

LA PROGRAMACIÓN COMO HERRAMIENTA ESENCIAL PARA EL INGENIERO QUÍMICO DEL SIGLO XXI

PROGRAMMING AS AN ESSENTIAL TOOL FOR THE 21ST-CENTURY CHEMICAL ENGINEER

Jesús Andrés Arzola Flores*
Fernanda Saviñon Flores
Miguel Angel García Castro
Fausto Díaz Sánchez
Fidel Aaron Maruri Valderrabano

Facultad de Ingeniería Química, BUAP. Av. San Claudio S/N, San Manuel,
C.P. 72560. Puebla, México.

*Autor de correspondencia: jesus.arzolaflores@correo.buap.mx.
Tel. (221) 203 8067

maria.savinon@alumno.buap.mx
miguel.garciacastro@correo.buap.mx
fausto.diazs@alumno.buap.mx
mv223470624@alm.buap.mx

<https://orcid.org/0000-0001-9839-982X>
<https://orcid.org/0009-0005-9242-0655>
<https://orcid.org/0000-0003-4459-873X>
<https://orcid.org/0000-0002-0084-0427>
<https://orcid.org/0009-0009-0182-7492>

Año 11 No. 32

Recibido: 1/abril/2025

Aprobado: dd/mmmm/2025

Publicado: 30 de agosto/2025

Resumen

La programación se está volviendo una herramienta clave para los ingenieros de hoy, por lo que para los ingenieros químicos no es la excepción. Frente al boom de tecnologías propias de la industria 4.0 y 5.0, tales como el internet de las cosas, el big data y por supuesto, la inteligencia artificial, saber programar ya no es opcional. En este artículo reflexionamos sobre la necesidad de integrar la programación en la formación de los futuros ingenieros químicos, no solo en asignaturas como métodos numéricos, simulación de procesos o control, sino también a lo largo de toda el currículo del programa de estudios de ingeniería química. Además, mostramos algunos paquetes de Python que pueden ser muy útiles en diferentes materias de ingeniería química. Finalmente, proponemos algunas estrategias y consejos para motivar el aprendizaje de la programación tanto para estudiantes como para docentes, con el objetivo de fomentar un cambio en el proceso de enseñanza-aprendizaje de esta disciplina.

Palabras clave: Ingeniería Química, Programación, Python, Simulación de Procesos, Control de Procesos

Abstract

Programming is becoming a key tool for today's engineers, and chemical engineers are no exception. With the rise of Industry 4.0 and 5.0 technologies—such as the Internet of Things, big data, and, of course, artificial intelligence—knowing how to program is no longer optional. In this article, we reflect on the need to integrate programming into the education of future chemical engineers, not only in subjects like numerical methods, process, simulation, or control, but also throughout the entire chemical engineering curriculum. We also highlight several Python packages that can be very useful across different courses. Finally, we propose strategies and tips to encourage programming education among both students and educators, aiming to promote a shift in the teaching and learning approach to this essential discipline.

Keywords: Chemical Engineering, Programming, Python, Process Simulation, Process Control

¿Es importante la programación para los ingenieros químicos?

Hoy en día, dentro del campo de la ingeniería química se ha experimentado un cambio drástico impulsado por las nuevas tecnologías digitales, tales como el internet de las cosas (IoT), el big data y por supuesto, por la inteligencia artificial (Cedeño y col., 2024; Paredes y col., 2025). Lo interesante de esto, es que estas tecnologías tienen como base a la programación, por lo que esta herramienta se ha convertido en una habilidad que los ingenieros químicos del siglo XXI deben conocer y dominar, principalmente porque les permitirá ser competitivos en este nuevo auge de la industria 4.0 y 5.0 (Acosta y col., 2025; Mereles, 2025; Novoa, 2025).

Actualmente, no hay un área donde la inteligencia artificial no haya permeado, por lo que aquí la interrogante principal es ¿qué tan capacitados se encuentran actualmente los ingenieros químicos para enfrentar esta nueva ola de oportunidades?, y a la par, surge otra pregunta ¿qué tanta importancia le están dandolas facultades y escuelas de ingeniería química de México al desarrollo de habilidades de programación en los docentes y sobre todo en la comunidad estudiantil?

Como es sabido, la programación no es un tópico fundamental que los estudiantes de ingeniería química desarrollan durante su formación profesional. Aunque existen bastantes herramientas de software dentro de la currícula de ingeniería química, muchas de ellas suelen ser de licencia y, en la mayoría de las veces, poco accesibles para los estudiantes. En raras ocasiones los estudiantes pueden experimentar con ellas para proponer nuevos algoritmos o modificar las metodologías ya existentes dentro de las plataformas. Aunque hoy en día hay muchos lenguajes de programación que son de libre acceso, en la mayoría de las ocasiones no se utilizan en las aulas. Lo anterior, debido a la falta de capacitación o bien porque existen herramientas que no usan como tal un lenguaje de programación, solo se ocupan para resolver los problemas estándar de los cursos de ingeniería química. En consecuencia, aquí surge la siguiente pregunta

¿Estas herramientas son suficientes para que los estudiantes resuelvan problemas reales?, por ejemplo, aquellos en donde es necesario manipular grandes volúmenes de datos (Rodríguez, 2025).

Actualmente, gracias al auge de herramientas como la inteligencia artificial, en México la implementación de la programación en los planes de estudio de Ingeniería Química ha experimentado un constante crecimiento, debido a que tanto los docentes como los estudiantes están reconociendo su valor como una habilidad fundamental para los ingenieros químicos del siglo XXI. Universidades tanto públicas como privadas han incorporado asignaturas que buscan introducir a los estudiantes en la programación, sin embargo, en muchas ocasiones la falta de capacitación e interés por parte de la academia obstaculiza la creación de un ecosistema en donde la programación sea una herramienta utilizada en el día a día.

Algunas de estas asignaturas del currículo de Ingeniería Química que pueden servir como catalizador del uso de la programación por parte de los estudiantes y docentes:

- **Métodos Numéricos:** Esta asignatura es pilar dentro de la formación de todo ingeniero químico, ya que se centra en el uso de técnicas numéricas para resolver problemas complejos que surgen en todas las áreas de la ingeniería química, tales como la solución de sistemas de ecuaciones algebraicas no lineales, problemas de optimización no lineal, sistemas de ecuaciones diferenciales ordinarias y parciales, diferenciación e integración numérica, etc (Beers, 2007). Esta asignatura es fundamental y puede ser un curso en donde se aplique la programación, por ejemplo, los estudiantes podrían recibir un curso previo de fundamentos de programación y posteriormente, aplicar la programación en problemas numéricos. En esta asignatura, los estudiantes pueden emplear lenguajes de programación de alto nivel

abiertos tales como Python, R o Julia para implementar diferentes técnicas numéricas y con ello, mejorar su comprensión sobre el desarrollo de algoritmos computacionales.

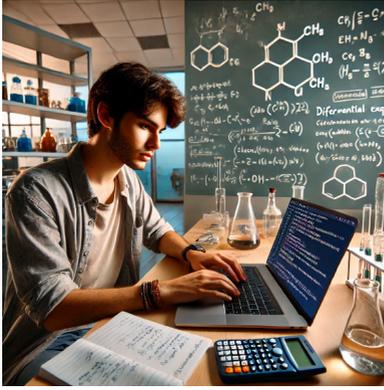


Figura 1. Imagen generada con DALL-E. Prompt: Un estudiante universitario mexicano de Ingeniería Química sentado en un escritorio en un laboratorio o sala de estudio, concentrado en la pantalla de su laptop. En la pantalla se ve código en Python relacionado con métodos numéricos (por ejemplo, ecuaciones diferenciales o resolución de matrices). El estudiante viste ropa casual (playera y jeans) y lleva consigo una calculadora científica y un cuaderno con anotaciones de fórmulas químicas y ecuaciones matemáticas. En el fondo, se aprecian detalles relacionados con la ingeniería química, como matraces, tubos de ensayo o esquemas de procesos químicos en una pizarra. La atmósfera transmite estudio y concentración; el estudiante podría estar tecleando activamente o revisando los resultados de la ejecución del código en su computadora. Iluminación clara, mostrando el ambiente académico y el interés del estudiante en la resolución del problema.

- Simulación de Procesos:** Esta asignatura utiliza gemelos digitales de licencia como Aspen Plus, HYSYS, ChemCAD, así como simuladores de código abierto como COCO Simulator y DWSIM, con el fin de modelar, analizar y optimizar diversos procesos químicos. Aunque muchas de estas herramientas pueden considerarse como “cajas negras”, hoy en día, gracias al auge del modelado híbrido es posible vincular estos simuladores con herramientas como Matlab y

Python, para combinar modelos de aprendizaje automático o aprendizaje profundo con modelos mecanísticos para optimizar el rendimiento de los procesos, validación de modelos, sintonización de parámetros de control, entre otras aplicaciones (Chaves y col., 2016; Sansana y col., 2021; Haydary, 2019).



Figura 2. Imagen generada con DALL-E. Prompt: Un estudiante mexicano universitario de Ingeniería Química, de unos 20 años, sentado frente a una computadora de escritorio en un moderno laboratorio de ingeniería. En la pantalla se aprecia un software de simulación de procesos (similar a Aspen Plus o Aspen HYSYS) mostrando un diagrama de flujo de una planta química. El estudiante viste bata de laboratorio blanca abierta sobre ropa casual, lleva gafas de seguridad sobre la frente y sostiene un cuaderno con apuntes de fórmulas y balances de materia y energía. En el fondo se observan equipos de laboratorio, matraces, columnas de destilación a escala y otros instrumentos de ingeniería química. La atmósfera es luminosa, profesional y muestra la concentración del estudiante mientras analiza los datos y configura el modelo de la planta química.

- Control de Procesos:** Dentro de la formación de los estudiantes de ingeniería química, también se busca que tengan las herramientas para implementar sistemas de control básicos y avanzados, con el objetivo de que tengan la habilidad de controlar procesos y con esto, asegurar la calidad y seguridad de los mismos (Stephanopoulos,

1984; Smith & Corripio, 2005). En esta asignatura, es de vital importancia que los estudiantes utilicen la programación, ya que al emplear esta herramienta desarrollarán su capacidad de abstracción a la hora de programar, por ejemplo, un controlador PID. Aquí, los aprendientes no solo pueden emplear la programación para implementar sistemas de control estándar, si no también, combinarlas con herramientas de inteligencia artificial para identificación de sistemas, sintonizar sistemas de control y por supuesto, diagnosticar y predecir fallas.



Figura 3. Imagen generada con DALL-E. Prompt: Un estudiante universitario mexicano de Ingeniería Química trabajando en un laboratorio o sala de control industrial, sentado frente a una computadora. En la pantalla se ven líneas de código y diagramas de control de procesos químicos (pueden incluir bloques de PID, esquemas de lazo de control, etc.). El estudiante viste una bata de laboratorio abierta, con ropa informal debajo, y tiene un cuaderno con anotaciones técnicas al lado del teclado. En el fondo, se observan equipos de proceso a escala de laboratorio o pantallas de simulación industrial (columnas de destilación, reactores, intercambiadores de calor), reflejando un entorno de ingeniería química. El estudiante parece concentrado, ajustando parámetros o analizando gráficas de respuesta de control. La atmósfera general es ordenada, con iluminación blanca de laboratorio, mostrando un ambiente profesional y académico dedicado al desarrollo y validación de un sistema de control para un proceso químico

Algunas herramientas para aplicar la programación en ingeniería química

Es importante destacar que la programación no solo debería encontrarse presente en estas asignaturas específicas, sino que también se puede integrar dentro del plan de estudios de todas las materias formativas de Ingeniería Química. Un ejemplo claro se encuentra dentro de los cursos de fisicoquímica, en donde se pueden utilizar herramientas de Python como el paquete thermo para calcular propiedades de sustancias puras o de mezclas. Similarmente, en cinética química, donde también existen librerías de Python como chemPy para simular diferentes tipos de reacciones. Incluso, para la materia de fenómenos de transporte, con librerías como ht y fluids, que son muy útiles para estudiar el transporte de calor y masa, etc. A continuación, se muestra una tabla donde se exponen las librerías de Python que podrían ser utilizadas en algunas asignaturas del área de formación general en ingeniería del plan de estudios de Ingeniería Química de la Benemérita Universidad Autónoma de Puebla (<https://ingenieriaquimica.buap.mx/?q=content/licenciatura-en-ingenier%C3%ADa-da-qu%C3%ADmica>). En dicha tabla se observan las asignaturas, las librerías de Python que se pueden utilizar, así como algunas aplicaciones.

Tabla 1. Librerías de Python que pueden utilizarse para las materias básicas

Asignatura	Herramientas de Python	Aplicaciones
Álgebra para ingeniería	NumPy https://numpy.org/	<ul style="list-style-type: none"> - Cálculos de álgebra lineal básica (inversión y operaciones con matrices). - Solución de sistemas de ecuaciones (p. ej., balances de materia).
	SymPy https://www.sympy.org/en/index.html	<ul style="list-style-type: none"> - Cálculo simbólico de sistemas de ecuaciones en balances o reacciones químicas. - Generación de soluciones exactas para modelos cinéticos.
Química general y Taller de Introducción a Ingenierías del Área Química	ChemPy https://github.com/bjodah/chempy	<ul style="list-style-type: none"> - Cálculos de estequiometría y equilibrio químico. - Evaluación de constantes de equilibrio, pH y cinéticas simples. - Simulación de reacciones con parámetros químicos básicos.
	Mendelev https://pypi.org/project/mendelev/	<ul style="list-style-type: none"> - Acceso rápido a propiedades de los elementos de la tabla periódica (números atómicos, masas, radios, electronegatividad, etc.). - Consultas de datos químicos fundamentales para problemas de química general.
Química orgánica I y II	RDKit https://www.rdkit.org/	<ul style="list-style-type: none"> - Representación y edición de estructuras orgánicas (SMILES, Mol, etc.). - Análisis de subestructuras, búsqueda de fragmentos funcionales y generación de descriptores moleculares.
	Open Babel https://openbabel.org/	<ul style="list-style-type: none"> - Conversión entre distintos formatos de archivo químico (SDF, SMILES, Mol2, etc.). - Limpieza y optimización geométrica inicial de moléculas orgánicas.
Cálculo I, II y III	SymPy https://www.sympy.org/	<p>Cálculo simbólico: derivadas, integrales, límites, series, factorización.</p> <p>Herramientas para simplificar expresiones, matrices simbólicas y cálculo con polinomios.</p>
	SciPy https://scipy.org/	<ul style="list-style-type: none"> - Integración numérica (módulo <code>scipy.integrate</code> para integrales ordinarias y ecuaciones diferenciales).
Fisicoquímica I, II y III	thermo https://thermo.readthedocs.io/	<ul style="list-style-type: none"> - Propiedades termodinámicas de sustancias puras y mezclas (densidad, Cp, entalpía, etc.). - Ecuaciones de estado (Peng-Robinson, SRK, etc.) y correlaciones para VLE (Equilibrio líquido-vapor). - Estimación de propiedades fisicoquímicas y cálculos de ingeniería de procesos.
	cantera https://cantera.org/	<ul style="list-style-type: none"> - Simulación de cinética química y termodinámica para mezclas de gases, líquidos y sólidos. - Modelado de reacciones en combustión, análisis de flama y equilibrio químico. - Útil para procesos donde la velocidad de reacción y el equilibrio son relevantes.

Física I y II	PyDy https://pydy.readthedocs.io/	<ul style="list-style-type: none"> - Modelado y simulación de sistemas mecánicos usando la formulación de mecánica analítica (Lagrangiana o Newtoniana). - Generación de ecuaciones de movimiento de manera simbólica y numérica. - Animación y visualización de sistemas multi-body (péndulos, engranajes).
	VPython https://vpython.org/	<ul style="list-style-type: none"> - Creación de simulaciones 3D interactivas en física (por ejemplo, órbitas planetarias, movimiento de partículas, colisiones). - Visualización fácil de vectores y objetos en un entorno tridimensional. - Muy útil para enseñanza y prácticas de laboratorio virtual.
Química Analítica y Análisis Instrumental	SpectroChemPy https://www.spectrochempy.fr/	<ul style="list-style-type: none"> - Procesamiento y análisis de datos espectroscópicos (IR, NMR, Raman, UV-Vis). - Herramientas de quimiometría (preprocesamiento, correcciones de línea base, etc.). - Análisis multivariante para interpretación de resultados (PCA, MCR, etc.).
	PyOpenMS https://pyopenms.readthedocs.io/	<ul style="list-style-type: none"> - Interfaz en Python para la suite OpenMS de análisis de espectrometría de masas (MS). - Procesamiento de espectros MS, búsqueda de picos, alineación de espectros, identificación de compuestos.
Probabilidad y Estadística y Diseño de Experimentos	statsmodels https://www.statsmodels.org/	<ul style="list-style-type: none"> - Regresiones (lineal, logística, GLM) con análisis detallado (p-values, intervalos de confianza, diagnósticos). - ANOVA de uno o varios factores, verificación de suposiciones. - Series de tiempo (ARIMA, SARIMAX) y otras pruebas estadísticas profundas.
	pingouin https://pingouin-stats.org/	<ul style="list-style-type: none"> - Biblioteca ligera para estadística: ANOVAs, tests no paramétricos, regresión lineal, correlaciones. - Fácil de usar, con resultados estilo "tabla" listos para reportar. - Ideal para análisis rápidos en diseño de experimentos con técnicas clásicas.
Ecuaciones diferenciales	SciPy https://scipy.org/	<ul style="list-style-type: none"> - Funciones de integración numérica para ecuaciones diferenciales ordinarias: <code>scipy.integrate.odeint</code>, <code>solve_ivp</code>. - Métodos de Runge-Kutta, Adams-Bashforth, BDF, etc. - Muy útil para sistemas de EDOs en modelado, simulación y análisis.
	FiPy https://www.ctcms.nist.gov/fipy/	<ul style="list-style-type: none"> - EDPs (ecuaciones diferenciales parciales) con método de volúmenes finitos. - Pensado para problemas de difusión, electroquímica, reacciones químicas, flujos, etc. - Incluye ecuaciones estacionarias y transitorias en mallas regulares.

Análisis numérico y programación	SciPy https://scipy.org/	<ul style="list-style-type: none"> - Colección principal de métodos numéricos en Python: optimización, integración, interpolación, ecuaciones diferenciales, etc. - Submódulos <code>scipy.optimize</code>, <code>scipy.integrate</code>, <code>scipy.interpolate</code>, <code>scipy.linalg</code>, <code>scipy.fftpack</code>, entre otros.
	SymPy https://www.sympy.org/	<ul style="list-style-type: none"> - Cálculo simbólico de derivadas, integrales, series, solución exacta de ecuaciones. - Puede combinarse con métodos numéricos para validación o para generar expresiones simplificadas antes de discretizar.

Cada una de las librerías o paquetes de la tabla anterior son de código abierto, es decir, pueden ser consultadas en su página web para saber con precisión cómo se utilizan las librerías, cómo se generan los códigos y cómo se realizan los cálculos. Además, es posible adaptar las subrutinas para resolver problemas específicos, lo cual abre un gran abanico de posibilidades para trabajar con la programación en ingeniería química.

Asimismo, existen múltiples librerías de Python que podrían ser utilizadas en cada una de las asignaturas del área de formación básica en ingeniería para sentar las

bases y generalizar el uso de la programación. Esto, con el fin de que los estudiantes puedan estar lo suficientemente familiarizados al llegar a los semestres avanzados. Por lo tanto, es crucial que los estudiantes comiencen a tener un acercamiento con los fundamentos de programación desde el primer semestre de su formación superior. En la siguiente tabla, se muestran algunas librerías de Python que pueden ser utilizadas en las asignaturas formativas del plan de estudios de Ingeniería Química de la Benemérita Universidad Autónoma de Puebla.

Tabla 2. Librerías de Python que pueden utilizarse para las materias formativas.

Asignatura	Herramienta de Python	Aplicaciones
Balances de Materia y Energía	NumPy https://numpy.org/	<ul style="list-style-type: none"> - Manipulación de arreglos y operaciones matemáticas. - Base para resolver sistemas de ecuaciones lineales en balances (por ejemplo, $A \cdot x = b$). - Soporte para cálculos vectorizados y manejo de grandes volúmenes de datos.
	SciPy https://scipy.org/	<ul style="list-style-type: none"> - Métodos de optimización, integración y resolución de ecuaciones diferenciales (útil en balances con acumulación o reacciones). - Submódulo <code>scipy.optimize</code> para solucionar balances no lineales de materia y energía.
	ChemPy https://github.com/bjodah/chempy	<ul style="list-style-type: none"> - Cálculos de estequiometría, equilibrio químico y cinética. - Permite plantear y resolver balances de materia con reacciones químicas, buscando conversiones y selectividades. - Simulación de pH, reacciones ácido-base, etc.
Fenómenos de Transporte I y II	FEniCS https://fenics-project.org/	<ul style="list-style-type: none"> - Método de Elementos Finitos para ecuaciones diferenciales parciales (PDEs) de flujo, calor y masa. - Resolución de problemas de mecánica de fluidos (Navier-Stokes), difusión, convección-difusión, etc. - Potente y escalable (paralelización en HPC).

	<p>FiPy</p> <p>https://www.ctcms.nist.gov/fipy/</p>	<ul style="list-style-type: none"> - Método de Volúmenes Finitos: simulación de difusión, convección-difusión, electroquímica, reactivos, etc. - Muy útil para problemas 1D, 2D y 3D en celdas de malla regular. - Facilidad para modelar flujos y transferencia de calor/masa con reacciones químicas.
	<p>PyVista</p> <p>https://docs.pyvista.org/</p>	<ul style="list-style-type: none"> - Visualización 3D de resultados de simulación (campos de velocidad, temperatura, concentraciones). - Sencilla integración con Fenics, FiPy, etc. para postprocesar mallas y campos en problemas de transporte.
Flujo de fluidos	<p>fluids</p> <p>https://pypi.org/project/fluids/</p>	<ul style="list-style-type: none"> - Cálculos de mecánica de fluidos en tuberías y equipos: factores de fricción (Darcy-Weisbach, Hazen-Williams), pérdida de carga en accesorios, cálculo de caudal en orificios y válvulas. - Estimaciones de propiedades y correlaciones para flujos.
	<p>OpenPNM</p> <p>https://openpnm.org/</p>	<ul style="list-style-type: none"> - Estudio de flujo de fluidos en medios porosos a nivel microscópico (redes de poros). - Modelado de permeabilidad, difusión, transporte reactivo (filtración, catálisis, baterías/pilas de combustible).
	<p>SfePy</p> <p>https://sfepy.org/</p>	<ul style="list-style-type: none"> - Paquete de Elementos Finitos en Python para PDEs (incluyendo ecuaciones de flujo de fluidos). - Permite resolver problemas de mecánica de sólidos, transferencia de calor y acoplarlos con ecuaciones de flujo.
Diseño de Intercambiadores de Calor	<p>ht</p> <p>https://pypi.org/project/ht/</p>	<ul style="list-style-type: none"> - Cálculo de coeficientes de película (convección forzada, natural, condensación y ebullición). - Factores de ensuciamiento, correcciones para intercambiadores y transferencia por radiación.
	<p>thermo</p> <p>https://thermo.readthedocs.io/</p>	<ul style="list-style-type: none"> - Propiedades termodinámicas y de transporte. - Modelado de mezclas de fluidos y su comportamiento en intercambiadores de calor. - Compatibilidad con ecuaciones de estado (Peng-Robinson, SRK).
	<p>fluids</p> <p>https://pypi.org/project/fluids/</p>	<ul style="list-style-type: none"> - Cálculo de coeficientes de transferencia de calor en tuberías y equipos. - Factores de corrección para intercambiadores de calor. - Pérdida de carga y diseño hidráulico complementario.
Cinética y Reactores Homogéneos	<p>ChemPy</p> <p>https://github.com/bjodah/chempy</p>	<ul style="list-style-type: none"> - Cálculo de equilibrio químico y estequiometría. - Simulación de cinética de reacciones homogéneas. - Resolución de ecuaciones diferenciales en modelos cinéticos simples.
	<p>cantera</p> <p>https://cantera.org/</p>	<ul style="list-style-type: none"> - Simulación de reactores homogéneos (PFR, CSTR, batch). - Modelado de cinética química compleja (multicomponente, reacciones en cadena). - Cálculo de propiedades termodinámicas y de equilibrio químico.
	<p>SciPy</p> <p>https://scipy.org/</p>	<ul style="list-style-type: none"> - Resolución de ecuaciones diferenciales en reactores batch, PFR y CSTR (scipy.integrate.solve_ivp). - Optimización de parámetros cinéticos (scipy.optimize).

Procesos de Separación I y II	thermo https://thermo.readthedocs.io/	- Cálculo de propiedades termodinámicas en separación de fases. - Modelado de equilibrios de fases (VLE, LLE, SLE). - Determinación de coeficientes de actividad y ecuaciones de estado
	pyomo http://www.pyomo.org/	- Optimización de procesos de separación (tamaño de columnas, número de etapas). - Modelado algebraico de procesos de adsorción y destilación reactiva.
	Matplotlib https://matplotlib.org/	- Visualización de curvas de equilibrio, perfiles de temperatura y concentración en columnas de separación. - Análisis gráfico de eficiencia y optimización de procesos de separación.
Catálisis y Reactores Heterogeneos	Cantera https://cantera.org/	- Simulación de reacciones heterogéneas en catalizadores sólidos. - Modelado de adsorción, desorción y reacciones superficiales. - Simulación de reactores de lecho fijo y fluidizado.
	ChemPy https://github.com/bjodah/chempy	- Modelado de cinética química en superficies catalíticas. - Simulación de equilibrio químico en reacciones heterogéneas. - Cálculo de parámetros cinéticos a partir de datos experimentales.
	pyomo http://www.pyomo.org/	- Optimización de reactores catalíticos (carga de catalizador, flujo, temperatura). - Modelado algebraico de reactores con múltiples reacciones y etapas.
Investigación de Operaciones	pyomo http://www.pyomo.org/	- Optimización matemática avanzada (lineal, no lineal, entera, estocástica). - Modelado de sistemas complejos en logística, producción y cadenas de suministro.
	PuLP https://coin-or.github.io/pulp/	- Optimización lineal (PL) y entera mixta (MILP). - Modelado de problemas de asignación, transporte, flujo de redes y scheduling. - Compatible con CBC, GLPK, Gurobi, CPLEX.
	CVXPY https://www.cvxpy.org/	- Optimización convexa y cuadrática. - Aplicaciones en portafolios financieros, control de procesos y problemas de diseño óptimo.
Dinámica y Control de Procesos	control https://python-control.readthedocs.io/	- Análisis de sistemas dinámicos (ecuaciones diferenciales, funciones de transferencia, espacio de estados). - Diseño de controladores PID, LQR, MPC.
	gekko https://gekko.readthedocs.io/	- Control predictivo basado en modelos (MPC). - Aplicaciones en optimización de procesos químicos e industriales.

	pyDynamic https://pydynamic.readthedocs.io/	<ul style="list-style-type: none"> - Identificación de sistemas dinámicos. - Modelado basado en datos y ajuste de parámetros en procesos industriales.
Diseño Procesos y Productos Químicos	BioSTEAM https://biosteam.readthedocs.io/	<ul style="list-style-type: none"> - Simulación de procesos biotecnológicos. - Modelado de fermentaciones, purificaciones, producción de biocombustibles y biofármacos.
	DWSIM-Python API https://dwsim.org/	<ul style="list-style-type: none"> - Simulación de procesos químicos completos. - Modelado de reactores, intercambiadores de calor, separaciones y reciclajes. - Equilibrio de fases y balances de masa y energía.
	pyomo http://www.pyomo.org/	<ul style="list-style-type: none"> - Optimización de diseño de procesos químicos. - Modelado de problemas de planificación y síntesis de procesos.
Ingeniería de Servicios Auxiliares	thermo https://thermo.readthedocs.io/	<ul style="list-style-type: none"> - Simulación de sistemas de generación de vapor y agua de proceso. - Modelado de propiedades termodinámicas en calderas y condensadores.
	pyomo http://www.pyomo.org/	<ul style="list-style-type: none"> - Optimización del uso de servicios auxiliares. - Diseño de redes de vapor, aire comprimido y sistemas de distribución de energía térmica.
	CoolProp http://www.coolprop.org/	<ul style="list-style-type: none"> - Cálculo de propiedades de fluidos térmicos (vapor, refrigerantes, aire, agua). - Aplicado en diseño de sistemas de refrigeración y vapor.

Aunque ha existido un gran avance en el desarrollo de lenguajes de programación gratuitos como Python, R o Julia, la mayoría de los docentes y estudiantes de ingeniería química aún no incorporan la programación en su conjunto de habilidades (Sanatinia & Noubir, 2016; Mayer & Bauer, 2015; Bezanson, 2024). Lo anterior se debe en gran manera a que la programación comúnmente se asocia a ingenieros en sistemas o computación. No obstante, en el contexto actual, aprender herramientas de programación, especialmente aquellas gratuitas, abre una gran oportunidad laboral para los futuros ingenieros químicos, ya que les facilitará la comprensión de aquellas tecnologías fundamentadas en la programación, como lo es la inteligencia artificial generativa, la cual ha facilitado el uso de la programación para resolver problemas complejos en ingeniería química.

Es así como se ve reflejada dicha necesidad para realizar correctamente la construcción de “prompts” teniendo conocimientos básicos de programación y, por consiguiente, comprender el código proporcionado por los grandes modelos de lenguaje (LLM, por sus siglas en inglés), (Patil & Gudivada, 2024; Ye, 2023). Hoy en día, existen múltiples lenguajes de programación de acceso libre que pueden ser utilizados dentro de las asignaturas del currículo de Ingeniería Química. En la siguiente tabla se muestra una lista de algunos lenguajes de programación gratuitos, sitios de descarga, tutoriales y posibles aplicaciones en ingeniería química.

Tabla 3. Algunos lenguajes de programación open source que se pueden utilizar en la enseñanza de la ingeniería química.

Lenguaje de programación	Recursos para aprender	Aplicaciones de Ingeniería Química
Python https://www.python.org/downloads/	- Documentación oficial: https://docs.python.org/3/tutorial/ - El libro de Python https://ellibrodepython.com/	- Simulación de procesos - Modelado matemático - Análisis de datos - Automatización de tareas
R https://cran.r-project.org/	- Curso gratuito de R en Datacamp: https://www.datacamp.com/courses/free-introduction-to-r - Documentación oficial: https://www.r-project.org/other-docs.html	- Análisis estadístico de procesos - Modelado de datos experimentales - Visualización de datos
Julia https://julialang.org/downloads/	- Curso gratuito de Julia: https://juliaacademy.com/ - Documentación oficial: https://docs.julialang.org/en/v1/	- Cálculo numérico avanzado - Simulación de reactores y procesos - Optimización de procesos químicos
Octave https://www.gnu.org/software/octave/download.html	- Curso de Octave en Coursera: https://www.coursera.org/learn/matlab-octave-basico - Documentación oficial: https://www.gnu.org/software/octave/doc/interpreter/	- Análisis de ecuaciones diferenciales - Simulación de reactores - Modelado de fenómenos de transporte
Scilab https://www.scilab.org/download/latest	- Tutoriales oficiales de Scilab: https://www.scilab.org/tutorials	- Simulación de procesos químicos - Análisis y optimización de sistemas térmicos - Modelado de transferencia de calor y masa



Figura 4. Imagen generada con DALL-E. Prompt: Una estudiante mexicana de ingeniería química con cabello rizado y gafas trabaja en su laptop en un laboratorio moderno. En la pantalla de su computadora, se observan gráficos de datos, modelos de simulación y líneas de código en Python. Alrededor de ella, hay equipos de laboratorio como reactores, espectrómetros y sensores conectados a una interfaz de inteligencia artificial que analiza datos en tiempo real. La estudiante tiene una expresión de concentración mientras revisa los resultados y ajusta parámetros en su modelo. En el fondo, hay una pizarra con ecuaciones matemáticas y diagramas de procesos químicos.

Las herramientas de código abierto, como se muestra en las tablas anteriores, pueden mejorar significativamente la experiencia de aprendizaje y enseñanza en la ingeniería química. Un buen ejemplo es el uso de Python en la simulación de procesos químicos. Tradicionalmente, los simuladores comerciales como ASPEN Plus y ASPEN HYSYS han sido la opción preferida, pero Python ofrece una alternativa flexible y potente al vincularse con estos simuladores a través de una interfaz COM o la API de ASPEN para Python. Además, Python se puede usar con Microsoft Excel y librerías como pandas para analizar datos de simulación. También existen simuladores de código abierto como DWSIM que se integran directamente con Python, ampliando aún más las posibilidades para la enseñanza y el aprendizaje.

Asimismo, en cuanto al modelado y optimización de reactores, la programación podría ser muy útil, ya que se puede utilizar para resolver sistemas de ecuacio-

nes relacionados con diferentes tipos de reactores. Aquí, los estudiantes pueden explorar cómo pequeños cambios en las condiciones de operación, por ejemplo, cambios de temperatura, presión y concentración, pueden afectar el rendimiento de los reactores. Aparte de eso, los estudiantes de ingeniería química podrían indagar sobre la teoría del caos y su presencia en reactores químicos, el cual es un tópico muy importante y que tampoco se aborda a profundidad dentro de la currícula de ingeniería química.

En general existen muchas áreas de la ingeniería química en donde la programación podría ser muy útil, no obstante, el punto medular es motivar tanto a docentes como a estudiantes en el uso de la programación, mostrando sus ventajas así como la importancia que tendrá en los próximos años.

Algunos consejos

Finalmente, se mencionan algunos consejos tanto para docentes como para estudiantes, con el fin de motivarlos a utilizar la programación en su día a día.

1. En la manera de lo posible, trata de enfocarte en casos prácticos, mostrando ejemplos reales o proyectos en donde la programación haya impactado de forma importante a la industria. Hoy en día ya existen muchos casos de éxito, por lo que esto puede servir de motivación para los estudiantes. En la plataforma de Kaggle encontrarás muchos ejemplos (<https://www.kaggle.com/>)
2. Utiliza herramientas 100% accesibles, tales como Python, R o Julia, ya que son muy fáciles de utilizar y gracias a su gran comunidad de desarrolladores, existen muchos paquetes o complementos que puedes utilizar en tus clases, por ejemplo en el caso de Python, puedes buscar en su repositorio de paquetes (<https://pypi.org/>)
3. Siempre busca fomentar la creatividad de los estudiantes, anímalos a proponer sus propios proyectos, siempre buscando resolver algún problema real. Actualmente, existen plataformas gratuitas en donde puedes obtener datos que te pueden servir durante tus clases, por ejemplo, para realizar análisis de datos. Aquí algunas: Kaggle (<https://www.kaggle.com/>), Google data set research (<https://datasetsearch.research.google.com/>) y UCI Machine Learning (<https://archive.ics.uci.edu/>).
4. Muestra perfiles de profesionales de la ingeniería química que utilicen la programación en su día a día, con el fin de que los estudiantes reconozcan cómo estas habilidades son valoradas en el mercado laboral y cómo puede abrirles puertas a oportunidades. Un buen lugar para comenzar es la plataforma de LinkedIn, muchos profesionales están dispuestos a compartir sus conocimientos gratuitamente y publican

- constantemente consejos sobre el uso de la programación en ingeniería química. Aquí te dejamos algunos perfiles: Dr. Victor M. Zavala de la Universidad de Wisconsin-Madison (<https://www.linkedin.com/in/vzavala/>) y Dr. Luis Ricardez Sandoval de la Universidad de Waterloo (<https://www.linkedin.com/in/luis-ricardez-sandoval-182b72298/>),
- Integra la programación en tus clases de manera gradual, comenzando con ejemplos sencillos y busca utilizar plataformas de acceso libre como Google Colab (<https://colab.research.google.com/>) o Kaggle ya que muchas veces no es práctico realizar la instalación de distribuciones como ANACONDA en las computadoras de los estudiantes por los recursos que necesita. No intentes enseñar todo de una vez. Por ejemplo, podrías realizar algunos videos para utilizar la metodología de aula invertida y diseñar ejercicios cortos y prácticos para que los estudiantes los resuelvan en clase. Existen plataformas como Edpuzzle (<https://edpuzzle.com/discover>) o Nearpod (<https://nearpod.com/>) para trabajar aula invertida.
 - Lo anterior, promoverá que los estudiantes resuelvan problemas por sí mismos, por lo que trata de fomentar el aprendizaje activo. No se trata solo de mostrarle código ya hecho, sino de ir elaborando junto a ellos para que adquieran confianza en sí mismos. Puedes realizar un concurso donde los estudiantes resuelvan un problema de ingeniería química utilizando programación, incluso podrías comenzar con un concurso sobre generación de imágenes con inteligencia artificial mencionando a los estudiantes que una de las bases de esta herramienta es la programación. Aquí te dejamos algunas plataformas que te podrían ser de utilidad, Midjourney (<https://www.midjourney.com/home>), Stable Diffusion (<https://stablediffusionweb.com/es>) y Leonardo AI (<https://leonardo.ai/>).
 - Habla sobre las ventajas que tiene usar herramientas como Chat GPT, Gemini, Deep Seek, entre otras para construir código, pero muestra la importancia de que antes de que utilicen estas herramientas de inteligencia artificial, aprendan conceptos básicos de programación.
 - Proporciona recursos y soporte, es decir, trata de ofrecer a los estudiantes materiales para su autoaprendizaje, por ejemplo, puedes crear un canal de youtube, un blog, guías y repositorios para que los estudiantes tengan acceso en todo momento a estos recursos. También podrías crear un repositorio de Github o un libro digital en Gitbook para que los estudiantes lo utilicen como guía en su proceso de aprendizaje. Aquí te dejamos el canal de youtube de nuestro grupo (https://www.youtube.com/playlist?list=PL6-33Ua-bIgtUBGYIjn_fhXTdb0mfgL1cz). También te dejamos algunos repositorios de Github: Python for Chemical Engineers (<https://github.com/CACHE/Python-Chemical-Engineers>), CBE20255 Introduction to Chemical Engineering Analysis (<https://github.com/jckantor/CBE20255?tab=readme-ov-file>) y Numerical Methods in Chemical Engineering (<https://github.com/hamidrezanorouzi/numericalMethods>).
 - Algo muy importante, incorpora herramientas visuales e interactivas. Los estudiantes de hoy requieren tecnologías que sean simples de utilizar y fáciles de interpretar. Por ejemplo, podrías utilizar los cuadernos de Google Colab para agregar apuntes, links de videos de youtube, imágenes de diagramas de flujo de proceso, bloques y de tubería e instrumentación, además de incluir código de Python.
- Ahora algunos consejos para que estudiantes de ingeniería química se motiven para aprender a programar:
- Empieza con un lenguaje de programación amigable y versátil, por ejemplo, podrías comenzar con Python, ya que es muy fácil de aprender, además tiene una sintaxis muy clara y fácil de entender. Este lenguaje de pro-

gramación es bastante utilizado en muchas áreas para análisis de datos, automatización e incluso creación de videojuegos. Puedes comenzar con Google Colab (<https://colab.research.google.com/>), para usarlo solo necesitas tener una cuenta de correo de gmail.

2. Realiza proyectos prácticos y no te limites a la teoría. Toma el libro que más te guste de ingeniería química e intenta resolver los problemas con el uso de algún lenguaje de programación, no te frustres, el aprendizaje lleva su tiempo. Por ejemplo, podrías proponer un pequeño proyecto sobre un balance de materia de un reactor, resolverlo e incluso si te es posible, comparar la solución con datos experimentales. Si estás iniciando en la programación, los siguientes sitios pueden ser de tu interés: PyClubs (<https://docs.pyclubs.org/python-across-all-disciplines/disciplines/chemical-engineering>) y el Libro de Python (<https://ellibrodepython.com/>).
3. Emplea librerías de Python que puedes utilizar en tus materias durante tus estudios de ingeniería química. En la Tabla 1 y 2 de este documento podrás encontrar algunas librerías o paquetes que te pueden ser de utilidad. Trata de leer su documentación oficial y busca problemas de ingeniería química en donde podrías utilizarlas.
4. Algo muy importante, colabora y aprende con otros estudiantes, por ejemplo únete a comunidades o clubes de programación, por ejemplo, la Facultad de Ingeniería Química de la Benemérita Universidad Autónoma de Puebla cuenta con su Club de Inteligencia Artificial y Programación.
5. Lo más importante, no tengas miedo a cometer errores, recuerda que la programación implica prueba y error. Aprende poco a poco y con el tiempo identificarás fácilmente los errores en tu código, sin embargo, es de vital importancia que la pongas en práctica, no se trata solo de tomar cursos, sino de aplicar todo el conocimiento adquirido.

Para ambos grupos:

1. Siempre mantén la curiosidad por la programación, recuerda que es un campo en constante evolución, por lo que debes mantenerte actualizado. Por ejemplo, explora cada mes las nuevas herramientas, lenguajes y aplicaciones que puedan facilitar tu trabajo o proceso de enseñanza-aprendizaje. Recomendamos visitar constantemente el sitio del índice TIOBE (<https://www.tiobe.com/tiobe-index/>) para conocer las tendencias en cuanto a lenguajes de programación.
2. Es muy importante lo siguiente, siempre celebra los pequeños logros, recuerda que lo que se celebra siempre se fortalece. Es claro que aprender a programar puede ser desafiante, pero cada pequeño avance es un gran paso hacia el dominio de una habilidad muy solicitada en la actualidad.

Declaración de privacidad

Los datos personales facilitados por los autores a RD-ICUAP se usarán exclusivamente para los fines declarados por la misma, no estando disponibles para ningún otro propósito ni proporcionados a terceros.

Declaración de no conflicto de intereses

Los autores declaran que no hay conflicto de intereses

Agradecimientos

Los autores agradecen a la SECIHT por el financiamiento para los estudios de posgrado de los estudiantes María Fernanda Saviñon Flores (2023-000002-01NACF-00835), Fausto Díaz Sánchez (300349) y Fidel Aaron Maruri Valderrabano (1140530).

Referencias

- Acosta, K. R., Nasimba, E. D. L. Á. M., & Díaz, A. C. (2025). Importancia de las habilidades digitales para la enseñanza universitaria: Revisión de literatura desde el año 2020. *Conectividad*, 6(1), 240–248.
- Beers, K. J. (2007). *Numerical methods for chemical engineering: Applications in MATLAB*. Cambridge University Press.
- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2024). The Julia programming language. <https://julialang.org>
- Cedeño, E. I. B., Quintero, A. R. T., Quiñónez, O. G. A., Zamora, M. E. P., & Prado, N. G. V. (2024). Análisis de tendencias y futuro de la inteligencia artificial en la educación superior: Perspectivas y desafíos. *Ciencia Latina Revista Científica Multidisciplinar*, 8(1), 3061–3076.
- Chaves, I. D. G., López, J. R. G., Zapata, J. L. G., Robayo, A. L., & Niño, G. R. (2016). *Process analysis and simulation in chemical engineering*. Springer.
- Haydary, J. (2019). *Chemical process design and simulation: Aspen Plus and Aspen HYSYS applications*. John Wiley & Sons.
- Mayer, P., & Bauer, A. (2015, Abril). An empirical analysis of the utilization of multiple programming languages in open source projects. In *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering* (pp. 1–10).
- Mereles, J. I. (2025). Transformación digital en educación: Competencias, innovación pedagógica y desafíos sistémicos. *Revista Paraguaya de Educación a Distancia (REPED)*, 6(1), 1–2.
- Novoa, M. G. G., Alcalá, M. D. S. P., & Gamboa, R. M. (2025). Habilidades identificadas de pensamiento computacional, programación y motivación académica en estudiantes de pregrado. *Transregiones*, (9), 13–40.
- Paredes, D. A. V., Arias, O. O. F., & Martínez, L. C. C. (2025). Impacto de la inteligencia artificial en el aprendizaje de la programación informática en los estudiantes universitarios. *Ciencia y Educación*, 6(3), 33–50.

Patil, R., & Gudivada, V. (2024). A review of current trends, techniques, and challenges in large language models (LLMs). *Applied Sciences*, 14(5), 2074.

Python Software Foundation. (n.d.). PyPI – the Python Package Index. <https://pypi.org>

Rodríguez, L. V. P. (2025). Integración de proyectos STEAM en el aula potenciando el aprendizaje interdisciplinario y las habilidades en los estudiantes de educación. *Conocimiento Global*, 10(1), 34–45.

Sanatinia, A., & Noubir, G. (2016). On GitHub's programming languages (arXiv preprint arXiv:1603.00431). arXiv. <https://arxiv.org/abs/1603.00431>

Sansana, J., Joswiak, M. N., Castillo, I., Wang, Z., Rendall, R., Chiang, L. H., & Reis, M. S. (2021). Recent trends on hybrid modeling for Industry 4.0. *Computers & Chemical Engineering*, 151, 107365.

Smith, C. A., & Corripio, A. B. (2005). *Principles and practices of automatic process control*. John Wiley & Sons.

Stephanopoulos, G. (1984). *Chemical process control* (Vol. 2). Prentice Hall.

Ye, Q., Axmed, M., Pryzant, R., & Khani, F. (2023). Prompt engineering a prompt engineer (arXiv preprint arXiv:2311.05661). arXiv. <https://arxiv.org/abs/2311.05661>