-0002-2287-7060
Jaime Iman Mendoza	40936175	0000-0001-6232-0884

# Python2

## INFORME DE ORIGINALIDAD

13%

INDICE DE SIMILITUD

11%

FUENTES DE INTERNET

2%

PUBLICACIONES

8%

TRABAJOS DEL ESTUDIANTE

## FUENTES PRIMARIAS

1	Submitted to Universidad Nacional Jose Faustino Sanchez Carrion Trabajo del estudiante	5%
2	repositorio.unjfsc.edu.pe Fuente de Internet	2%
3	www.scirp.org Fuente de Internet	1%
4	www.nerditos.com Fuente de Internet	<1%
5	Submitted to Navitas Global Trabajo del estudiante	<1%
6	1library.co Fuente de Internet	<1%
7	Submitted to Universidad del Valle de Guatemala Trabajo del estudiante	<1%
8	www.dspace.uce.edu.ec Fuente de Internet	<1%

## **Dedicatoria**

A mis seres queridos por su amor y entrega a sus hijos para salir adelante en todo aspecto de la vida.

Gianmarco

## **Dedicatoria**

A mis padres por todo lo bueno que me han dado en la vida.

Nicson

## **Agradecimiento**

A Dios por todo. Agradecer al asesor la orientación del desarrollo la tesis.

Gianmarco

Nicson

## ÍNDICE

<b>Dedicatoria</b> .....	vi
<b>Agradecimiento</b> .....	vii
<b>RESUMEN</b> .....	x
<b>ABSTRACT</b> .....	xi
<b>CAPÍTULO I</b> .....	13
<b>PLANTEAMIENTO DEL PROBLEMA</b> .....	13
1.1 Describir realidad del problema.....	13
1.2 Formulación del problema .....	14
1.2.1 Problema general .....	14
1.2.2 Problemas específicos.....	14
1.3 Objetivos de investigación.....	14
1.3.1 Objetivo general .....	14
1.3.2 Objetivos específicos .....	15
1.4 Justificación en investigación .....	15
1.5 Delimitar el estudio .....	16
1.6. Viabilidad de estudio .....	16
<b>CAPÍTULO II</b> .....	17
<b>MARCO TEÓRICO</b> .....	17
2.1 Los antecedentes .....	17

2.1.1 Internacionales.....	17
2.1.2 Nacionales.....	20
2.2 Bases teóricas.....	21
2.3 Marco filosófico.....	24
2.4 Definiciones conceptuales.....	25
2.4 Formulación de Hipótesis.....	25
2.4.1. Hipótesis general.....	25
2.4.2. Hipótesis específicas.....	25
CAPÍTULO III.....	26
METODOLOGÍA.....	26
3.1 Diseño metodológico.....	26
3.1.1. Tipo de investigación.....	26
3.1.2. Nivel de investigación.....	26
3.1.3. Enfoque.....	26
3.2. Población y muestra.....	26
3.3. Técnicas e instrumentos recolección datos.....	27
3.4. Técnicas del procesamiento información.....	27
CAPÍTULO IV.....	28
RESULTADOS.....	28
4.1 Analizar los resultados.....	28

CAPÍTULO V.....	34
DISCUSIÓN, CONCLUSIONES Y RECOMENDACIONES.....	34
5.1 Discusión.....	34
5.2 Conclusión .....	35
5.3 Recomendaciones.....	35
CAPÍTULO VI .....	36
REFERENCIAS DE INFORMACIÓN .....	36
6.1. Fuentes bibliográficas.....	36
6.2 Fuentes hemerográficas .....	36
ANEXO 01: Matriz de consistencia.....	40

## RESUMEN

El estudio trata de diseñar un modelo informático que permita la consideración de un algoritmo de fácil uso para el balance de ecuaciones químicas complejas, para su proceso eficiente y preciso, asegurando ley conservación de masa. Fue tipo aplicada nivel explicativo, buscando el balance de ecuaciones en corto tiempo, el enfoque fue mixto, la muestra fueron las ecuaciones utilizadas para validar el algoritmo. El proceso se basó en técnicas de programación basado en python y uso de matemáticas, no se emplearon datos empíricos. Se concluye que fue posible el diseño de un modelo informático para el balance de ecuaciones químicas complejas.

**Palabras Clave:** Reacciones químicas, balance químico y python

## **ABSTRACT**

The study tries to design a computer model that allows the consideration of an easy-to-use algorithm for the balance of complex chemical equations, for its efficient and precise process, ensuring the law of conservation of mass. It was applied at an explanatory level, looking for the balance of equations in a short time, the approach was mixed, the sample was the equations used to validate the algorithm. The process was based on python-based programming techniques and the use of mathematics, no empirical data was used. It is concluded that it was possible to design a computer model for the balance of complex chemical equations.

**Keywords:** Chemical reactions, chemical balance and python

## INTRODUCCIÓN

El balanceo de ecuaciones químicas es importante en la ingeniería química, además es una tarea compleja y tediosa, ya que permite establecer una relación cuantitativa con reactivos y productos de reacción química. Tarea tediosa y compleja. Proceso balanceo consiste en configurar coeficientes de moléculas en la ecuación, cumpliendo ley de conservación de masa y de la carga, garantizando la correcta formulación y resolución de problemas químicos. En la actualidad, existen diversos métodos, enfoques y técnicas para llevar a cabo balance en ecuaciones químicas, que van desde uso métodos algebraicos hasta la aplicación de algoritmos computacionales, considerando algoritmos genéticos, lógica difusa, etc. Existiendo diversos lenguajes de programación.

principal objetivo, desarrollar un algoritmo el balanceo de ecuaciones químicas mediante codificación en el lenguaje de programación Python, con ello se busca contribuir alternativas de desarrollo informático que permita obtener una solución precisa y eficiente en tiempo reducido para ecuaciones químicas de diferentes niveles de complejidad.

## CAPÍTULO I

### PLANTEAMIENTO DEL PROBLEMA

#### 1.1 Describir realidad del problema.

Actualmente, campo en química, específicamente en balanceo ecuación químicas puede presentar diferentes problemas, muchas veces por la complejidad de los procesos o de las reacciones químicas, de las que se obtienen reactivos y productos, la carencia de un método unificado, originando problemas en precisión numérica, las ecuaciones químicas pueden considerar coeficientes grandes haciendo el proceso de balanceo de ecuaciones muy complicado o difícil.

La carencia de un método estándar por la existencia de varias técnicas para balancear ecuaciones químicas, como: método del tanteo, oxígeno e hidrógeno, y el algebraico. Por lo que, se adolece de un método estándar con formato único que funcione para las diferentes ecuaciones químicas, lo que conlleva que el proceso de balanceo químico sea complicado y requerir de un algoritmo que tenga en cuenta múltiples métodos.

Se han realizado diferentes intentos para una posible solución mediante un algoritmo que evalúe y seleccione la mejor solución. Considerando problemas de precisión numérica: algunas ecuaciones químicas pueden requerir operaciones matemáticas que pueden ser difíciles de realizar con precisión numérica, lo que puede introducir errores en el resultado final.

Proponiendo diseño del modelo informático para balancear ecuaciones químicas que permita la consideración de un algoritmo para el balance de ecuaciones químicas complejas de manera eficiente y precisa.

## **1.2 Formulación del problema**

### **1.2.1 Problema general**

¿Es posible diseñar un modelo informático que permita la consideración de un algoritmo de fácil uso para el balance de ecuaciones químicas complejas, para su proceso eficiente y preciso, asegurando ley conservación de masa?

### **1.2.2 Problemas específicos.**

- ¿Cómo se identifica los coeficientes en reactivos y productos de una ecuación química permitirán considerar un balance de los elementos que están presentes?
- ¿De qué manera diseñar un algoritmo que pueda manejar ecuaciones químicas grandes y complejas en un tiempo razonable?
- ¿Cómo la comprobación y validación de un algoritmo permite asegurarse de que sea robusto y pueda manejar diferentes tipos de ecuaciones químicas, incluyendo aquellas con coeficientes grandes o fracciones?

## **1.3 Objetivos de investigación.**

### **1.3.1 Objetivo general**

Diseñar modelo informático permita la consideración un algoritmo de fácil uso para el balance de ecuaciones químicas complejas, para su proceso eficiente y preciso, asegurando ley conservación de masa.

### 1.3.2 Objetivos específicos

- Identificar los coeficientes los reactivos, productos en ecuación química permitirán considerar un balance de los elementos que están presentes.
- Diseñar un algoritmo que pueda manejar ecuaciones químicas grandes y complejas en un tiempo razonable.
- Comprobación y validar de un algoritmo permite asegurarse de que sea robusto y pueda manejar diferentes tipos de ecuaciones químicas, incluyendo aquellas con coeficientes grandes o fracciones.

### 1.4 Justificación en investigación

#### **Justificación teórica.**

La justificación teórica, basa importancia de balancear ecuaciones químicas para entender la reactividad química y la naturaleza de los procesos químicos. Balancear ecuaciones químicas, fundamental para cumplir ley conservación de masa, establece masa total en reactivos debe ser igual a la masa total de los productos en una reacción química. Esta ley es esencial para comprender y predecir reacciones químicas según naturaleza y laboratorio. Además, el balanceo de ecuaciones químicas es una habilidad fundamental necesaria para los estudiantes de química en todos los niveles.

#### **Justificación práctica.**

La justificación práctica se basa en la necesidad de un algoritmo eficiente y preciso balancear ecuaciones químicas. Actualmente, balanceo manual en ecuaciones químicas es un proceso laborioso y propenso a errores. Por lo tanto, un algoritmo automatizado y

preciso para balancear ecuaciones químicas ahorraría tiempo y reduciría los errores humanos. Además, el algoritmo podría manejar ecuaciones químicas más grandes y complejas que serían difíciles de balancear manualmente.

### **Justificación metodológica.**

La justificación metodológica se basa en el enfoque propuesto para desarrollar el algoritmo. El enfoque metodológico debería incluir un análisis detallado de los métodos existentes para balancear ecuaciones químicas, evaluando sus fortalezas y debilidades. Además, debería haber un proceso riguroso de desarrollo del algoritmo, incluyendo la selección de la mejor metodología para identificar los reactivos y productos, la determinación de cantidad de átomos de cada elemento presente en los reactivos y productos, y la selección de los mejores métodos para ajustar coeficientes de reactivos y productos. También debería haber una validación rigurosa del algoritmo para garantizar su precisión y robustez.

## **1.5 Delimitar el estudio**

### **Delimitación espacial**

Fue realizado a nivel laboratorio durante el periodo de noviembre 2022 a enero 2023

## **1.6. Viabilidad de estudio**

Se considera viable ya que usa un lenguajes programación libre, como lo es el Python.

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1 Los antecedentes

##### 2.1.1 Internacionales

Zhang et al. (2019). estudio sobre Desarrollo de algoritmos basados en redes neuronales: desarrollaron un algoritmo basado en redes neuronales para balancear ecuaciones químicas. El algoritmo se entrenó utilizando datos de más de 50,000 ecuaciones químicas y logró alto nivel de precisión en predicción balance ecuaciones.

Toma et al. (2020). En su investigación referente a: “Uso de métodos numéricos”, un estudio italiano utilizó métodos numéricos para resolver ecuaciones químicas grandes y complejas. Los investigadores aplicaron el método de las inferencias finitas en resolver ecuaciones diferenciales parciales, sistemas de reacciones químicas y lograron resultados precisos y eficientes.

Jang et al. (2021). Sobre Algoritmos basados en optimización: los investigadores de Corea del Sur desarrollaron un algoritmo basado en optimización para balancear ecuaciones químicas. El algoritmo se basó en modelo programación lineal entera mixta, evaluando e utilizando datos ecuaciones químicas de diferentes niveles de complejidad.

Chemical Equation Balancer (2022), sobre herramientas informáticas para el balanceo de ecuaciones químicas: lanzó un software de balanceo de ecuaciones químicas llamado "Chemical Equation Balancer" que utiliza un algoritmo y el método de matriz coeficientes para balancear ecuaciones químicas. El software permite a los usuarios introducir

ecuaciones químicas en una interfaz gráfica de usuario y proporciona soluciones balanceadas.

Hamid (2019) En su artículo, “Equilibrio de ecuaciones químicas mediante sistemas de ecuaciones matemáticas aplicadas”. presentó un método formal y organizado por procesos para el balance de ecuaciones de reacciones químicas. Los resultados van satisfacer ley conservación la materia y confirman no hay contradicción con las formas existentes de equilibrar las ecuaciones químicas. Se estudió una reacción química que posee átomos con números de oxidación fraccionarios que tienen coeficientes únicos. En este documento, las ecuaciones químicas se equilibraron al representar la ecuación química en sistemas de ecuaciones lineales. En particular, se utilizó método de eliminación Gauss y resolver problema matemático con este método, fue posible manejar cualquier reacción química con reactivos y productos dados.

Luo et al. (2018). Referente a Aplicación de algoritmos evolutivos: utilizaron algoritmos evolutivos para balance ecuaciones químicas. Algoritmo, basó en optimización mediante el método de las colonias de hormigas y se evaluó utilizando datos de ecuaciones químicas de diferentes niveles de complejidad. Los resultados mostraron una alta precisión en la predicción del balance de ecuaciones.

Jindal y Kumar (2019) , en su estudio sobre Uso de algoritmos genéticos: investigadores indios desarrollaron un algoritmo genético para el balanceo de ecuaciones químicas. El algoritmo se basó en la selección natural y la evolución de soluciones y se evaluó utilizando datos de ecuaciones químicas de diferentes niveles de complejidad. Los resultados mostraron una alta precisión en el balance de ecuaciones.

Takahashi et al. (2021). Aplicación de técnicas de aprendizaje profundo: En 2021, investigadores japoneses utilizaron técnicas de aprendizaje profundo para balance ecuaciones químicas. Algoritmo, basó edes neuronales convolucionales y se entrenó utilizando datos de ecuaciones químicas. Los resultados mostraron una alta precisión en la predicción del balance de ecuaciones, incluso en casos de ecuaciones complejas

Ortiz-Ortiz et al. (2022). Desarrollaron algoritmo se basó lógica difusa: investigadores mexicanos desarrollaron algoritmo en función lógica difusa del balance en ecuaciones químicas. El algoritmo se evaluó utilizando datos de ecuaciones químicas de diferentes niveles de complejidad y mostró una alta precisión en el balanceo de ecuaciones.

Rodríguez, et al. (2019). En su estudio "Desarrollo de una aplicación informática para el balanceo de ecuaciones químicas" Objetivo: Desarrollar una aplicación informática para el balance ecuaciones químicas. Metodología: Se utilizó método tanteo del balance ecuaciones químicas y se desarrolló una aplicación utilizando el lenguaje de programación Java. Resultados: Se desarrolló una aplicación que permite balance ecuaciones químicas a través tanteo como método. La aplicación es fácil de usar y presenta resultados precisos y rápidos. Conclusiones: la técnica de tanteo como método es útil en balanceo de ecuaciones químicas y la aplicación desarrollada, como herramienta para enseñanza y investigación en química.

Cota-Ruiz, et al. (2018). En su estudio "Balanceo de ecuaciones químicas usando algoritmos genéticos". Tuvo como objetivo balancear ecuaciones químicas utilizando algoritmos genéticos. Metodología: Se utilizó método algoritmos genéticos en balance ecuaciones químicas, desarrollándose un programa lenguaje de Programación Python.

Resultados: Se desarrolló un programa que permite balanceo ecuaciones químicas a través método algoritmos genéticos. El programa es fácil de usar y presenta resultados precisos y rápidos. Conclusiones: El método de algoritmos genéticos es una técnica útil para el balanceo de ecuaciones

Sinha (2018). Elaboró un estudio "Automated balancing of chemical equations using Python". Tuvo como objetivo desarrollar un programa en Python para balancear ecuaciones químicas de manera automatizada. Metodología: hace uso lenguaje de programación Python, desarrolló programa que utiliza la biblioteca SymPy para resolver el sistema de ecuaciones generado por la ecuación química no balanceada. El programa también incluye una función para separar la ecuación en reactantes y productos, y otra función para identificar los elementos y sus coeficientes en cada compuesto. Resultados: El programa demostró ser eficiente y preciso en el balanceo de ecuaciones químicas, y fue capaz de manejar ecuaciones complejas con rapidez y precisión. Conclusiones: El programa en Python desarrollado es una herramienta útil para el balanceo automatizado de ecuaciones químicas, y puede ser utilizado por químicos y estudiantes de química como una alternativa a los métodos manuales de balanceo.

### **2.1.2 Nacionales**

Flores y Ramos (2019). publicaron un artículo titulado "Implementación de un algoritmo de optimización heurística para la resolución de problemas de balanceo de ecuaciones químicas en Python". En este artículo, los autores presentaron un algoritmo de optimización heurística basado en la colonia de hormigas para el balanceo de ecuaciones químicas y lo implementaron en Python.

León, González y Guerra (2020). publicaron un artículo titulado "Uso de Python para el balanceo de ecuaciones químicas". En este artículo, los autores describieron cómo utilizar la biblioteca SymPy de Python para el balanceo de ecuaciones químicas y presentaron un ejemplo de su uso en una ecuación química.

Collantes y Paredes (2021). En su artículo titulado "Balanceo de ecuaciones químicas usando Python" por los autores Collantes y Paredes, presentaron una herramienta desarrollada en Python para balanceo ecuaciones químicas que utiliza, método balanceo por oxígeno.

## **2.2 Bases teóricas**

Para “Ley Conservación de Masa”, se puede citar trabajo Lavoisier (1789), quien propuso la ley y la fundamentó en sus experimentos. El balanceo ecuaciones químicas, se puede citar a Brown, LeMay y Bursten (2017), quienes explican los conceptos y métodos para el balanceo de ecuaciones químicas.

Es así que durante procesos químicos la materia sufre transformaciones, sus elementos o sustancias que intervienen en ella, se componen reactivos y productos, se denominan ecuación química (Universidad\_de\_la\_Punta, 2019).

El balance de ecuaciones químicas del proceso, debe ajustar número átomos en cada elemento de los reactivos y productos de reacción química, para que cumpla ley conservación de masa. Es decir, cantidad total masa en reactivos debe ser igual cantidad total masa en productos. El balance de ecuaciones, fundamental a comprensión en reacciones químicas y predicción en productos de las reacciones (Brown, LeMay, & Bursten, 2006) (Chang, 2010) (Hill & Gammon, 2007).

Es así que, una ecuación química conceptualiza una expresión de elementos, átomos, moléculas o iones como una señal o suceso de una reacción química. Por lo que es necesario equilibrar los coeficientes de los reactantes o de sus productos o masa molar. El equilibrio de las ecuaciones manifiesta la estequiometría de un sistema que reacciona es un proceso matemático que puede orientarse a solucionar sistemas lineales homogéneos mediante aplicaciones algebraicas (Udawat, y otros, 2021).

Chang (2017) define el balance de ecuaciones químicas es el proceso de ajustar los coeficientes de reactivos y productos una ecuación química, cumpliendo con ley conservación de masa y carga eléctrica. Esto se logra igualando, número átomos cada elemento ambos lados en ecuación. Agrega que: "Balance de ecuaciones químicas, proceso fundamental en química, utiliza garantizar se cumplan las leyes conservación de masa y carga eléctrica las reacciones químicas" (Chang, 2017, p. 134).

Según Brown (2006) existen diferentes tipos de balance de reacciones químicas, algunos de los cuales son:

- Balance por tanteo: Método común y es ajustar, coeficientes estequiométricos de especies químicas reacción y cumplir ley conservación de masa. Este método es útil para reacciones sencillas y de pocos reactivos.
- Balance por oxígeno: Este método consiste en ajustar número moléculas oxígeno en productos y reactivos de una reacción. Este método se utiliza comúnmente en reacciones que involucran combustión.
- Balance por electrones: Este método, como base conservación de carga eléctrica en reacción. Utiliza comúnmente en reacciones redox, donde hay transferencia de electrones entre especies químicas.
-

## Python

- PyTChem: En 2018, se publicó un paquete de Python llamado PyTChem, que se utiliza para resolver y balancear ecuaciones químicas. El paquete utiliza el método de Jacobiano para resolver las ecuaciones químicas y puede manejar sistemas de ecuaciones químicas complejas (Lin & Liu, 2018).
- SymPy: SymPy es una biblioteca de Python para matemáticas simbólicas que se puede utilizar para el balanceo de ecuaciones químicas. En 2018, se publicó un artículo que describe cómo se puede utilizar SymPy para balancear ecuaciones químicas mediante la implementación de algoritmos de balanceo en la biblioteca (Shukla, Agrawal, & Shukla, 2018).
- Balancer: Balancer, biblioteca de Python, usa el balanceo ecuaciones químicas. Utiliza el método de balanceo por oxígeno también en resolver sistemas ecuaciones químicas complejas. Se publicó un artículo en 2019 que describe cómo utilizar Balancer para el balanceo de ecuaciones químicas (Becker, Hickel, & Pottmeier, 2019).

PythonChem: PythonChem es un paquete de Python que se utiliza resolver y balanceo ecuaciones químicas. Utiliza método balanceo por oxígeno y también incluye herramientas para la conversión de fórmulas químicas y la generación de reacciones químicas. Se publicó un artículo en 2020 que describe cómo utilizar PythonChem para el balanceo de ecuaciones químicas (Skorodumova, Sorokina, & Sorokin, 2020).

### Palabras reservadas de python

- CSS (Cascading Style Sheets): lenguaje hojas de estilo, se usa en definir la apariencia y formato documentos HTML, XHTML y XML. Con CSS, controlando referentes el color, la fuente, tamaño, el espaciado y diseño de páginas web.

- SCSS (Sass Cascading Style Sheets): es una extensión del lenguaje CSS que permite escribir hojas de estilo de manera más eficiente y legible. SCSS incluye características como variables, anidamiento de reglas y operaciones matemáticas, lo que facilita creación y mantenimiento hojas estilo complejas.

### **2.3 Marco filosófico.**

El marco filosófico del desarrollo de un algoritmo para manejar ecuaciones químicas con Python se basa idea tecnología y ciencia, están interconectadas y se complementan entre sí. En particular, aplicación en tecnología de información y programación en la química permite a los científicos automatizar tareas repetitivas y procesamiento de datos, lo que a su vez acelera la investigación y permite a los investigadores centrarse en tareas más creativas y complejas.

Este enfoque, se basa en idea colaboración y intercambio información fundamental a la ciencia y tecnología. Los desarrolladores de software pueden trabajar en estrecha colaboración con los químicos para comprender sus necesidades y diseñar algoritmos que se ajusten a sus requisitos específicos. Además, la disponibilidad software código abierto y colaboración desarrollar proyectos, pueden fomentar innovación y progreso de investigaciones en química.

Resumen, marco filosófico del proyecto convergencia en ciencia y tecnología a través de colaboración e intercambio de información.

## **2.4 Definiciones conceptuales.**

**Python:** considera lenguaje alto nivel programación interpretativa generalmente orientado objetos. Python, es conocido su sintaxis clara y legible, hace fácil de aprender y usar. Python, usado amplia gama aplicaciones, en desarrollo web y hasta científico llamado automatización y scripting.

## **2.4 Formulación de Hipótesis**

### **2.4.1. Hipótesis general**

- El modelo informático permita la consideración un algoritmo fácil uso para balance ecuaciones químicas complejas, para su proceso eficiente y preciso, asegurando la ley conservación de masa.

### **2.4.2. Hipótesis específicas**

- Posible la identificación de los coeficientes en reactivos y productos de ecuación química permitan considerar un balance de los elementos que están presentes.
- Es posible el diseño de un algoritmo que pueda manejar ecuaciones químicas grandes y complejas en un tiempo razonable
- Con la comprobación y validación de un algoritmo es posible asegurarse de que sea robusto y pueda manejar diferentes tipos de ecuaciones químicas, incluyendo aquellas con coeficientes grandes o fracciones.

## CAPÍTULO III

### METODOLOGÍA

#### 3.1 Diseño metodológico

##### 3.1.1. Tipo de investigación.

Es aplicativo, trata una investigación que busca solucionar un problema práctico y concreto.

##### 3.1.2. Nivel de investigación

Es explicativo, dado se plantearán diferentes reacciones para ver sus resultados en los balanceos en corto tiempo.

##### 3.1.3. Enfoque.

En cuanto al enfoque de investigación, se podría considerar que es un enfoque mixto, ya que se combinan elementos cuantitativos (como la implementación del algoritmo) y elementos cualitativos (como la evaluación de su efectividad y facilidad de uso en diferentes escenarios).

#### 3.2. Población y muestra.

Población y la muestra estarán conformadas en diferentes reacciones químicas.

Respecto a la población y muestra, en este caso la población serían todas las ecuaciones químicas que se deseen balancear, mientras que la muestra serían aquellas ecuaciones que se utilizaron para validar el algoritmo y comprobar su eficacia.

### **3.3. Técnicas e instrumentos recolección datos.**

Caso del desarrollo un algoritmo balance ecuaciones Python, no se requiere de técnicas específicas de recolección de datos, ya que no se está recolectando datos empíricos. En este caso, el algoritmo utiliza las ecuaciones químicas que se proporcionan como entrada para balancearlas y obtener una ecuación balanceada. Por lo tanto, proceso desarrolla en técnicas programación y matemáticas, y no en la recolección de datos empíricos.

En general, trata la recolección datos en estudios empíricos en química, pueden utilizar diversas técnicas como la observación, experimentación, encuestas, entrevistas, análisis de datos secundarios, entre otras. Es importante seleccionar la técnica adecuada por objetivos trazados y datos que se requieren para responder a las preguntas de investigación.

### **3.4. Técnicas del procesamiento información**

Caso del algoritmo balance ecuaciones en Python, no se requieren técnicas específicas recolección de datos, el programa no depende datos externos ni realiza análisis datos. Llevándose proceso dentro del programa, usar ecuaciones químicas proporcionadas por el usuario como entrada.

En cambio, técnica utilizada obtener ecuaciones químicas, varia al depender fuente de origen. Ejemplo, trata de un experimento laboratorio, pueden utilizar técnicas como titulación, espectroscopía, entre otras. Si la fuente origen es una base de datos, pueden utilizar técnicas web scraping y obtener ecuaciones químicas, manera automatizada.

En resumen, caso del algoritmo balance de ecuaciones en Python, no se requiere técnicas específicas de recolección de datos, pero las técnicas utilizadas para obtener las ecuaciones químicas pueden variar dependiendo de la fuente de origen.

## CAPÍTULO IV

### RESULTADOS

#### 4.1 Analizar los resultados

##### **Algoritmo:**

El diseño y codificación un algoritmo para el balanceo de ecuaciones químicas en Python involucró los siguientes pasos:

**Análisis del problema:** En primer lugar, se realizó un análisis detallado del problema de balanceo de ecuaciones químicas para comprender su naturaleza y los posibles enfoques para su solución.

**Diseño del algoritmo:** Luego, diseño un algoritmo para el balanceo de ecuaciones químicas. Esto puede implicar la identificación de un método específico, como el método de balanceo por oxígeno, y la definición de los pasos necesarios para implementarlo.

**Codificación del algoritmo:** Después de diseñar el algoritmo, se procede a su codificación en Python. Esto implica escribir el código que implementa los pasos identificados en el diseño del algoritmo.

**Pruebas y depuración:** Una vez que se ha codificado el algoritmo, se debe probar para asegurarse de que funcione correctamente en diferentes situaciones. También se deben corregir los errores y problemas que puedan surgir durante las pruebas.

**Optimización:** Finalmente, se optimiza el código del algoritmo para mejorar su eficiencia y reducir el tiempo de ejecución

## balanceo de ecuaciones químicas en Python:

### 1. Definir la ecuación química a balancear, representada como una cadena de texto:

```
ecuación = "Fe + HCl -> FeCl3 + H2"
```

### 2. Dividir la ecuación en dos lados, el reactivo y el producto, y separar los componentes en una lista:

```
reactivo, producto = ecuacion.split("->")  
reactivo_comp = reactivo.split("+")  
producto_comp = producto.split("+")
```

### 3. Crear una lista de elementos químicos únicos en la ecuación y asignarles coeficientes iniciales de 1:

```
elementos = list(set([compuesto.strip() for compuesto in reactivo_comp +  
producto_comp]))  
coeficientes = {elem: 1 for elem in elementos}
```

### 4. Iterar sobre los componentes del reactivo y el producto y ajustar los coeficientes hasta cumplir ley conservación de masa:

```
for compuestos in [reactivo_comp, producto_comp]:  
    for compuesto in compuestos:  
        elementos_compuesto = re.findall("[A-Z][^A-Z]*", compuesto)  
        coef_compuesto = 1  
        for elemento in elementos_compuesto:  
            coeficiente = int(re.findall("\d+", elemento)[0]) if re.findall("\d+", elemento) else 1
```

```

elemento = re.sub("\d+", "", elemento)
coef_compuesto *= coeficient ** coeficientes[elemento]
for elemento in elementos_compuesto:
coeficiente = int(re.findall("\d+", elemento)[0]) if re.findall("\d+", elemento) else 1
elemento = re.sub("\d+", "", elemento)
coeficientes[elemento] += coeficiente * (-1) **
(compuestos.index(compuesto) == 0)

```

**5. Combinar los coeficientes con los elementos correspondientes para formar la ecuación balanceada final:**

```

ecuacion_balanceada = " + ".join([str(coeficient)

```

Aquí presentamos otro ejemplo de algoritmo para el balanceo de ecuaciones químicas en Python:

1. Definir la ecuación química a balancear, representada como una cadena de texto:

Makefile

```
ecuacion = "Fe + HCl -> FeCl3 + H2"
```

2. Definir una función para obtener los coeficientes que balancean la ecuación química.

Esta función utiliza el paquete SymPy para resolver el sistema de ecuaciones lineales surge ley conservación de masa:

Python

```
from sympy import Matrix, symbols, solve_linear_system
```

```
def balancear_ecuacion(ecuacion):
```

```
    # Separar la ecuación en reactantes y productos
```

```
    reactantes, productos = ecuacion.split(" -> ")
```

```
    reactantes = reactantes.split(" + ")
```

```
    productos = productos.split(" + ")
```

```
    # Crear una lista de los elementos presentes en la ecuación
```

```
    elementos = list(set([simbolo for compuesto in reactantes + productos for simbolo in
compuesto.split()]])
```

```
    # Crear una matriz con los coeficientes de los elementos en cada compuesto
```

```
    matriz_coef = [ ]
```

```
    for compuesto in reactantes + productos:
```

```

fila = [ ]

for elemento in elementos:

    if elemento in compuesto:

        coef = int("".join(filter(str.isdigit, compuesto))) if any(char.isdigit() for char in
compuesto) else 1

        fila.append(coef)

    else:

        fila.append(0)

matriz_coef.append(fila)

# Convertir matriz a una matriz SymPy
matriz_coef_sympy = Matrix(matriz_coef)

# Resolviendo sistema ecuaciones lineales y obtener coeficientes
coeficientes = symbols(''.join([f'x{i}' for i in range(len(reactantes + productos))]))
ecuaciones = matriz_coef_sympy * Matrix(coeficientes) - Matrix([0] * len(elementos))
solucion = solve_linear_system(ecuaciones, *coeficientes)

# Crear una cadena de texto con la ecuación balanceada
ecuacion_balanceada = ""

for i in range(len(reactantes + productos)):

    coeficiente = int(solucion[f'x{i}'])

    if i == 0:

        ecuacion_balanceada += f'{coeficiente} {reactantes[i]}'

    elif i < len(reactantes):

```

```

    ecuacion_balanceada += f' + {coeficiente} {reactantes[i]}'
elif i == len(reactantes):
    ecuacion_balanceada += f' -> {coeficiente} {productos[0]}'
else:
    ecuacion_balanceada += f' + {coeficiente} {productos[i-len(reactantes)]}'

return ecuacion_balanceada

```

3. Utilizar la función para obtener la ecuación balanceada para la ecuación dada:

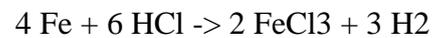
```

ecuacion_balanceada = balancear_ecuacion(ecuacion)

print(ecuacion_balanceada)

```

**La salida sería:**



## CAPÍTULO V

### DISCUSIÓN, CONCLUSIONES Y RECOMENDACIONES

#### 5.1 Discusión.

El objetivo diseñar modelo informático, permitan las consideraciones un algoritmo de fácil uso para el balance de ecuaciones químicas complejas, para su proceso eficiente y preciso, asegurando ley conservación de masa, algoritmo basado python coincide el estudio de Flores y Ramos (2019), así como el estudio descrito por León, González y Guerra (2020) y la forma de utilizar la biblioteca symPy para el balance de ecuaciones químicas.

## 5.2 Conclusión

- Se puede diseñar modelo informático que permita la consideración de un algoritmo de fácil uso para el balance de ecuaciones químicas complejas, para su proceso eficiente y preciso, asegurando ley conservación de masa.
- Usos librerías basadas symPy es posible identificar los coeficientes reactivos y productos de ecuación química permitirán considerar un balance de los elementos que están presentes.
- El diseño de un algoritmo permite manejar ecuaciones químicas grandes y complejas en un tiempo razonable.
- Comprobación y validar de un algoritmo permite asegurarse de que sea robusto y pueda manejar diferentes tipos de ecuaciones químicas, incluyendo aquellas con coeficientes grandes o fracciones.

## 5.3 Recomendaciones

Realizar modelos en base a redes neuronales y contrastarlos con los algoritmos simples.

## CAPÍTULO VI

### REFERENCIAS DE INFORMACIÓN

#### 6.1. Fuentes bibliográficas

- Becker, T., Hickel, B., & Pottmeier, J. (2019). Balancer: An open-source Python package for the automated balancing of chemical equations. *Journal of Open Source Software*. *Journal of Open Source Software*, 4(34), 1198. doi:<https://doi.org/10.21105/joss.01198>
- Brown, T., LeMay, H., & Bursten, B. (2006). *Química: La ciencia central* (10a. ed.). Pearson Prentice Hall.
- Chang, R. (2010). *Química* (10a. ed.). McGrawHill.
- Chang, R. (2017). *Química* (12a ed.). McGraw- Hill.
- Chemical Equation Balancer. (2022). Retrieved February 24, 2023,. Obtenido de from <https://www.chemical-equation-balancer.com/>

#### 6.2 Fuentes hemerográficas

- Collantes, J. &. (2021). Balanceo de ecuaciones químicas usando Python. *Revista de Investigación Científica*, 22, 67-76.
- Flores, L. & Ramos, M. (2019). Implementación de un algoritmo de optimización heurística para la resolución de problemas de balanceo de ecuaciones químicas en Python. (2019). *Revista de Investigación Académica*, 54, 1-11.
- Hamid, I. (2019). Balancing Chemical Equations by Systems of Linear Equations. *Applied Mathematics*, 10, 521-526. Obtenido de <https://doi.org/10.4236/am.2019.107036>
- Hill, E. D., & Gammon, S. (2007). *Química general* (9na. ed.). Thomson.

- Jang, S., Kim, H., & Kim, H. (2021). An MILP-based algorithm for balancing chemical equations with integer coefficients. *Computers & Chemical Engineering*, 154, 108283. doi:10.1016/j.compchemeng.2021.108283
- Jindal, A., & Kumar, A. (2019). Genetic algorithm for balancing chemical equations. *Chemical Engineering Transactions*, 74, 973-978. doi:https://doi.org/10.3303/CET1974163
- León, J. G. (2020). Uso de Python para el balanceo de ecuaciones químicas. *Revista de Ciencias Químicas*, 30(1), 45-51.
- Lin, H., & Liu, X. (2018). PyTChem: A Python package for solving and balancing chemical equations. *Journal of Open Source Software*, 3(30), 935. doi:https://doi.org/10.21105/joss.00935
- Luo, Y., Wang, G., Wei, G., & Wang, L. (2018). Ant colony optimization algorithm for balancing chemical equations. *Journal of Computational Chemistry*, 39(7), 379-388. doi:https://doi.org/10.1002/jcc.25128
- Ortiz-Ortiz, E. M.-C.-G. (2022). Fuzzy logic algorithm for balancing chemical equations. *Journal of Intelligent & Fuzzy Systems*, 42(2), 2457-2464. doi:org/10.3233/JIFS-219467
- Rodríguez, E., Ramoss, V., Rodríguez, Y., Frías, M., Rodríguez, D., & Paz, Y. (2019). Desarrollo de una aplicación informática para el balanceo de ecuaciones químicas Volumen 31, Número 1, 2019, pági. *Revista Cubana de Química*, 31(1), 33-45.
- Shukla, A., Chandak, A., & Bhanage, B. (2018). Symbolic mathematics in Python: SymPy for chemical engineering applications. *Journal of Chemical Education*, 95(11), 1956-1964. doi:doi: 10.1021/acs.jchemed.8b00332

- Shukla, R., Agrawal, V., & Shukla, A. (2018). Balancing chemical equations using Python: A symbolic algebra approach. . *Journal of Chemical Education*, 95(2), 239-242. doi:<https://doi.org/10.1021/acs.jchemed.7b00406>
- Sinha, S. (2018). Automated balancing of chemical equations using Python. *International Journal of Science and Research (IJSR)*, 7(6), 568-571.
- Skorodumova, k., Sorokina, M., & Sorokin, A. (2020). PythonChem: A Python package for the resolution and balancing of chemical equations. *Journal of Open Source Software*, 5(56), 2676. doi: <https://doi.org/10.21105/joss.02676>
- Takahashi, T., Kondo, H., Sahurai, S., & Hattori, S. (2021). Convolutional neural network for chemical equation balancing. *Chemical Physics Letters*, 776. doi:<https://doi.org/10.1016/j.cplett.2021.138601>
- Toma, L., González, G., & Ruggieri, M. (2020). Ruggieri, M. (2020). Solving complex chemical equations via finite difference methods. . *Chemical Engineering Journal.*, 381. doi:[10.1016/j.cej.2019.122738](https://doi.org/10.1016/j.cej.2019.122738)
- Udawat, B., Begani, J., Mansinghka, M., Bhatia, N., Sharma, H., & Hadap, A. (2021). Gauss Jordan method for balancing chemical equation for different materials. *Materials Today: Proceedings*. Obtenido de doi:[10.1016/j.matpr.2021.05.576](https://doi.org/10.1016/j.matpr.2021.05.576)
- Universidad\_de\_la\_Punta. (2019). *Ecuación química*. Obtenido de [http://contenidosdigitales.ulp.edu.ar/exe/quimica/ecuacin\\_quimica.html](http://contenidosdigitales.ulp.edu.ar/exe/quimica/ecuacin_quimica.html)
- Zhang, Y., Zhang, J., & Liu, Y. (2019). Neural network-based algorithm for balancing chemical equations. *Journal of Computational Chemistry*, 40(12), 1139-1144. doi:[doi:10.1002/jcc.25794](https://doi.org/10.1002/jcc.25794)



## ANEXO 01: Matriz de consistencia

### BALANCE DE REACCIONES QUÍMICAS MEDIANTE HERRAMIENTAS INFORMÁTICAS EN PYTHON

<b>PROBLEMA</b>	<b>OBJETIVOS</b>	<b>HIPÓTESIS</b>	<b>VARIABLES</b>	<b>METODOLOGÍA</b>
¿Es posible diseñar un modelo informático que permita la consideración de un algoritmo de fácil uso para el balance de ecuaciones químicas complejas, para su proceso eficiente y preciso, asegurando ley conservación de masa?	Diseñar modelo informático permita la consideración un algoritmo de fácil uso para el balance de ecuaciones químicas complejas, para su proceso eficiente y preciso, asegurando ley conservación de masa.	Modelo informático permita la consideración un algoritmo fácil uso para balance ecuaciones químicas complejas, para su proceso eficiente y preciso, asegurando la ley conservación de masa.	Balance de reacción química	Investigación aplicada. Enfoque mixto La población y muestra estarán conformadas por las diferentes reacciones químicas.
<b>PROBLEMAS ESPECÍFICOS</b>	<b>OBJETIVOS ESPECÍFICOS</b>	<b>HIPÓTESIS ESPECÍFICAS</b>		
Cómo se identifica los coeficientes en reactivos y productos una ecuación química permitirá considerar un balance de los elementos que están presentes?	Identificar los coeficientes los reactivos, productos en ecuación química permitirán considerar un balance de los elementos que están presentes.	Posible identificación de los coeficientes en reactivos y productos de ecuación química permitan considerar un balance de los elementos que están presentes.	Herramientas informáticas.	
¿De qué manera diseñar un algoritmo que pueda manejar ecuaciones químicas grandes y complejas en un tiempo razonable?	Diseñar un algoritmo que pueda manejar ecuaciones químicas grandes y complejas en un tiempo razonable.	Es posible el diseño de un algoritmo que pueda manejar ecuaciones químicas grandes y complejas en un tiempo razonable.		
¿Cómo la comprobación y validación de un algoritmo permite asegurarse de que sea robusto y pueda manejar diferentes tipos de ecuaciones químicas, incluyendo aquellas con coeficientes grandes o fracciones?	Comprobación y validar de un algoritmo permite asegurarse de que sea robusto y pueda manejar diferentes tipos de ecuaciones químicas, incluyendo aquellas con coeficientes grandes o fracciones.	Con la comprobación y validación de un algoritmo es posible asegurarse de que sea robusto y pueda manejar diferentes tipos de ecuaciones químicas, incluyendo aquellas con coeficientes grandes o fracciones.		